Imperial College of Science, Technology and Medicine
University of London
Department of Computing

CONTEXTUAL MEDIATION TO SUPPORT UBIQUITOUS COMPUTING

Daniel Chalmers

12th August 2002

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering of the University of London, and for the Diploma of the Imperial College of Science, Technology and Medicine

ABSTRACT

The circumstances affecting the use of devices may be termed the *context* of use. This includes location, social situations, tasks and the characteristics of the device itself. The ubiquitous computing vision predicts that the context of use of computing devices will become much wider and more variable than today. A default presentation of data may not always be ideally suited to the context of its use. The user's expectations and experience of data presented will vary widely with context. In more restrictive situations the volume of data which may be available is likely to overwhelm the network, display capability, or user. Where the data can be transformed, or presented in part, then there may be a mediation process to find the best presentation for the context of use. In order to manage this *contextual mediation* in an automatic and acceptable manner a process which is sensitive to the effects on the user is required.

This thesis examines the description of context and data and uses this to enable a general specification to direct contextual mediation. Structured data is described in terms of its semantic content and attributes of alternative representations of semantic elements. Description of context as both states, such as speed and the user's activity; and consumable resources, including network bandwidth and screen space, are presented. Context is used to enable specifications which reflect needs and limitations due to context in a rich and highly flexible manner. Preferences are considered for both semantic and syntactic properties of data. In pursuit of this general solution, we have developed a test-bed application related to the use of map data. Map data is an interesting exemplar as location-correlated data are clearly useful within a ubiquitous computing scenario; and vector map data offers a level of structure which is now emerging in many other classes of data. Results are presented showing that our techniques can indeed both describe and enable operation within a range of contextual constraints. In order to support our claim for generality, a case study of the technique's application to web data is also presented.

ACKNOWLEDGEMENTS

Firstly, thanks to Professor Morris Sloman and Dr. Naranker Dulay for all their guidance, constructive criticism and support over the past four-plus years.

Many thanks to Sarah Chalmers for so much support, patience, love and proof-reading. I couldn't have done this without you. Thank you to Eloise Daisy Chalmers, who gave me the push to get this finished, as well as for being a wonderful daughter. Thanks also to Mum and Dad for encouraging me throughout my education.

Many thanks to all the rest of the DSE group and the Department of Computing who have made working on this PhD enjoyable. In particular Anne, Arosha, Bashar, Christos, Dimitra, Duncan, Emil, Gawesh, George, Guang-Zhong, Ioannis, Jeff K., Jeff M., Julie, Krish, Leonidas, Mark, Mike, Nat, Nicodemos, Nicola, Paul, Pete, Sebastian, Sophia, Susan, Tim, Tony, Will and Ulf, who have variously drunk coffee, shared offices and gone to the pub with me, provided advice and feedback on my ideas and otherwise been great to work with. Thanks also to Danius, Dave, Les and Wendy at the University of Southampton who were encouraging and asked insightful questions throughout.

I should like to thank the EPSRC for the funding under which I was employed while working on this, specifically grant GR/M 26428 "Link-me" and grant GR/L 06010 "Location Service for Mobile Multi-Media Environments".

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CMIF	Centrum voor Wiskunde en Informatica (`http://www.cwi.nl/`) (CWI) Multimedia Interchange Format
CSS	Cascading Style Sheet
DTD	Document Type Definition
DOM	Document Object Model
ESRI Shape	A map data format
EXMAScript	European Computer Manufacturers Association Script
GIF	Graphics Interchange Format
GIS	Geographic Information System
GML	Geography Mark-up Language, an XML based map data format
GRiNS	GRaphical INterface for Creating and Playing SMIL
GPRS	General Packet Radio Service
GPS	Global Positioning System
GZIP	GNU ZIP, a GNU free software compression utility
HCI	Human Computer Interaction
HTML	Hyper-Text Mark-up Language
HTTP	Hyper-Text Transport Protocol
IETF	Internet Engineering Task Force
IR	Information Retrieval
JPEG	Joint Photographic Experts Group, used to refer to the image format standard this committee produced
LAN	Local Area Network
LCD	Liquid Crystal Display
MIME	Multipurpose Internet Mail Extensions
MPEG	Moving Picture Experts Group, used to refer to the moving image format standard this committee produced
NTF	National Transfer Format, defined by BS 7567, refers to NTF v2.0 level 3 in this thesis
PC	Personal Computer
PDA	Personal Digital Assistant
PDF	Portable Document Format
PNG	Portable Network Graphics (format)
QML	Quality of Service Modelling Language
QoS	Quality of Service
RDF	Resource Description Framework
RFC	Request for Comments, a class of document produced by the IETF and others
RSVP	ReSource reserVation Protocol

SMIL	Synchronized Multimedia Integration Language
TCP	Transmission Control Protocol
UbiComp	Ubiquitous Computing
UDP	User Datagram Protocol
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World-Wide-Web Consortium
WAP	Wireless Application Protocol
WML	Wireless Mark-up Language
WWW	World Wide Web
XSL	eXtensible Stylesheet Language
XML	eXtensible Mark-up Language

Map Symbols Key

The map symbols used in this thesis are taken from, or are similar to, those used by the Ordnance Survey. We provide here a reference to the most common symbols we use.

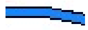
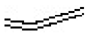
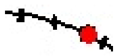
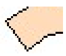
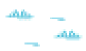
Symbol	Meaning
	Motorway
	Main road (A-road)
	Secondary road (B-road)
	Minor road
	Footpath (national trail)
	Railway, with station
	District boundary
	County and district boundary
	River
	Lake
	Woodland
	Buildings
	Antiquity
	Camp site
	Caravan site
	Castle
	Cathedral
	Church (used for with-tower, with-spire, with neither tower nor spire in this thesis)
	Golf course or links
	Historical building
	Marsh

Table Pre-1: Key to Map Symbols

Symbol	Meaning
	Museum
	Nature reserve
	Parking place
	Picnic area
	Telephone, motoring organisation
	Telephone, public
	Tourist information centre
	Triangulation point
	Viewpoint
	Windmill
	Youth hostel

Table Pre-1: Key to Map Symbols

CHAPTER 1 INTRODUCTION

1.1 MOTIVATION

Weiser's ubiquitous computing vision [120] suggests that computing devices will be found in many situations and applied to many more tasks than they are today. This is already occurring in the area of mobile computing – laptop computers, Personal Digital Assistants (PDAs) and web-enabled mobile phones are now in common use. In the future we can expect to find computing devices embedded in the infrastructure of buildings and in everyday artifacts. This view of computing naturally leads us to expect an even wider range of devices than we encounter today and a wider set of circumstances in which they will be used. The term *context* is used to refer to the circumstances relevant to the interaction between a user and their computing environment (we discuss the definition of context in chapter 2, other definitions can be found in [23,44,46,97,103,111,120]). The relevant circumstances include those which motivate the interaction, such as the task being worked on and current location; and those which impact the interaction, such as the device characteristics and environment.

Many applications today are designed with the assumption that they are to be run on a specific class of device or used in a limited way. Mobile and ubiquitous computing devices exhibit: a wide range of user interface capabilities, a variety of network connections subject to various environmental effects and may be applied to a wider range of tasks than traditional computing systems [4,31,103,120]. The deployment of applications on mobile or embedded devices exposes the user to this range of variation in behaviour, which many of today's applications and system software are unable to accommodate gracefully.

Systems which can react to context in some way are termed *context aware*. There is a need to use context awareness in mobile and ubiquitous computing to manage the incompatibility and frustration which can arise from these variations in context and user preferences. The range of variation in context renders many of the low-level network approaches to Quality of Service (QoS) management inappropriate or infeasible. The wide range and rapid evolution of devices and user needs makes tailored design of applications and data a moving target which is hard to meet. Application aware adaptation [69] seems to be more appropriate. Describing both data and user preferences enables application level decisions about the most appropriate way to adapt to the context of use.

As pervasive availability of- and range of subjects considered by data increases, it is likely that a wider variety of data may be offered in response to any request. In a scenario where the context of use is much more varied than today this wide range of information is both an advantage and a disadvantage. By considering a wider range of data the user's needs are more likely to be met within some portion of that data. Some of this data may be generated to reflect common classes of context and some will be generated dynamically to meet specific needs. However, much of the data is also likely to be of little value. In any case there is often more data available which is somewhat applicable to a user's context than the user could want, even in ideal circumstances.

In this thesis we shall put forward a system for supporting users of web-style applications in a wide range of contexts. This operates by selecting a subset of the data which is offered as potentially satisfying a request. The selection mechanism must therefore be able to differentiate between necessary, useful and unwanted data (with many degrees in between). This must be balanced with factors including timely delivery, screen space availability and cost. Different users, in different situations, will have widely varying preferences. The user is best served by data which is finely adjusted to their needs. However, many users are unlikely to be willing to spend much time describing those needs, being more inclined to take the simpler approach of choosing amongst a set of profiles which abstract those needs. Successful management of these issues, in a predictable and unobtrusive manner, will be key to the success of advanced applications. The range of devices and tasks users apply them to will constantly be expanding, therefore it is undesirable for content providers to be obliged to provide multiple versions of their content, or application designers to produce many versions of a program, each adjusted to a small range of devices or users. A mechanism for content negotiation which allows for general specification of requirements according to context is needed. We call this *contextual mediation*.

The problem addressed in this thesis is that of providing a flexible approach to contextual mediation. This requires that the approach is capable of taking account of the many context aspects which may limit or motivate operation. These context aspects include, but are not limited to:

- Device characteristics, such as:
  - Network performance, e.g. throughput and round trip time.
  - Screen characteristics, e.g. size, resolution, colour depth and readability under current lighting conditions.
- The task which the application is supporting, e.g. making deliveries, commuting to work.
- Other activities engaged in at the same time as the supported task, e.g. the process of driving, distinct from the task of making deliveries.
- Location, e.g. path to be followed and distance from that path.
- Cost sensitivity, e.g. users on business may be less sensitive to cost than others.
- Privacy, due to use in a public place, also expressed as co-location with others. This would include the display or non-display of sensitive data, or the use of audio in public.
- Environmental effects on use. This would include the impact of noise on the use of audio data and lighting on screen visibility, both of which would affect the appropriateness of the media form in extreme conditions and the presentation of the data in less severe cases.

One class of applications which are clearly applicable to ubiquitous computing are those which use location-correlated data, such as maps. We use this example to motivate many of our examples and in the proof-of-concept application we have developed. Map data can be extended with many media types beyond the traditional 2D representation of topology. Even within this application domain there are a wide range of media and classes of features of varying utility which may be offered to describe an area. This wide and well defined semantic range aids our approach. While we are working with map-based applications, our techniques are intended and implemented to be generally applicable to many other applications, such as web browsing and message, news and context notification services. The standard vector map formats available provide data with a rich structure typical of many emerging media standards (e.g. MPEG-4 and the many eXtensible Markup Language (XML) based formats), making the techniques described directly applicable over other media. Structure may be within a single data stream or file, where the composition of the stream is structured and may be varied. Alternatively structure may be realised through a collection of data elements with a description of structure which is either separate to- or embedded within the data, e.g. web pages.

We also consider the applicability of the techniques to web pages. Web pages exhibit structure which allows selection of partial presentations and differentiation by both semantics and syntactic properties of data elements. Web pages have the potential for wider variation in data *encoding* and *modality* than maps – different variants of data elements in a web page may make substantially different representations of data. For example, text may be presented as Hyper-Text Mark-up Language (HTML), Portable Document Format (PDF), plain text, as audio data, in different languages, summarised to different degrees or as hyperlinks to sub-sections of the document which may themselves be presented in different ways. As the recent use of the world-wide-web (WWW) has shown web pages and associated protocols and encodings, such as Hyper-Text Transport Protocol (HTTP) and XML, may be used to encode a wide variety of data and application interfaces.

1.2 Contribution

To address the impact of context and support contextual mediation we present several techniques:

- A model of contextual conditions. This includes both modelling of abstract contexts and monitoring of dynamic resource availability. The range of aspects of context our models can be applied to is not limited in the model's definition.
- A model of the data over which mediation is to be applied. We address issues both of the semantics of the data and issues in the encoding of the data.
- A technique for describing preferences arising from the context. The flexibility of these specifications is a central benefit of our approach. Preferences written independently to address different classes of data and different contexts require a means for resolving conflicts between them.
- We describe the criteria to be used in performing the mediation. The selection algorithm employed and its applicability are addressed. The selection algorithm used defines a view on how to use the specifications to derive an order amongst offered data and hence a *best selection* – which is the starting point for mediation.

The main contribution is a system for specifying the needs and limits arising from context. Specifications consist of preferences due to semantics of the data, preferences for different values taken by the attributes of different variants of the data and goals which reflect resource constraints. The specification techniques apply common techniques from multimedia (weights, utility functions and resource use goals) for encoding preferences. These can be written to reflect different aspects of the context. For instance, preferences arising from "shopping" may be separated from those arising from "driving a car". Preferences may also exhibit structure allowing the context which they reflect to be refined. For instance, a general set of preferences for "driving a car" may be defined and a limited sets of changes extending these defined to reflect needs for “driving at 30mph”, “driving at 50mph”, etc.

This thesis provides a description and prototype implementation of those elements of a system which provides contextual mediation. The goal of the system is to provide a generic mechanism which may be applied to many applications in many contexts. We developed models for representing context states and limited resources such as network throughput and screen space. These are generic models for aspects of context which impact a wide variety of computer use and do not assume or limit the aspects of context which they may be applied to. Similarly the model of data presented is not unique to the map example used to motivate the discussion, but can be applied in general to structured data.

The use of context to control mediation is unusual. The prior work concentrates on using context to trigger actions, discover resources or annotate data. The need for a mediation mechanism is mentioned in early work on context awareness [103,120]. The literature however only describes mediation due to varying context in terms of a general desired function. Work describing implementation focuses on adaptation to precisely defined needs and the mechanisms for adaptation, whilst we examine techniques for deriving these needs from the context and setting them against a range of limitations. Our technique offers a degree of flexibility and generality in the range of context aspects and response to context which we have not seen elsewhere in the literature. In terms of system constraints our resource models also offer a range and responsiveness which contrasts with the narrow focus of many of those in the literature. Our model of the screen goes substantially further than reflecting screen size with different size images, splitting text into sections, or changing font size. Our network model monitors a wide range of conditions rather than statically defining a small class of restricted devices or assuming wide deployment of specialised protocols.

1.3 Prior Work Published

Relevant work published prior to this thesis involving the author are detailed below:

a. Chalmers, D. “Quality of Service in Mobile Environments” *MSc Thesis* Imperial College, 1998

b. Chalmers, D., and Sloman, M.S., “A Survey of Quality of Service in Mobile Computing Environments” *IEEE Communications Surveys, http://www.comsoc.org/pubs/surveys/,* Vol.2, No. 2. (pub. IEEE 1999).

c. Chalmers, D., and Sloman, M.S., "QoS And Context Awareness For Mobile Computing" *Proceedings of 1st Intl. Symposium on Handheld and Ubiquitous Computing (HUC'99)* Karlsruhe, Germany, Sep. 1999, pp 380-382 (LNCS 1707) (pub. Springer-Verlag 1999).

d. Chalmers, D., Sloman, M.S., and Dulay, N., "Map Adaptation for Users of Mobile Systems" *Proceedings of 10th Intl. World-Wide-Web Conference (WWW-10)*, pp 735-744, Hong Kong, May 2001 (pub. ACM 2001).

Elements of this work, particularly the WWW10 conference paper, d [32], are included in this thesis. The author was responsible for the implementation of the applications described, performing tests and gave a substantial input to the ideas developed and the descriptions written. This thesis gives a more comprehensive description of the ideas involved and work undertaken than the combination of these publications.

1.4 Outline of Thesis

Chapter 2 presents the background to this work. We review the literature relating to the various parts of our problem domain and evaluate existing solutions to the various parts of our approach. From this we draw a scope for the problem we address and establish its place in the field.

Chapter 3 provides an introduction to our work, establishing a model which is used by the more detailed chapters. We give an architectural view of our system and a brief summary of the scope of the various parts implemented. In this chapter we also introduce our main exemplar application, adapting map data, to support the examples used in the following descriptions.

In chapter 4 we discuss the data we are seeking to mediate and how this data may be modified to meet contextual needs. We discuss data and adaptation in general terms and with more specific examples from cartography. We also discuss the provision of meta data and the needs we have from a description to support the flexible mediation we seek. A second key input to the problem domain is explored in chapter 5 – that of context awareness and resource management. We discuss a model for aspects of context and how contextual data is used in our work. As a specialised form of context we also discuss resource management. Our work has included the implementation of management of consumable resources and monitoring of network characteristics. The techniques employed are presented.

The description of our main contribution is presented in chapter 6. Our techniques for specifying adaptation through preferences for data and limitations due to context are presented. Following the presentation of the techniques employed, we offer some support for their effectiveness. Building directly from the specification of adaptation is the mechanism for adaptation, which is described in chapter 7.

In chapter 8 we elaborate on the map application case study and illustrate how the techniques are applied. This is intended to illustrate the validity of their application in the context of a real problem. Chapter 9 takes our implementation of this exemplar and presents an evaluation. Firstly, we present a summary of the testing environment; followed by results of objective measurements in simulated and real environments. This is followed by results from a limited user study. Chapter 10 describes a further case study, to support our claim that the techniques can be employed outside the domain of mobile use of map data. Chapter 11 concludes the case studies and evaluations by presenting a critical evaluation of the prototype application and reviewing the lessons learned while undertaking the work.

Chapter 12 summarises our work, draws conclusions, examines the wider applicability of our results and indicates directions for future work.

CHAPTER 2 MOBILE SYSTEMS AND ADAPTIVE APPLICATIONS

The capabilities of mobile devices and wireless networks available today are diverse and widening, with many practical solutions which free users from the constraints of the desktop [3] (Abowd et. al.). The availability of compact devices with significant computing power, memory and network connectivity continues to increase. In many cases this greater power is available at consumer prices and it is a common assumption (fed by advertising and technological evangelists) that the capability of these devices will continue to increase for the foreseeable future, so that soon mobile devices will not face today's limitations. However, there are technological, business and human barriers to this evolution. It cannot be expected that the limitations of mobile devices will disappear in the next few years [9,66,103,120]. Limitations in the use of these devices include:

- Limited network coverage; lower speed; higher connection times and costs than desktop systems, for example due to short range wireless networks and cellular telephony based networks.
- Limited power source capacity, e.g. battery life vs. mains electricity; output of solar cells for a given size; output of solar cells in poor light; cost and bulk of batteries or fuel.
- The constraints practical user interfaces place on size, e.g. buttons need to be pressed individually; displays need to be large enough and bright enough for users to see.
- The limitations of user interfaces within adverse environments, e.g. audio output and speech recognition become harder to use in noisy environments; LCD displays become hard to read under bright light.
- Some modes of interaction are socially inappropriate or insecure, in some situations, e.g. audio interaction in cinemas and theatres; generally visible or audible interaction on public transport. This is a limitation due to the environment of use of a device, rather than the device itself.

Software and users' expectations have consistently evolved to take up the benefits of new technology over the history of computing. Where devices are less able than the best device we routinely use then they will give an impression of being limited, however powerful they are. Mobile or embedded devices and wireless networks naturally have to compromise in order to meet portability and power consumption needs, so can be expected to remain limited (with respect to desktop systems) for the foreseeable future. At the same time the usual approach to data provision is to cater for the most capable or most common systems, ignoring the problems of less than ideal circumstances.

In section 2.1 we shall examine in more detail some of the possibilities technology offers and limitations it creates, in the world of mobile and ubiquitous computing. In the subsequent sections we shall summarise the techniques which may be applied to resolving some of the issues raised.

2.1 Mobile Devices and Networks – Enabling and Limiting Technology

2.1.1 Mobile, Embedded and Environmental Devices

A key consideration when working in the field of ubiquitous computing is that the devices to be interacted with are not simply smaller versions of a desktop PC. While laptops and Personal Digital Assistants (PDAs) now offer many of the capabilities of desktops in terms of application portability, display size etc., there remains a large class of devices and situations where desktop computing modes of interaction are simply not feasible. Mobile phones, watch based computers, dashboard computers in cars, seat back computers in planes and trains and PDAs are all limited, to some extent, by the practical size of the device. There are limits to the amount of information a user will be able to see, especially if their eyesight is not perfect, the lighting conditions are not ideal, or the display is moving in relation to their eyes (however far display fabrication advances). As Jones et. al. [68] remind us, there has been much work on HCI for small displays and the manner in which data is presented. The selection of which data can be presented and understood is key to making these smaller devices usable.

Treating devices which are used to support other tasks and devices which do not present themselves as traditional “computers with a screen and keyboard” takes a further step in understanding modes of interaction. In ubiquitous computing it cannot be expected that the computer is the focus of attention: for instance it may be in use to support driving a car, or enjoying a museum. In these cases the computing device is a tool which should recede into the environment, effortlessly providing support when needed and not distracting at other times. Ubiquitous computing where the computing devices area blend into the environment is also termed *invisible computing*. Interaction here can benefit from research into usability, e.g. Vanderheiden [112]. Where design anticipates that all senses may not be fully available the possibility that data has to be presented in substantially different ways is core to application development and information provision.

The ubiquitous computing vision also suggests that there will be many devices to interact with in any given situation, with each offering different capabilities. [24] gives an overview of the EasyLiving project at Microsoft, which addresses some of the issues of use of environmental computing. Their focus is on using location within an intelligent environment to support interactions, e.g. by tracking users to migrate applications and settings to select the most appropriate devices for the user to interact with.

Connection Type	Bandwidth (b/s)	Typical Connection Delay (s)	Coverage and Mobility	Typical Ongoing Costs
Dial-up modem (Home use)	19.2k - 56k (33 - 56k typical)	20	Some connectivity in most countries. Higher speeds in developed countries. Connection at fixed points only.	Telephone time and subscription.
(A)DSL (Home and small business use)	128k - 2M (128k - 512k downstream typical)	0	5km from suitable exchange. Connection at fixed points only.	Subscription.
Local wired ethernet (Business use)	10M - 1G (100M typical) shared or switched	0	Office site, typically slower links between sites. Connection at fixed points only.	Infrastructure maintenance.
GSM data (2G) (Personal mobile devices)	9.6k	20	Good coverage in developed countries, some deployment elsewhere. Highly mobile.	Telephone time and subscription.
GPRS (2.5G) (Personal mobile devices)	115k (20-50k typical)	5	Initial deployment in developed countries. Coverage improving, 2G networks available as fall-back.	Packet charge and subscription.
UMTS (3G) (Personal mobile devices)	100k - 2M (384k downstream expected [113])	3	Experimental deployment in developed countries.	Packet charge and subscription.
IEEE 802.11 (Business and home networking)	1M - 54M (5M - 23M typical) shared	0	Coverage of 10m-30m in a typical office, up to 1km an open environment. Highly mobile within and between base station coverage areas. Becoming more common.	Infrastructure maintenance.
Bluetooth (Networking of personal devices)	57.6 k - 723.2 k (data)	30	5 - 20m range. Not designed for high mobility between piconets.	None.

Table 2-1: Characteristics of Common and Emerging (Last-hop) Networks

2.1.2 Wireless Networks

The range of networks a mobile user may connect over is highly variable. In any given day, connectivity ranging from low bandwidth mobile phones to fast office LANs may be available. While "3G" mobile networks are widely touted as opening up many possibilities for mobile network access, there is also some concern regarding their costs and the benefit they provide over "2.5G" networks, based on General Packet Radio Service (GPRS). For instance, as discussed in the computer industry newspaper *Computing* [92] the cost per head of 3G licences in Europe ranges from $0 (Finland, Sweden) to $573 (UK). They discuss the possibility that 2.5G services will be found to be "good enough, cheap enough and sufficiently well marketed", as was VHS in comparison with other video tape technologies, that 3G will not be widely adopted. In any case availability and economics will dictate that a wide range of network qualities will be met by anyone who travels beyond their own "back yard". We summarise the most popular current last-hop networks and some emerging network types in table 2-1. This is not an exhaustive list of technologies, but describes technology in active use in Europe and illustrates a range of bandwidths and connection delays which a globally travelling user may reasonably be expected to experience in the near future.

2.2 Context Awareness for Mobile and Ubiquitous Computing

2.2.1 A Definition of Context

The notion of *context awareness*, from the point of view of mobile or ubiquitous computing, has been a topic of interest since 1993 [46,103,120]. This has given rise to a variety of definitions of context:

- Brown et al [23] suggest that context can be used "to describe the environment, situation, state, surroundings, task and so on".
- In the field of agent research Turner [111] defines situation as "the entire set of circumstances surrounding an agent, including the agent's own internal state" and from this context as "the elements of the situation that should impact behaviour".
- A fundamental paper on context awareness by Schilit, Adams and Want [103] emphasises that context depends on more than location i.e. proximity to other users and resources or environmental conditions such as lighting, noise or social situations.
- Dey and Abowd [44] take a definition of "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".

- Petrelli et al [97] spend some time discussing what context means in different fields and arrives at a definition similar to that in [44], restricting the entity to “the person who is interacting with that specific application at a certain time”.
- The *Oxford Dictionary* (1995) gives a definition of “the circumstances relevant to something under consideration”.

Given the dictionary definition, if we take the thing under consideration to be some interaction with a computing device, we have a definition which encompasses the others quite satisfactorily. Taking this slightly computing oriented view of the dictionary definition and also remaining close to that in [44], we shall use a definition of:

> *Context is the circumstances relevant to the interaction between a user and their computing environment.*

The terms used in the definitions in the literature and in their examples suggest that some aspects of context are more important than others, these include: location, identity, activity and time [44]. We have spent some time discussing the network and device as enabling and limiting technologies, clearly these can be included as part of the context. We prefer a definition which stands without the benefit or restriction of examples, although in this thesis screen size, network characteristics, task, location (in absolute terms and logical terms), speed and social situation (including co-location) are all widely used in examples.

2.2.2 How Context May Be Used

As we have discussed, the experience of users of mobile or ubiquitous computing devices is subject to a wide range of external influence. We identify five uses of contextual information, drawing from Dey & Abowd and Schilit, Adams & Want [44,103]:

a. *Contextual sensing* – where the context is sensed and information describing the current context, e.g. location, temperature, can be presented to the user.
b. To associate context with data, known as *contextual augmentation*, e.g. records of objects surveyed can be associated with location, meeting notes can be associated with people in the meeting and the place the meeting was held [23] (Brown et. al.).
c. To enable *contextual resource discovery*, e.g., to cause printing to be on the nearest printer.

Contextual adaptation [44] is used in the literature to describe both of the following two cases. We draw a distinction between the case where context causes an action and where it is involved in modifying actions, which have been caused separately.

d. *Context triggered actions* [103] to trigger actions such as loading map data for an area to be entered, or exchange business cards [105]. In our map based application we make some use of changes in location to trigger data loading.

e. *Contextual mediation* – using context to modify a service. For instance (the subject of this thesis, especially chapters 6 and 7) to describe limits and preferences over a large range of offered data, in order to display the most appropriate parts. The request for the data being mediated need not arise from the context.

A summary definition of a *context-aware system* as "(one which) uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task" is provided in [44]. We take the user's task to be part of the context (context being all the relevant circumstances) and would like to include indirect use of context, such as contextual mediation. We therefore modify their definition to:

> *A system is context-aware if it uses contextual information to provide relevant information and/or services to the user, or to enhance the provision of services.*

The research field of context-aware systems is broad and there has been much work, particularly with respect to contextual sensing, contextual resource discovery and context triggered actions. Similarly examination of location, screen size and network capability have received much attention in isolation, while other aspects of context are relatively unexplored. Kermarrec et. al. [71] identifies open research problems (in 1998) for supporting contextual objects in adaptive mobile computing. These include:

- *Definition of a context* – a way to clearly express context.
- *Adaptation* – "The ability to retrieve the right value of a contextual object depending on users' requirements and local context", which we have termed contextual mediation.

Definition of context can also be interpreted as a process of abstracting context. Context as perceived by humans is based on a wide ranging and subtle set of inputs and memory. Context as perceived by computer systems is a rather blunt tool in comparison. A description of context should allow us to harness the inputs we have as effectively as possible.

The process of contextual mediation should provide the best rendering of the most appropriate information within the constraints of the context (such as in a timely manner). This is a different prospect from providing as much information as possible at any time, or of making maximum use of resources at any time. The most appropriate presentation of information may be to do nothing.

In more cases providing the essential information clearly and without distraction will in fact be more powerful than providing all information offered.

We shall address the issue of what to do with contextual information later, but to conclude this section we shall briefly examine techniques for sensing context and describing context.

2.2.3 Encoding Accessability Limits in Context

Presenting media in suitable forms for those with sensory impairment and selecting data which inform of provisions for the disabled is not a topic which requires one to be using mobile devices or interacting with any particular form of media presentation, although much of the relevant prior work in this area concentrates on web pages. As discussed by Thimbleby in [110] poorly designed systems can be disabling (in that they remove normal ability) in themselves and poorly designed systems used while involved in activities such as driving can be very dangerous. As Fink and Kobsa noted in [48], techniques for user modelling and adaptation should be useful for a range of people with disabilities. Under our definitions of context and aims for generality, we should expect to be able to encode these requirements as easily as other context factors, e.g:

- Output in a tactile or audio format rather than a screen for blind users.
- Screen output with thicker lines, or reduced use of colour for the partially sighted, those with colour blindness, or normally sighted users using an LCD display in a bright area.
- Inclusion of wheelchair access information, availability of induction loops, etc., for those for whom the information is important (and omission otherwise). This might be applied to maps where these may indicate such facilities and also to web pages describing locations.

2.2.4 Sensing Context

Before we get too carried away with the benefits of context awareness, we need to be clear that automating this awareness is not trivial. We have defined context in a wide ranging way and highlighted aspects of context which both lend themselves to machine sensing, e.g. device capability; and aspects which are harder to detect, e.g. social situation and user’s task. As Cheverst, Davies et. al. [35] highlights, incorrect sensing of context, or inappropriate reactions to context, can be as great a problem as insensitivity to context. Any reactions must support the user in a way which the user expects or does not notice, otherwise they will be perceived as intrusive.

Schmidt [104] notes four primary approaches to sensing context:

- Active environments, e.g. active badges. Here context is sensed by the environment, which may inform the user's device on request. Active badges do not in themselves know their location, they provide a beacon from which sensors in a room can determine the badge's location. The user can then query the room to find their location.
- Specific sensors, e.g. (Global Positioning System) GPS. This is the inverse case of the active environment. In this case the device carried senses the environment and determines the context.
- Direct input to the application, e.g. calendar application looking up a date. In this case context is really a statement of interest – looking at a future date in the diary does not change today's date. Similarly a location service may offer the user the facility to look up a location of interest, overriding the current sensed location.
- Databases, e.g. calendar, to do-lists, profile. In this case context is inferred from expected actions – a calendar may describe where one expects to be, a profile may describe where one usually is at this time of day.

In [44,105] contextual data from sensors are combined by matching combinations of sensed conditions, in order to describe overall contexts. Some combination of input from different sensors may be used to improve accuracy or confidence in correctness for an aspect. This fusion of sensed data is described for location by Leonhardt and Magee in [81]. A toolkit of *context widgets* is described by Sabler, Dey and Abowd in [101], drawing from widgets in GUIs, these include:

- IdentityPresence – which reports the arrival and departure of people at a location, e.g. by use of active badges or network monitoring.
- Activity – which senses the level of activity in a room, e.g. by use of noise level.

This kind of library of common functions coupled with the common deployment of system monitoring and resource control functions will form the basis for widespread development of context aware applications.

Some forms of context are not easily amiable to automatic sensing or require that the user is able to override or provide initial values to a model: the task the user is engaged in, the user's interests, or a location to be visited some time in the future all fall in this category, e.g. [41]. It is also important to be careful when trying to extract or assume preferences from known information. In [98] Petrelli et. al. found that users of a museum guide system could not be characterised by their age, profession or education (parameters of user profile). At the same time, they found that the most important models were the group making the visit (social context / co-location), whether the visit was the first or a repeat (history), the foreseen visit duration (may be available from user input or a diary) and preferences for a given level of interactivity (user profile). In this last case an initial value from the user is likely to be needed in order that the system is not disregarded as unpleasant to use.

2.2.5 Describing Context

Some of the work on describing context resolves it to a single value, e.g. [76] (van Laerhoven and Aidoo). The sensed aspects of the context are combined and compared with a model to provide a value, e.g. “driving to work”, or “shopping”. While this is useful in areas such as contextual augmentation and context triggered actions, in other cases it may be more useful to maintain separation between aspects. For instance, when mediating the type of notification to use for incoming messages the fact that audio is inappropriate in a lecture theatre may well be constant. It is not desirable to have to extract this from one of many possible contexts, e.g. “giving a lecture”, “in a lecture”, “in a meeting” or to have to describe “be quiet” in all those contexts. An example of contextual augmentation where context is described as several separate values in triggers can be found in Brown et. al. [23].

A mechanism is required to describe a range of contextual data, which may be held with a variable degree of certainty. Schmidt et. al. [105] uses a model of context of a set of two-dimensional vectors. The dimensions being a symbolic value describing a situation and a certainty rating that the user or device is currently in this situation. Certainty may be due to many factors, e.g. trust in the sensor, corroboration by multiple sensors, age of reading, as seen in [44,81,105].

As noted by Efstratiou et. al. in [47] once context is described and available to adaptation mechanisms and applications, care must be taken that conflicting responses do not occur. Our system for combining specifications takes a view on resolving differences.

2.3 NETWORK VARIATION

Related to contextual mediation, in the field of networking, is the area of QoS management. QoS provision (including over wireless networks) has been a research issue for some time, the literature is extensive and treatments varied. In this section we shall touch on the issues which are most applicable to our interests. There are subtleties and issues at the lower levels of the network stack or due to specific network or system architectures which we shall not address. Other papers which provide more in-depth surveys of the issues in QoS management include [7,16,31,66].

One of the key concepts in applying QoS to mobile environments is that of change and adaptation to change [39,109] (Davies and Srivastava & Mishra). These changes manifest themselves in various ways, we describe three classes below:

- Hideable change
- Fine-grained change
- Large-grained change

It should be noted that depending on the application and the QoS management techniques employed the boundary between fine-grained, large-grained and hideable change may vary.

Hideable changes are those minor fluctuations, some of which may be peculiar to mobile systems, which are small enough in degree and duration to be managed by traditional filtering and buffering techniques. Techniques applicable here include buffering to remove jitter by smoothing a variable (bit or frame) rate stream to a constant rate stream.

Fine-grained changes are those changes which are often transient, but significant enough in range of variation and duration to be outside the range of effects which can be hidden by low-level QoS management methods. These fluctuations can only reasonably be managed in conjunction with applications, e.g. automatic adjustment of video frame rate, or intermediate frame delivery.Typical causes of fine-grained change are:

- Movement between base stations in wireless networks.
- Environmental effects in wireless networks.
- Flows starting and stopping, thus affecting resources available to service other flows.
- Changes in available power causing degradation in functions such as radio transmission.

Noble, Satyanarayanan et al [93,94] discuss the use of a model of bandwidth, which is used to cause adaptation during admission and initiate response to changing network conditions. One of their focuses is on having *agile* adaptation, where changes are responded to quickly. This is balanced by the need for applications to show some stability in their presentation to the user and the costs of making an incorrect adaptation where the model is wrong. As found by Bouch and Sasse [20] users will rate media of low objective quality more highly if the quality is stable.

Large-grained change will occur infrequently, but are likely to involve large steps in quality presented to users, as storage or media scaling to many levels for data intensive streams is generally expensive. Causes include:

- Change of network class, e.g. movement from office IEEE802.11 network to cell-phone based network.
- Change of display, e.g. as laptop is un-docked.

Each of these may be encountered a few times a day, but are unlikely to be encountered from minute to minute. “Making do” with a default behaviour, which is probably under performing in some situations and over-stretched in others, is a poor strategy to mitigate effects of this scale and duration.

In order to handle the fine- and large-grained changes, which are likely to be met in mobile and ubiquitous computing, application level adaptation is required. It is clear that the application must be involved in this adaptation process, although system support can be applied to support this. Adaptation to this type of change has been a focus of interest for some time, e.g. [15,39,40,93,108, 109,121,128].

2.4 Resource Management

Where resources are scarce, a natural response is to impose some access controls on those resources. This can be motivated by the need for fair access where there is contention and also the need for predicatable behaviour [20] (Bouch & Sasse). Some researchers, such as Srivastava [109], contend that network resource reservation is not relevant in wireless networks, as the available bandwidth in connections is too highly variable for reservations or guaranteed behaviour to be meaningful. We agree that guaranteed behaviour is not compatible with wireless networks, however some resource allocation and admission control would seem prudent when resources are scarce. Lu et. al. [83,84] proposes that guarantees be made in admission control on lower bounds of requirements, whilst providing best-effort service beyond this. The issue of resource reservation is also given some consideration by those working on base-stations and wired parts of mobile infrastructures, as these high bandwidth components must be shared by many users, so the traditional resource management approach still applies. Adapting media to meet multiple resource constraints is a topic which has been applied in many areas of computing. Mohan et. al. describe the problem in the context of adapting to limited clients in [89].

An issue in the use of these small systems (PDAs and embedded systems), in comparison with desktop systems, is that they tend to be used in a more focused way. On a desktop a user may set a few web pages downloading then check their mail and use office applications which store data over the network, while receiving an internet radio station. Mobile devices are often used to achieve one task, possibly while engaging in another task away from the device, such as driving. Ubiquitous systems will often be used in the manner of consumer electronics, e.g. as a television, rather than as highly multi-functional systems. The limitations of the user interface, slower network connections and environment all reinforce this difference. When considering resource management and adaptation strategies for workstation based systems, competition between applications has been a major consideration. At the same time, each application has tended to be characterized as a single network flow. Where data are highly structured or combined from multiple sources, the competition can often be between flows for one application – which ought to be co-operating. We shall not be considering mechanisms for scheduling between competing applications here. As these are well covered in the literature, e.g. [63], our work does not contribute in this area. Ensuring that the various flows making up the presentation of information in one application reflect their combined effects and result in a coherent presentation is, however, a key interest to us.

We shall now review techniques and issues in a few key areas of resource management.

2.4.1 Network Delay

Ideally, all data would be presented with no perceived delay. Neilson in [91] cites 0.1 second as feeling like immediate feedback for web users. After 10 seconds the average user will divert some of their attention to another task. Various studies have been made into how delay affects perception of web browsing. Provided some feedback is provided to indicate that loading is taking place then delays of 5 seconds are generally regarded as good [12] by Bhatti and Knight, while delays of 10 seconds are still acceptable to many. For larger web pages incremental loading greatly increases tolerance for delay. Possibly as important as overall delay is variation in delay [91]: if users expect a certain delay then, even if it is not ideal, they will tolerate it more than if they were expecting a shorter wait but experience a wide ranging delay.

Pre-fetching to a local cache is an appropriate technique for content whose access can be predicted, e.g. [82]. However, where caching is off the client, this approach can do little to mitigate the problems of slow last-hop wireless network access. If on the client then pre-fetching must be performed around other tasks and caching within available memory. There is also a trade-off between how far in advance data should be pre-fetched and how quickly it becomes out of date. This approach also cannot provide a benefit for unpredicted requests.

For many classes of media detailed patterns in resource characteristics can be ignored, although overall behaviour remains crucial. For our work, we require a resource model which provides an accurate description of behaviour we can expect for a flow over a period of between seconds and a few minutes. It must tend to smooth over transient changes in resource characteristics in favour of stability. The ability to under-play improvement rather than raising user's expectations is desirable in order to maintain deadline guarantees and bound flow duration where characteristics fluctuate. At the same time agile reaction [93] to change which has a substantial and ongoing impact is necessary.

Many networks are of sufficiently good quality that requests performed by an application need suffer little or no constraint to meet users' needs for interactivity. However, there are other resources which may be regarded as dynamic parts of the context which may impose other limits on requested data, including cost, screen space and power. These obvious constraints are examined in the following sub-sections.

2.4.2 User Interfaces

To build a portable yet usable system will continue to place size constraints on designers. Devices which are too large will not be carried by choice. Devices which are too small present problems in enabling usable interaction – screens and keyboards can only be shrunk so far before they cease to be usable. Projected displays, head mounted displays, audio interaction, braille output etc. may all be used by different users at different times to access information; keyboards, hand-held chording devices, gesture control, speech control, etc. may be used to control computing systems. In this world of diversity few constraints will remain constant – through and despite device evolution and user needs.

Without having to consider substantially different modes of rendering data, it is clear that mobile devices offer a wide range of screen capabilities. Screen space is consumed by each piece of information shown on it, although it is not highly variable like network throughput. The literature on information visualisation deals with techniques for display of- and mechanisms to convey meaning from large and complex data sets, we shall not address this here. We shall start from the point of view that the user wants to use some familiar visualisation technique, such as text, images or geographic maps. If zooming is an option then this may be controlled by the user. In general, some trade-off between zooming and having sufficient data to give context in its interpretation is required. The data available to be displayed may be overwhelming when interacting via a limited device. Screen space resource management needs to ensure that the space available is put to effective use. In this consideration screen space is much like throughput: in the context of the request we have a finite amount, which is replenished for each new request. If we under-utilise what we have available we may fail the user by not conveying as much relevant information as we might. If we over-utilise the resource we are likely to render the information valueless – in the case of meeting a deadline by showing the data after it has ceased to be useful; and in the case of screen space by cluttering the screen to an extent that it becomes unreadable.

Much of the work in this area has focused on providing support for a narrow class of devices, whereas our interest is in a general support mechanism rather than custom solutions. We shall note some of the most relevant examples below, although this is an area with a long history. HCI has been a key factor in the design of many systems, such as bank ATMs. We do not intend to survey the literature relating to usability of small-screen systems in detail here. Many of the issues in user interface design for small screens are discussed by Jones et. al. in [67,68].

The Wireless Application Protocol (WAP) [117] and its various protocols and languages, describe a system designed to support users of wireless communications, particularly mobile phones, pagers and PDAs. The model followed is to provide a somewhat specialised version of the World-Wide-Web, with proxy / gateway devices performing translation with common internet services. Wireless Mark-up Language (WML) [117] is designed to describe web-page type data for mobile phones. It incorporates models of interaction that take account of the small screen and limited keyboard. It is also more compact than Hyper-Text Mark-up Language (HTML) and so more appropriate for use over low bandwidth links. While the WAP technology has significant deployment it is widely perceived as unsuccessful, this is in part due to marketing which built unrealistic expectations in users. It is also a result of the user interface limitations of mobile phones, on which it is generally seen – it is hard to convey much information or a service of any great power on a screen and keyboard so limited, however powerful the technology behind it. The application of WAP features, such as the deck of cards model of pages, to PDA class devices is described by Björk et. al. in [14].

The work by Fox et. al. described in [52,54] focused in part on adapting images and text (HTML and Postscript), for display on a variety of devices, in particular on a Palm PDA. This was achieved through *distillation*, or *transcoding*, of image content to meet screen size limits or through a standard lossy compression mechanism. However, their work focused on providing support for a specific device and on limiting download time as far as possible. The results published from extensive user studies, especially in [54], illustrate that these adaptations provides a significant benefit when browsing the web using a slow modem and PDA devices. The approach however was tailored to pre-defined devices and did not take into account the use of images or other needs of the user.

A system which provides “device-dependent access to the world wide web” is described by Bickmore and Schilit in [13]. They treat the problem of adapting web pages to be suitable for use with small screen devices. The approach described notes that while device-specific authoring provides the best looking results, it is not practical as few sites will be rewritten for any specific device. Their solution makes use of a set of re-authoring techniques. These techniques are categorized along two divisions: syntactic / semantic and elision / transformation. Syntactic elision would include section outlining, while syntactic transformation would include image size reduction. Semantic elision would involve the removal of irrelevant content and semantic transformation includes techniques such as text summarisation. These techniques are all potentially applicable, their desirability varying with the device being used, the intention of the data use and the degree to which they are applied.

The design of an application for field workers, such as zoologists, by Pascoe, Ryan and Morse was described in [96]. In this case the application operated on a limited device (Palm PDA), needed to support users whose main focus of attention was not on the computer and whose modes of interaction was limited by the need for quiet when observing animals.

The expected context of use is attracting interest in the fields of design and HCI. The impact of form factor and other activities engaged in have received substantial efforts in response to the need for usable applications on PDAs and mobile phones. This area is largely concerned with the arrangement of data input, rendering of data and control functions. These modification have generally been static configurations or mode selections for tailored applications to date. In most cases there has been little, if any, dynamic adaptation of the user interface to context. Dynamic modification of the user interface (distinct from the data conveyed by that interface) is an interesting area of context awareness for future research.

2.4.3 Cost

Charges are typically made according to a combination of two factors:

- What a service costs to provide, both in per-use terms and by spreading fixed costs over the anticipated users.
- What users will pay for a service, which may be used to limit usage and resource contention and also simply to make a profit.

Much effort is being put into charging systems for data, through subscriptions and micro-payments. Charges may be for access, or quality. When coupled with network access charges, it can be assumed that a large proportion of users will have to pay for access to data in some way.

Costs which are incurred for data transfer, due to time on-line or packet counts, can again be modelled as consumption of a resource. However, money as a resource tends to be finite over a number of requests, unlike either throughput or screen space. We shall not consider here a possible interaction between a model of preference and a charging mechanism, such as Bouch and Sasse describe in [21]. We shall assume that a simple model of price charged for data and/or time and of available money would be of benefit to a wide base of users. This will be especially true in data intensive applications, where network charges may become significant and for applications which make use of data which is expensive to generate, or offers particularly timely information.

2.4.4 Power

Battery technology continues to develop at a much slower pace than much of the rest of computing systems. This limits devices such as laptops and PDAs. Much of the ubiquitous computing vision depends on devices being largely independent of any wired infrastructure, or regular maintenance effort. This dictates an approach to power consumption which equates to the care paid to memory usage in the 1960s and ‘70s. Power sources such as solar cells would be effective here, but cannot yet meet the needs of all but the simplest devices without becoming the largest component of the device. In recent times this issue has spread beyond battery powered mobile computing to general purpose computing. An increasing awareness of the need for energy efficiency; taxes and charges designed to control emissions and demand; and power shortages which occur to some extent everywhere (and in some places quite frequently) all contribute to a need for power-aware computing.

It is clear that any operation of a computing or communications system draws some power. In energy-limited systems power consumption can be critical – battery life and heat dissipation resulting from consumption are both problems for highly portable systems. In [50] Flinn and Satyanarayanan's work with the Odyssey platform to implement sensitivity to energy usage through adaptation is described. They found that by adapting the information presented energy savings of around 30% are possible for video viewing, web browsing and map display. Other techniques are also important here:

- Controlled use of energy draining features such as back-lights.
- Hardware power management, reflecting system load.
- Adjusting wireless transmission power, when a drop in speed or range can be tolerated.
- It is desirable that applications can be made to cooperate to ensure effective management.

In many cases power management takes a binary approach: switching devices on when needed and off when idle, or only starting to conserve power once the power source starts to run low on energy. However, as can be seen in [50] while a large benefit can be found from turning off idle devices, a substantial reduction in energy consumption can be found by reducing data communicated and/or processing. All data transferred over the network takes power to send and all data processing requires CPU activity which draws power. The load on the power source can be adjusted by mediation and power drained only where the result provides sufficient benefit to the user. If the mediation occurs in conjunction with a view of the power available and an estimate of power required in the future then the resource can be managed in such a way as to meet requirements. This requires an understanding of future use, e.g. how long before the next charge, what activity may occur in that time, which could be determined from a combination of automatic and explicit context input. However, it meets the user's needs for data quality better than a system that takes a blind approach to simply prolonging the life of the power source.

2.5 MEASURES OF MEDIA QUALITY

In order to make effective selections to support contextual mediation, we need to be able to measure quality. For example, if there are three elements to a document, each of which is available in several variants, we need:

- Firstly, to be able to define which variant of each element would be the best, given the context of use.
- Then, in the case where the combined resource requirements of the best variants exceed our limits, to make the most satisfactory selection, based on:
  - Which elements are most important.
  - How much worse the other variants are and whether they help meet the resource limits.
  - Which variants are unacceptable.

An arbitrary decision is unlikely to be the most satisfactory and simply addressing the resource constraint cannot be guaranteed to provide a coherent or satisfactory selection from the user's perspective.

Noble in [94] describes measurement of data quality, referred to as *fidelity*, in comparison to some ideal reference copy. Fidelity is defined as the degree to which a data instance matches the reference copy and may be defined in many dimensions. They note that the total ordering their fidelity metrics provides does not correspond to a relative measure of quality between variants and that such a measure would depend on the use to which the data is being put. This model is elegant in that it incorporates the idea of variants of data, which differ in various attributes. However, the notion of an ideal version of the data is hard to justify for many media sources and causes the failing of being unable to use the measure to ascertain relative quality, taking into account contextual conditions, such as task and device. The definitions of the ideal versions of data are generally dependant on the beholder and their context, although some generalisations are possible. A technically ideal version of data may not be available, making comparison of degraded versions hard.

QoS control is described as providing higher layers (in software) with the ability to specify what constitutes an *acceptable* presentation and what *better* means when considering variants of data by Walpole et. al. in [115]. They use utility functions to define user's requirements over different dimensions of a multimedia presentation, their model of utility functions is one which we adopt in our work. The ability to separate the effects of different attributes on the overall perceived utility is very useful. Their model also acknowledges that the quality at which media is stored is not necessarily perfect and that the representation on the user's system may also affect the presentation.

“Payoff” functions as defined by Kravets et. al. [75] describe perceived utility in relation to aspects of a service such as reliability. These payoff values are then used to select amongst possible adaptations in response to resource constraints. Where two resource parameters are interdependent a trade-off can be made according to the payoff functions, e.g. where loss probability rises with data rate the appropriate service level can be chosen.

Work at Sussex [65] includes a user study which found that for images in web pages a substantial level of degradation could be applied to images before they were regarded as unusable. However, there were clearly situations when a degraded image would not be acceptable and the ability to summon an improved version would be required. As in Neilson’s work [91] they found that the user’s sensitivity to delay in waiting for images varied, although faster would be preferred. Varying tolerance by any individual can also be attributed to factors including why they want the image, the semantic content of the image and the cost of the link. Overall they found that a quickly received preview would be desirable. However, they point out that the user does not want to spend time or be exposed to the complexity of configuring how media should be adapted and will not be sensitive to slight variations in the trade-off taken. The set of selected variants which satisfy most users may be quite small, although care must be taken that the selection takes into account the user’s needs. When working within a constrained application / context domain assumptions can be made as to the perception of most users and a simple scale of quality conceived that maps to underlying adaptations, e.g. [6]. When building a more generally applicable system, in terms of applications and context supported, then the preferences must be drawn from the context and the available variants to select amongst drawn from meta data, while still screening the user from a stream of complex decisions.

In summary, we need to be able to describe a mapping from measurements of attributes of data to perceived user quality, in multiple dimensions, whilst taking into account the semantic content of the data element. The model of each parameter should reflect the limit of perceived improvement (for the context) and the lower limit of usability.

2.6 CONTEXTUAL MEDIATION

Enabling the presentation of data in a suitable manner for the context is a key requirement for mobile and ubiquitous computing. The adaptation may be reflecting resource limits and preferences due to context. There have been various approaches to this already, mostly focusing on providing adaptation to meet the limitations of the device being used. This section reviews relevant techniques.

We need to consider the space in which we are adapting. As discussed by Gecsei and Watson & Sasse in [57,119] there is a need to deliver data at a quality which enables the user to perform their task. Similarly, there comes a point at which no improvement in quality is perceived for an increase in the coding detail. The perceived quality will depend on various attributes of the encoding. Note that the quality perceived will be due in part to the context of the interaction and cannot generally be extracted directly from the quality of the encoding, e.g. [119]. Where the media type remains the same adaptation will generally result from the loss of information from the original version, e.g. making an image smaller, reducing the colour-depth. However adaptation should be seen as being the process of moving to the best point within a space rather than purely as degradation, although much of the literature, such as Fox, Gerfelder, Noble et. al. [53,58,93], refers to it in these terms. A high colour, high resolution image may not offer any benefit over a version which has lost much of this information where the display cannot convey the extra colour and detail.

The adaptation techniques that may be applied can include (as described by Ma et. al. [85]):

- Information abstraction, e.g. lossy image compression, text summarisation.
- Modality transformation, e.g. video to still image sequence, audio to text.
- Data transcoding, e.g. format conversion, often accompanied by some degree of abstraction.
- Selection, which may be by data prioritisation, or binary selection.

Reaction to the semantics of media elements (purpose classification in [85]) can be used to support these techniques by informing the selection of techniques to apply to different media elements.

We then need to consider how this adaptation may be best achieved. As a starting point, we see that there are two clear approaches to who makes the adaptation decision:

a. A server or (typically remote from the client) proxy based system, which takes descriptions of the context or capabilities of the client and using knowledge of the data available provides versions of the data requested tailored to meet the limitations described.

b. A client or (typically close to the client) proxy based system, which takes descriptions of the data available and using knowledge of the context or capabilities of the client selects and requests versions of the data suitable for use within the limitations described.

This classification differs from Ma et. al.'s [85] in that we classify by whether the decision is closer to the server or client, while they consider the server or a proxy to be the choices of location for the adaptor. Proxy based approaches may tend to combine elements of the two approaches, but in essence the difference between the approaches and which one is suitable for use in a situation depends on the following factors:

- The user's willingness to provide a detailed description of their context, to possibly unknown remote parties. The context may be inferred to some degree from the data requested, but various combinations of situations might lead to any given combination of data requested.
- The ability of the user's system to describe, in an efficient and timely manner, fine-grained resource fluctuations to the management system (wherever it is located). Client-located or client-side proxies have an advantage here.
- Servers will also experience fluctuations in resource characteristics. However, for static servers some degree of averaging over many connections should mask much of the effects of these. For the immediate future wireless connections at the client are likely to be the source of most instability in a typical connection.
- The data provider's willingness to describe the data on offer. Given that the data are being provided at some point, that the options described may be limited by the server and that a client could fake contexts to discover this information, excessive limitation here seems unnecessary.
- The ability of the client to handle the selection process, which may be processor or memory intensive. Where clients are most limited there is a need for a suitable proxy to perform this service.
- The ability of the server, or close-coupled proxy, to provide selection services in a scalable manner, particularly where the selection process is not trivial. This will require that server(s) maintain state information regarding each client. To be effective there will be a need for storage of client details and processing power to make selections.
- The extra data transmission time and network load caused by the communication of meta data where selection happens beyond a slow link (from the server) may be undesirable. A balance between privacy, location of computing power and network overheads must be made. Where meta data overhead is substantial, e.g. with a slow last hop network, selection may be moved to a proxy. This assumes that the cost of transmitting context information to the proxy is less than the cost of meta data transmission.

Another dimension in which adaptation may be characterised is that of the triggering of adaptation. Where media being delivered are formed from a stream of discrete samples, e.g. audio and video, it is reasonable to adapt between each sample which may stand independently, e.g. key frames in MPEG video. Where media being delivered are formed from discrete blocks of data, e.g. web browsing, it is reasonable to adapt only on a per-request basis. In some instances of this discrete case, where media are formed from large blocks of data, it may be reasonable to cancel and re-request data if adaptation is needed. However, it typically takes time to cancel a flow and the data transmitted so far is likely to be lost and require re-transmission in its adapted form. This strategy is only applicable for the early part of the transmission of a large block of data. The need for adaptation may be signalled in-flow from the server, the network, or the client; or may be examined on request, for instance during admission control. Much of the work on QoS adaptation has been based on adapting the stack or data path objects (or controlling parameters of those objects) to manage streaming media, as seen in [8,49,108]. Architectures for handling event notification have been described, e.g. [121].

2.6.1 Requirements for Mediation

We seek the following characteristics to enable effective contextual mediation:

- Handling of rich and dynamic descriptions of the client’s context and resource characteristics.
- Data will be structured, elements of the structure are adapted and requested separately. Meta data should be able to describe structure, parameters of encodings and semantics of structural elements.
- Selection should consider the total effects of all elements in a structured document, in order that trade-offs made may take into consideration all parts of the data to be requested.
- Data may be dynamically transformed by proxies, e.g. image scaling, text language translation.
- There may be more than one source of information for a presentation, e.g. traffic conditions may be overlaid on maps, web pages need not have all elements hosted by one server. The contextual mediation should be able to combine meta data from multiple sources and the distribution of data handled transparently.
- The solution should be scalable to many: users, application domains, data types, device classes and models and contexts. Solutions tailored to specific needs will ideally be supportable in the system, but the principles should be general.

2.7 PRIOR WORK TOWARDS CONTEXTUAL MEDIATION

2.7.1 HTTP Transparent Content Negotiation

As described in Internet Engineering Task Force (IETF) RFC 2295 [59], the Hyper-Text Transport Protocol (HTTP) supports the transparent negotiation among variants for a given Uniform Resource Identifier (URI). In this system a negotiable URI has a list of variants, whose attributes are described. The selection may occur at the server, taking account of accept-headers in the HTTP request, or the variant list may be returned for off-server negotiation.

The negotiation takes account of the following attributes: Multipurpose Internet Mail Extensions (MIME) type, character set, language and features. Features may include: HTML extensions, colour capabilities, screen size, output medium, preferences for speed / detail trade-offs. The description of variants includes: a numeric measure of quality (which assumes ideal rendering), attributes of MIME type, character set, language, length, features and a description, although this may be extended. The quality measure is intended to capture the effect of any lossy compression the variant has undergone.

Performing the variant selection at the server is generally assumed as this eliminates a round trip. This requires that the client describes its preferences using the accept-headers. These consist of a list of features specifications, which may indicate allowed, required and disallowed values (or ranges) of attributes and tags which should be present or absent. The negotiate header also allows the client to influence the remote negotiation. It is also possible to indicate a preference amongst acceptable values by indicating a *quality* rating for that value, e.g. where a user can read two languages, but prefers their native tongue.

Work in the IETF on selection in HTTP [60] describes a possible negotiation algorithm to be used with transparent negotiation as the core of the protocol does not assume one. The source quality and ratings derived from the attributes of the variant, are combined by product to give a rating for the variant, with the best result returned. The rating is based on the information in the accept headers. If the rating is termed “speculative”, being due to wild-cards rather than explicit preferences, then choices may be returned to the client.

Clearly this mechanism meets the basic need for selection amongst variants of media. However, we identify some areas in which it does not support the flexibility we seek:

- The negotiation is on a per-element basis. This makes it hard to perform negotiation over a complete request, which often consists of multiple resources. Any requirement to negotiate towards a deadline or to ensure the full presentation fits within a screen size cannot be met where negotiation over elements does not consider their cumulative effect, or trade-offs between elements.
- The description of variants does not include any information about the semantics of the element (although issues regarding this are starting to be captured in work by the Berners-Lee and the W3C [10]).
- Accept headers which encode preferences where the number of possible values a feature may take is not small are cumbersome. Where one can expect a small range of languages in text, or sizes and colour depths in images the system is sufficient. Where media are less consistent the negotiation will tend to revert to the client side.

2.7.2 W3C and IETF Preference Descriptions

The W3C Composite Capability Preferences Profile (CC/PP) [99] notes that mechanisms such as accept headers (in HTTP) and “ALT” tags (in HTML) are somewhat limited and described the application of the Resource Description Framework (RDF) [77] to describe user preferences and device capabilities in a general content negotiation solution.

The IETF CONtent NEGotiation (CONNEG) group’s work, described in RFC 2533 [73], gives a system for describing preferences for media based on attributes of the media. The CONNEG group have addressed similar issues to the CC/PP group and there are efforts to ensure that the two systems can inter-operate.

The descriptions include support for a detailed description of the hardware platform, software capabilities and preferences for languages and security. The encoding does indeed address the difficulties with the HTTP negotiation with regard to the richness of preferences which may be specified, due to the extensibility of the name-spaces and value definitions. While this addresses the need for transmittable preferences we shall have to look elsewhere for techniques to describe the data and perform the selection.

The question of where selection is performed aside, a system for encoding preferences is important. However, there is no mechanism described for producing these specifications, it may well be the case that different preferences arise due to different factors in the context.

2.7.3 CMIF and SMIL

W3C's Synchronized Multimedia Integration Language (SMIL) [62] and the CWI Multimedia Interchange Format (CMIF) GRaphical INterface for Creating and Playing SMIL (GRiNS) editor and player [25] describe temporal and spatial behaviour of a presentation and have similar constructs for describing multiple variants of media elements in a structured presentation. The selection may be subject to a test on various parameters of the system in SMIL, such as bit rate, language and screen size. The selection is encoded with a “switch” statement, the first acceptable version of the element being selected. This system assumes that the author's judgement of quality is the same as that of the user.

CMIF focuses on providing alternative content for various contexts or user abilities, as defined by the author. This system builds on SMIL to offer groupings of content for different users, e.g. due to the language they speak.

These systems meet our desire for a system which enables selection over whole documents and provide support for the context. However, in both cases the selection of alternatives is limited by the authored selection support and the author's definition of the conditions under which the variants will be appropriate. There is no support for the user to describe their preferences, although they may be able to choose amongst the offered alternatives. These techniques may be suitable for carefully authored multimedia documents, where the presentation is subject to careful production controls, however these techniques are less appropriate for more ad-hoc presentations where the data are used in unpredictable ways, or in contexts which were not planned for.

2.7.4 Tiempo

Like SMIL, the Tiempo model of multimedia documents [123] (developed at Stuttgart by Wirag) allows for structure which collects related multimedia elements together and can describe temporal constraints between them. There is a system of alternatives with author generated QoS “priorities”. In [124] Wirag goes on to describe a model of resource use, such as CPU time and a specification of how presentations may be adapted to meet resource constraints. The constraints are solved using mathematical constraint solving over a graph model of the document, taking into account the specified priorities. The system is similar in expressiveness to SMIL and is described with similar scenarios. Again, the specification is not designed to handle user preferences and would become unwieldy for multi-media where many, possibly dynamic, media objects may be accessed in a document.

2.7.5 XML Transformation

The eXtensible Stylesheet Language (XSL) may be used to describe transformations to be applied to documents in eXtensible Mark-up Language (XML) so that they may be formatted for a given use. Data may be encoded within XML without inferring anything about the presentation. Different XSLs may be provided which can be applied to many XML documents, which describe how to convert documents with a given XML Document Type Definition (DTD) into rendered data. For instance, different XSLs may describe different techniques for presenting the same document on different screens, in braille or audio.

A comparison of XML / XSL / Cascading Style Sheets (CSS), Document Object Model (DOM) / European Computer Manufacturers Association (ECMA) Script and SMIL for presenting films and distance education materials on mobile systems is presented by Marttila and Vuorimaa in [87]. Their approach is based on adapting the data at the presentation stage, on the client. The conclusion is that XML / XSL using XSL formatting objects provides a general, flexible solution for allowing media presentations to be adapted, however they do note that the data size can become rather large for wireless networks. They find that SMIL benefits from being an established standard for streamed media, although it is not able to accommodate new XML languages and the flexibility of SMIL presentations was found to be limited. DOM and ECMA Script have made progress in standardising the scripting environment, which suffered from differing implementations, for instance in JavaScript. DOM offers the potential for similar capabilities to XML / XSL but lacks needed features, such as an event model and style-sheets, in its first version.

2.7.6 Pythia and TopGun Wingman

There was substantial work on transcoding of data, which Fox, Gribble, Brewer et. al. refer to as distillation, to meet the (system) needs of thin-client computing described in [51,52,53]. They identified that lossy compression could feasibly be performed on-the-fly to meet the needs of systems limited by networks, hardware capability or software support for data formats. Reasonably they chose in-network proxies as locations for these operations to remove overload potential at the server or client. The additional latency of the dynamic transcoding can generally be offset against the delay from transferring a larger variant when addressing reasonable size data and slow links. The approach of proxies transcoding data and offering changes in format or quality is an important part of the solution to adapting to context. Their work focused on adapting images and HTML for display on a variety of devices, in particular Palm PDAs. However, their work focused on providing support for a specific device and on limiting download time as far as possible. The results published from extensive user studies, especially in [54] illustrate that the technique provides a significant benefit when browsing the web using a slow modem and PDA devices. Little consideration of the wider impact of context was considered, nor of defining in a more general manner what data would be best or of meeting goals such as deadlines. Another group working with proxies also notes that by not transcoding smaller elements page response time and proxy load can be substantially reduced [33] (Chandra et. al.).

2.7.7 Annotation-Based Web Content Transcoding

In [61] a mechanism of "external annotation" (metadata) which indicates properties of elements of a web document in order to facilitate transcoding is presented by Hori et. al. Their system supports the description of alternative representations of documents or their elements and a description of "hints" on splitting a web page into multiple pages. They support the description of *roles* of elements within documents such as: decoration, content, advertisement and the description of an *importance value* from the author. Transcoding to meet the needs of the client can be performed with knowledge of this information, thus enabling more aggressive transcoding of less important data. The ability to split pages for use of a small screen and to provide effective incremental loading is also useful. While we do not believe that the importance value is always valid the provision of semantic information about elements of a document, as well as properties of data, is important. A mechanism which supports this beyond web pages would clearly be useful in adapting to contextual needs.

A system for annotating web pages (including images etc. referenced within those pages) with semantic information, with the intention of facilitating automatic transcoding of the page is described in [90] (Nagao et. al.). The transcoding may offer content adaptation for disabled users, or to support use of mobile devices. Transcoding proxies maintain user profiles and perform appropriate transcoding according to these, with the semantics of the elements taken into account. The content adaptation is therefore dynamic and the data the user receives may be the product of multiple transcoders. Text is annotated with a semantic description of the information, including the grammatical structure of the sentences. Summaries of text may be produced to a specified size. Image transcoding may modify size, colour depth or compression ratio. Our approach has many similarities with this work. The location of the adaptation in a proxy is one possibility our approach allows. The separation of annotations or meta data is common to both approaches. The use of semantic information in identifying appropriate forms for presenting data is central in both cases. The work they present does not describe how the user’s specifications are arrived at. The preferences are maintained at the proxy, updated by direct interaction with the proxy separate to the flow of web requests. Our work focuses closely on how we arrive at the specifications and on maintaining appropriate specifications for the context.

2.7.8 Lowband

McIlhagga, Light and Wakeman [64,65] describe a software architecture for adaptive applications and a language for directing those adaptations. They describe interfaces to adapt various media types, together with experimental work on how adaptations are perceived by users. The language describes adaptations to be taken given a condition, such as scaling images to meet throughput and display size constraints. This work illustrates a practical method for adapting media for constrained applications and some useful results regarding acceptable adaptations. However there are two faults here, in common with many other web-browser, or video/audio stream focused work. Firstly, it defines adaptations to be taken and requires evolution of the software where media formats evolve. Secondly, there is no capacity to differentiate between media with different semantic input to the user, which is a particular problem where all media are of one format.

2.7.9 InfoPyramid

Smith, Mohan and Li describe the InfoPyramid model for adaptation [107], a system for adapting media for heterogeneous client devices. The key aspect presented is a decision system which models information presented with varying modality and fidelity, this model is used to evaluate transcoding options. For instance a presentation may be given as video, images, text or audio; depending on fidelity level and transcoding capability progression between modalities may or may not be possible. The fidelity levels can represent lossy compression, scaling of images etc. To aid selection the available variants of media are assigned scores, which can be used to select the preferred mode of interaction with given constraints. The model is intuitive and the potential for fundamentally modifying the presentation while representing the same information is important. In

[89] they go on to mention that the content value scores would be derived from some overall perceived value ratio between the original and transcoded versions. As the transcoding may involve significant changes to the presentation, it is not clear that the original would necessarily be the best in all cases. In this paper they also describe how one could apply the technique to describe structured data by assigning priorities to the elements, e.g. due to their semantics.

2.7.10 Odyssey

Noble, Satyanarayanan et. al.'s Odyssey system [93,94] has developed many of the ideas in application aware adaptation. In particular they have produced a system where an application can transparently have a system added to perform QoS adaptation on its networked interactions. Video viewing and web browsing are among the applications which have been addressed. The agility of adaptation was identified as a key parameter in [93], this characterises the ability of a system to respond and swiftly and accurately to change. There has also been work on the issue of data fidelity [102] and algorithms designed to cope with uncertain data quality. The process they have of monitoring resources and adjusting behaviour to reflect this is useful, however, we wish to address situations where blanket adaptation is not always the answer and where the data may not have a clear "best" variant.

2.7.11 Adaptive Maps

A proxy-based map adaptation client designed for use with hand-held devices is described in Abdelsalam's thesis [1]. Maps are adapted by feature elimination due to semantic type, feature size and by eliminating detail from features. These ideas are similar to the basis of those presented in this thesis. The adaptations performed are designed to reduce network load for slow links and to reduce clutter on small screens. The adaptations are performed at a proxy, according to given user preferences for types of data and a stated feature size / screen size metric. In the developed system no means for the user to define preferences was given and hard-coded values were used. While a demonstration of the possibilities for reducing data volume by feature (GIS layer) elimination and/or coarsening of detail is given, the possibility of meeting deadlines is not discussed and network testing is left as future work. The adaptation due to screen size simply filters small features out. What treatment point features (which may be rendered as more than single pixels) get is unclear and consideration of feature density on the screen is noted as possible future work.

The authors in [1] identify the approach to map adaptation for mobile systems as novel in 2001. The approach is in line with our desire for semantically driven response to ensure user satisfaction, while also meeting resource limits. The approach to map adaptation bears similarities to our work. In particular in adapting in multiple dimensions at once and in their client-focused architecture. However, the derivation of preferences from context (or indeed any user control), which is key to our work and some detail in implementation and testing are missing.

From a tourism perspective Zipf [129] considers various ways in which maps may be adapted to meet different needs, in terms of device, user task and user culture. They examine issues of layout and presentation as well as possibilities for modifying the data. However, little information is available beyond the ideas of techniques which may be useful. As a general framework it is informative and supports our ideas, but offers little guidance from experience.

2.7.12 QML and Worth-Based QoS Negotiation

Work at HP labs by Koistinen and Seetharaman [74] describes the use of a Quality of Service Modelling Language (QML) to describe required and offered QoS parameters. The descriptions are generally at a system level: delays, availability, etc. provided with mean and n-percentile values. These specifications are used within a negotiation protocol, where a client and various servers can negotiate a contract for service delivery. In order to support the trade-off between servers "worth functions" are used to describe how the "worth" of the service varies as each parameter varies. This idea that quality is reflected in different parameters of a service and that the overall quality is not necessarily linear with the variation in the parameters is one we believe is important to be able to capture in the specification of adaptation. The technique as presented is only really applicable when comparing similar services, but the principles can be extrapolated to selection between variants of a media element.

2.7.13 Web Server QoS Management

Adaptation of web pages in response to high load on the server is addressed by Abdelzaher and Bhatti in [2]. Where the server, or network connections to the server, are subject to congestion at busy periods the web site can be provided in an adapted form to enable a larger number of successful (and timely) responses. They use a combination of various strategies for adapting content: assigning priorities to different clients, sharing spare capacity (unused parts of guarantees) from virtual servers with low load to those with higher load and providing alternative content versions of the site. The degraded content tree is modelled as a static site. How well the adaptation of sites scales to highly dynamic sites is questionable, as much of the delay and CPU load will be created by the process of creating the dynamic content.

2.7.14 Other Adaptation Systems

A thorough study of the effectiveness of image transcoding in meeting data size constraints is presented by Chandra et. al. in [33]. They find that using the quite simple approach of modifying Joint Photographic Experts Group (JPEG) format image quality typical modem bandwidths could be adapted to, however they found that in a web page the larger images were transcoded most aggressively – as they presented both the largest resource use and most potential for size saving. Whether this was a good thing is not clear. If the semantic content of the images is also considered the degree of information loss which is acceptable may vary, although their results hold where larger elements of a web page are photos. Other limitations may also need to be considered. While research such as McIlhagga’s [65] suggests that quite low JPEG qualities can be tolerated, no study was performed to verify that the transcodings being made were the most effective in this case, alternatives such as reducing image size were not tested.

The literature, including [18,107] describe systems which allow for transformation of data between media types, while still presenting the same information. While this is often more complex and certainly less subtle, than modifying image sizes and compression ratios, in many cases it is this level of change which will be needed to support effective adaptation to a broad range of contexts. This poses technical problems in: automatic translation of data between presentation types, managing change in documents in different formats and in providing a document structure that can allow this level of change. We shall not investigate the production of data here, but should aim to support a world where data may be presented in highly diverse ways.

Kendra [28], developed at City University in London, is a system where a client and one or more servers cooperate to stream audio (music) with the best possible quality given the available bandwidth, CPU power and memory. Where more than one server offers the required service the fastest responder (closest in the network) is used, transparently. Where system limitations change, the quality of the stream is modified or the buffering used changed, under the client’s control. In [27] the use of meta data to describe, amongst other things, the characteristics of resources in order to facilitate selection and transfer of media is described. Their meta data system also describes user QoS information and system description.

A software architecture for supporting application level adaptation based on network QoS is described by Witana, Fry and Antoniades in [125]. Their system provides a separation into modules of the QoS management, adaptation and resource management systems; and uses policies to control adaptation and resource allocations. The control policies are not described in detail, however the system is presented as one which addresses adaptation of individual competing streams. The possibility that multiple factors will affect the perceived quality and media degradation paths following different weighting of parameters are described. The capacity of the system to manage cooperating streams and the expressive power of the control policies is not clear.

Bolliger and Gross describe a feedback based system where a model of network bandwidth is used to cause the adaptation of data rates at an application level, through a trade-off of data quality against download time in [19].

2.7.15 Summary

The literature describes a wide range of work which moves towards the needs we have outlined. However, no work described meets the following needs:

- Offers a detailed specification of preferences and limitations from the client which arise from a wide range of context aspects. Existing work tends to concentrate on a few system issues, notably network performance, or be focused on a narrow range of devices, or require monolithic preference specifications from the user. To be truly effective contextual mediation needs to support a flexible range of system limitations, including dynamically changing values, and user needs and preferences.
- Treats whole documents with one selection process, giving consideration to the semantic content of the data being selected or modified. Without treating all elements of a document together limits on the total page presentation cannot be managed. Without treating the semantic content of the data significant bias in trade-offs is missed and variants realising data with different parameters can only be selected according to the general case, again missing a degree of sophistication.

2.8 Information Filtering to Support Non-Physical Context Awareness

The fields of Information Filtering and Information Retrieval (IR) have input to the issues in supporting the less physical side of context awareness. Representing user’s interests and needs arising from tasks has been a motivational topic in this field for some time. The main areas of interest here have been in producing user profiles, often by explicit feedback and observed behaviour; and search mechanisms over various media types, text being the longest established.

The issue of enabling access to information, through search and selection techniques, for users with different interests as a technique for mitigating the expanding diversity information systems and users and the corresponding increase in data volume is raised in [22] (Bowman et. al.).

We shall be examining situations where the search for an information source has been performed and the information is being retrieved. This takes us past the application of much of the work in IR, however the filtering techniques may well be relevant as a form of adaptation and for classifying data elements. We shall regard this field as an enabler, rather than of particular interest.

2.9 META DATA SYSTEMS

In order to select appropriate variants of media, some description of the structure of the data and the adapted variants available must be available, Kerhervé, Pons, Bochmann and Hafid [70]. This may be achieved through meta data. There are various meta data standards, supporting a wide range of applications – through description of different attributes of data. We shall examine the more relevant of these in this section, along with the sorts of information they can describe.

In addition to the meta data we describe other forms may be relevant. For instance, layout and presentation may also be specified in meta data, or in the data (as in HTML). While presentation is clearly an issue for devices with small screens we shall not be examining the wider HCI issues relating to mobile devices here. Presentation adaptation techniques are discussed by Marttila and Vuorimaa in [87]. Information such as this may well be delivered through a mechanism such as style sheets. These can then be considered as a negotiable element of the data to request.

2.9.1 Structural Information

The description of structure is a key ability for any system intended to support common media encodings. For example:

- A web page is commonly built from a frameset, a number of HTML frames, a style-sheet, a number of images and a Java applet or a data for a plug-in. The relationship between these is generally described in the data. However, the structure is not clear from the initial page request. If one chooses to omit parts of the data, it may be important to know what other data is being omitted through references which are not then loaded. The role these different files play in the page may be instructive and we shall consider this in section 2.9.3.

- A Moving Picture Experts Group (MPEG) format video stream consists of a series of key frames, describing the whole scene and intermediate frames describing differences. These intermediate frames depend on the key frame, while the key frames are independent of each other.
- A vector map data file will commonly consist of some header data and a series of features. While it may be possible to omit some of the features, they may all depend on the header data, e.g. where co-ordinates are relative to an origin. Other data, such as area fill seeds and road names, may only be meaningful in conjunction with other features.
- A streamed multi-media document may have temporal structure, where timing constraints, for instance between video and audio, are described, as in [62,123] (W3C, Wirag).
- An image may be realised by one of a set of variants. The difference between the variants will need describing and this is discussed in section 2.9.2.

[70] describes a model for meta data where a document may be multi-media or mono-media. A multi-media document will be formed from one or more mono-media elements. The mono-media elements are represented by one or more variants.

2.9.2 Objective Measurements

A description of structure enables negotiation to take into account the relationships between media elements and overall effect of multiple elements. However in order to accurately respond to resource limits and many types of user preference a description of the individual elements is required. The following characteristics are clearly useful for meeting constraints such as download deadlines in the presence of slow networks, limited screen size and limited budget; and also for reflecting general preferences, such as for up-to-date information, levels of detail, etc.:

- Size (bytes for files, bit rate for streams)
- Creation date
- Data source (author)
- Encoding format
- Lossy compression applied, e.g. JPEG quality index.
- Cost

- Map scale
- Image size
- Image colour depth
- Text language

2.9.3 Semantic Information

We noted above that to describe the parts the various elements of multimedia document play some semantic information may be useful. We shall now consider this part of the meta data.

If semantic information is to be useful, the vocabulary problem described by Furnas et. al. [56] must be addressed. That is that there must be a path between the names used by the data provider and the names used by the user. Where meta data has been extracted automatically from data which gives a standard classification of terms there may need to be a mapping between these and the meta data provider's preferred terms, for consistency. The work on the "semantic web" in the W3C [10] provides a good foundation for this inter-operation between systems.

The problem of an effective description language requires a domain of reference and a published and well understood set of commonly used terms by that provider for that domain. Within a provider / domain situation it would be reasonable to expect consistency in naming. However, it would be useful to have a mapping between this scheme and as wide a range as possible of synonyms, possibly with some note on the degree of matching.

There is also a need for flexible precision, e.g. some maps describe woodland as simply woodland, others may distinguish deciduous and evergreen, others may describe in some detail the types of trees, their age and other data. Our common classification system used in meta data should be able to encode the necessary information in a semantic type and possibly some parameters. The map which simply gives "woodland" as a description should have meta data elements with a type of "woodland", not given some spurious accuracy.

Related to the issue of vocabularies and precision is the issue of multiple semantic types. A building may contain within it a range of shops, offices and living accommodation, with a wireless phone base-station on the roof. The building might be described simply as a building, which is true but represents a significant loss in precision. To describe the building as having multiple functions would be more useful in many situations.

There are many subtleties to the issue of naming, such as disambiguation and cultural terms of reference, which have been the subject of much research. For the purposes of this work we shall assume that semantic types can be described using a system of common ontologies providing typing for domains. Naming within a domain is assumed to be unambiguous.

The use of semantic information to support adaptation is not widespread. Hori et. al. and Smith et. al. [61,106] describe systems for transcoding elements of web pages. Their system includes analysis of image content, to differentiate between adverts, textures, bullets, lines, maps, logos, navigation buttons and content. They then describe the use of these types in selecting transcoding to apply, given resource constraints such as display capability and bandwidth.

2.9.4 Quality Information

Some work chooses to describe quality of media in meta data, such as Wirag [123], this however has the failing that it assumes that all users will perceive the quality in the same way. Where the quality is determined by the context in which the media are used this is not the case. Similarly, the results of a query may be ranked by a search engine and some rating given. Whether this ranking matches that a user would give depends on many factors, such as: the search engine used, the query given and any other data on the user used in the query.

2.10 Statement of Requirements

Contextual mediation is an important technique for enabling usability of ubiquitous computing, which has received little attention in the literature so far. The techniques which would seem to be key to achieving successful have received some attention in the fields of ubiquitous or mobile computing to some extent, these are listed below.

- A model of *context*, as a set of discrete *aspects*. This requires:
  - A context description language which can be understood by a range of applications.
  - Sensor interpretation to provide up-to-date context data.
- The description of context should include a model of *dynamic resources*, which may be reserved. Resource management is required at a system level, especially of:
  - Network characteristics, such as round-trip latency and throughput.
  - Screen space.
  - Power, for batteries.

- *Meta data* to describe potential data to be used. Factors to be considered here include:
    - A description of data *element structure*, so that a highly structured data may be presented for negotiation as one request.
    - A description of multiple data *variants* which may realise each element.
    - Support for data which may have origins at multiple sources, at both an element level (merging of semantic provision) and at variant level (use of transcoding proxies).
    - A description of data properties, to enable resource management and allow selection to reflect user preferences.
    - A description of *data semantics*, to enable differential treatment of data elements and further allow selection to reflect user preferences.
    - Common language(s) for structural, properties and *semantic type descriptions*, which may be extensible to manage developing application classes.
- A mechanism to specify *selection* over structured data, where elements of data may be available in many variants. This mechanism should reflect preferences due to context, including:
    - According to the context, different semantic types of element may be treated differently.
    - A description of *preference* and *usability* limits for various parameters of variants should be provided according to the context and semantic type of the elements realised.
    - A mechanism for defining limits over managed resources, for instance deadline for download. This may only be *best-effort*, but should offer more predictable, stable behaviour than "no-effort".

2.11 SUMMARY

We are interested in one form of response to context, contextual mediation. We have discussed the current (2002) state of computing devices in ubiquitous computing and identified a range of issues that related to networks, power and usability, which can be expected to persist for some time. For those interested in the wider field of research in ubiquitous computing, context awareness and mobile systems surveys and key papers include [4,23,26,31,34,41,55,66,103,120]. A solution to these problems ought to be general enough to encompass technological development for some time. We have set out what we believe to be key requirements for contextual modelling.

The prior work in the area includes work in the areas of the web, multimedia and data description, network modelling, context sensing, usability studies and Geographic Information Systems (GIS). A general approach to contextual mediation has to consider a wide range of effects and different responses to them. Necessarily we treat a range of these issues in this thesis. In particular modelling the context and data to be mediated required work in order to support the flexible approach to mediation we have outlined.

CHAPTER 3 A MOBILE ADAPTIVE APPLICATION

In this chapter we shall summarise the objectives of our work; introduce some key concepts; introduce the mobile use of map data, which we use as a motivating example throughout the thesis; and give an overview of the architecture of our solution. Further software engineering notes can be found in appendix 2 and appendix 4.

3.1 THESIS OBJECTIVES

There are clearly many aspects to producing effective context-aware systems. We shall focus on a model which can encode a broad range of responses to context in order to support contextual mediation. The facility should be portable across application domains.

To support this we shall develop:

- A demonstrator application which should have the potential to run on a wide variety of systems. This potential will be expressed both in system dependence issues and in user interface design.
- A model which can encode our general concept of context.
- A meta data system to demonstrate how meta data may support response to context.
- A system resource management system for networks, which can operate without any external support, but might abstract underlying systems such as Resource Reservation Protocol (RSVP). User interfaces which provide feedback to the user to manage their expectations and inform the user of activity can also make use of these resource managers.
- A location service as input to our application, which shall also serve to highlight the difference between context triggered actions and contextual mediation.

We have chosen to concentrate on applications with a discrete request / response style of interaction with a limited number of servers at each request phase, e.g. using HTTP. These "web-style" applications are an important class and have not received the same attention from the QoS community as have streaming media applications. However, the solution should be broadly applicable to streaming media applications, although there are problems these encounter (such as jitter, loss and other small duration variations) which we do not examine here.

While the specification and meta data languages described in this thesis do not follow exactly those described in the literature, encoding in XML and use of similar representational capabilities should ensure potential for migration to interwork with standards. At the time of system design not all of the relevant standards were finalised. In many cases their representations did not support all our interests. The task of implementing a simple XML encoding to demonstrate the concepts was not prohibitively complex, or restrictive in terms of future development and so was undertaken.

It was clear from the outset that our interests relate to other aspects of context aware research which we shall not address except in passing, such as:

- Work in HCI in adapting to context.
- Techniques for sensing context.
- Location awareness (a description of the location service used with our application can be found in appendix 1).
- Techniques for adapting, summarising or searching for media.

3.2 MOBILE MAP USE - AN EXEMPLAR APPLICATION

One clearly interesting class of applications in mobile computing are those which react to location, or provide information relevant to a user's location. There is a large body of location correlated data available which may support various tasks, can be displayed on many different devices and adapted to other aspects of context.

Data, such as maps, have some content which is of general interest, whilst other content is more specialised. When navigating in a car details of roads and traffic are important. A hiker, on the other hand, may consider that footpaths, hills and rivers are more significant. Delivery drivers and emergency services will probably know the general geography of an area, but may want to identify detail such as house numbers and temporary diversions. Other information, such as administrative boundaries and spot heights, would clutter the map on a small screen and so be undesirable.

While most map information is static in nature, feature updates, e.g. after building works, can be propagated by on-demand delivery in a more transparent manner than for statically stored maps, e.g. on CD. The potential volume of map data is also high, making static storage on limited devices a problem. Highly dynamic information, such as weather forecasts, traffic conditions or local entertainment information could be provided over the geographical features through overlays or hyperlinking. Just-in-time delivery of this information is usually preferable. However, the wide range of data may easily lead to information overload: for the display, network, or for the user's ability to digest it.

Most map applications provide either a standard view of the map, or completely expose feature selection to the user. A typical Geographical Information Systems (GIS) package can be configured to display any combination of specified features, however this generally requires some effort and sophistication on the part of the user or the use of pre-prepared queries. In the case of applications of map data which have mass appeal or involve the use of restricted devices, the effort or user interface required to manage this level of control may not be acceptable. In addition many GIS systems come from a background of desktop use and/or very specific data requirements. The provision of flexibility to varying context and resource limits is not a common feature of map display applications today – in most cases the requirement for feature layers in GIS packages in binary, while we are interested in providing selections which adapt to the context.

GIS applications may seem similar in many ways to our interests, however the approach here is generally different. GIS systems tend to require high specification machines (although mobile GIS systems are available), while our interest is in adapting to the device available. GIS applications tend to assume a high level of knowledge and involve the user in the configuration of the display of information precisely to their needs, whilst our interest is in a system of automatic adaptation which assumes rather less about the user's understanding of cartography and the time they are willing to spend configuring the system. We are interested in a ubiquitous computing approach where the computer will not generally be the focus of attention, but a supporting tool. The application we describe offers the presentation of information in a wide range of contexts. We have not investigated the use of analysis of the map data, or facilities to create maps – although these are areas with a potential interest in mobile computing and may be a subject for future research.

Contextual mediation is not the only technique which should be applied in seeking to enable context awareness in the presentation of map data. An overview of the process of providing custom maps for tourists is given in [129]. Many of the steps described can be at least partly provided or used by our techniques, e.g. spatial focus, reflection of task, cultural influences, shape simplification and landmark selection. However there are other aspects which we leave to the user or other aspects of context awareness, e.g. orientation, area displayed, layout and drawing styles.

There are various terms we use in our work to describe the structured multimedia data we are working with, drawing from use in the literature such as Kerhervé [70]. However, some of these suffer from many uses in computer science, so we shall briefly define the key terms below. The relationships between these terms are also illustrated in figure 3-1.

- A *document* is a unit of presentation for structured multimedia information. A document may be considered to be a collection of one or more *elements*.
- An *element* is a generalised part of the data which fulfils a specific role in a document. In a map example this may correspond to a *feature*, e.g., the M1 motorway, or river Thames. Elements are described by a *type* which defines its semantic content, e.g., road, river or building. Motorways, major-roads and minor-roads are all sub-types of road. Elements may have *parameters*, such as the area covered by a feature.
- An element may be represented by multiple *variants*. For example, a map may contain representations of the M1 motorway surveyed at 1:10000, 1:50000 and 1:100000 relating to lesser inclusion of fine detail on small bends, etc. Similarly a picture can have variants relating to different resolution. A variant has an *encoding format* which describes its syntactic encoding, in our work we use MIME-type descriptions. A variant is described by *attributes* which may be general, such as size (in bytes); or encoding format specific, such as survey scale.
- An element may *contain* other elements, e.g., to add labels to the representation of a road, or a picture in a web page contained by the HTML which refers to it. Being contained, these elements rely on the containing element to be present in order that they may be displayed.
- A variant may *contain* other variants indicating a dependency in the data which is required for realising the contained variant, e.g. data which requires but may be separated from its header.
- Elements may be aggregated into *groups*, e.g. major roads in an area, rather than describing all elements individually.

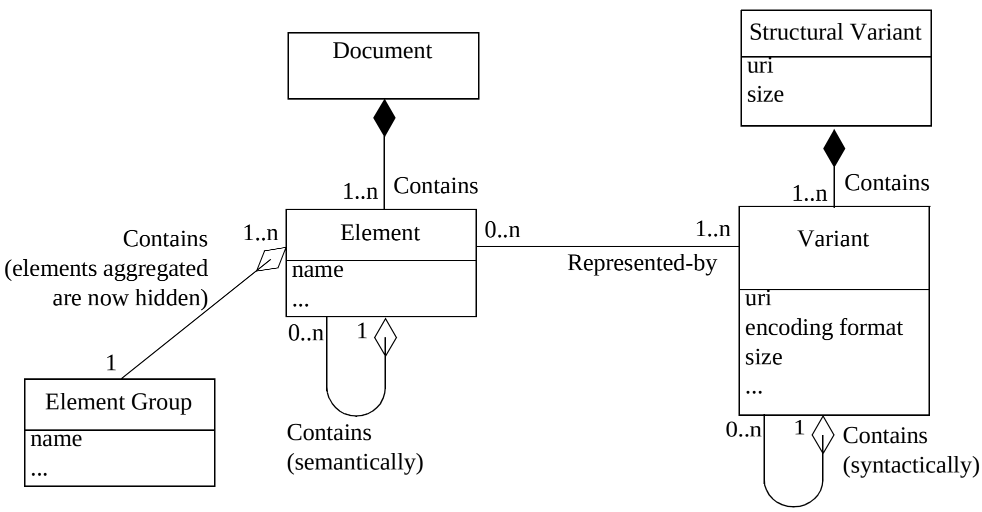


Figure 3-1: Class Model for Representation of Data

In general, new elements are introduced to contain semantically different information and variants used to offer different versions of the data. Variants are units of data which may be individually requested. There need not be a 1:1 relationship between element instances and variants.

Current solutions for enabling selection of specific information according to contextual or system related restrictions suffer from significant limitations, particularly for the class of applications we are studying. Where structured data are being used, selection according to the data's encoding format is unlikely to fully capture the user's needs. In web browsing it may be sufficient to say that one dislikes images. However, for a vector maps format selection based on encoding format does not perform a meaningful selection. Where a large amount of information is contained within one encoding format it is desirable to be able to express preferences related to the *semantics* of the information, i.e. display roads but not contour lines where both may have the same encoding format.

As the range of device capability and tasks expands, it is undesirable for content providers to be obliged to provide many different versions of their content, with each adjusted to a small range of devices or users. However, the user is best served by data which is finely adjusted to their needs. A mechanism for content negotiation which allows for general specification of requirements is needed. Many other proposals for media selection in response to context or resource constraints are applied to each element of the data separately, e.g. HTTP [59]. This per-element selection limits the range of possible trade-offs and does not allow for consideration of the interaction of the various data elements within the whole document presentation.

The network link capacity may also restrict the amount of data which can be transferred in a reasonable time, leading to long and variable delays between a request for a map tile or web page and its display. For many tasks timely data provision is key. A data selection mechanism should therefore enable deadlines to be met by restricting the data selected. Deadlines may be due to a number of factors, such as the user's patience and the need for timely display due to movement. The first case is a simple one experienced for many applications and often cited as a source of irritation in web browsing. The second is most relevant during mobile use where a map segment must be displayed in advance of needing it, e.g. in a car, a navigator prefers to understand the layout of a junction in advance of reaching it. Conversely, old data may give out-of-date information as congestion can change quite rapidly on roads. Deadline-based selection may be achieved in a similar manner to the selection according to display capability and the user's requirements and must be performed in conjunction with user preferences. For more capable devices the network connection may be the most significant limiting factor in selecting data.

We are addressing the use of map data, both general and specialised, in specific tasks. The restriction of the domain gives the user a clear means of identifying what information is relevant. The restriction of the data allows standardised descriptions of the semantics of the content to be developed. Similar techniques may be applied to less restricted domains once suitable and commonly understood task and data descriptions are developed.

3.3 An Architecture to Support Contextual Mediation

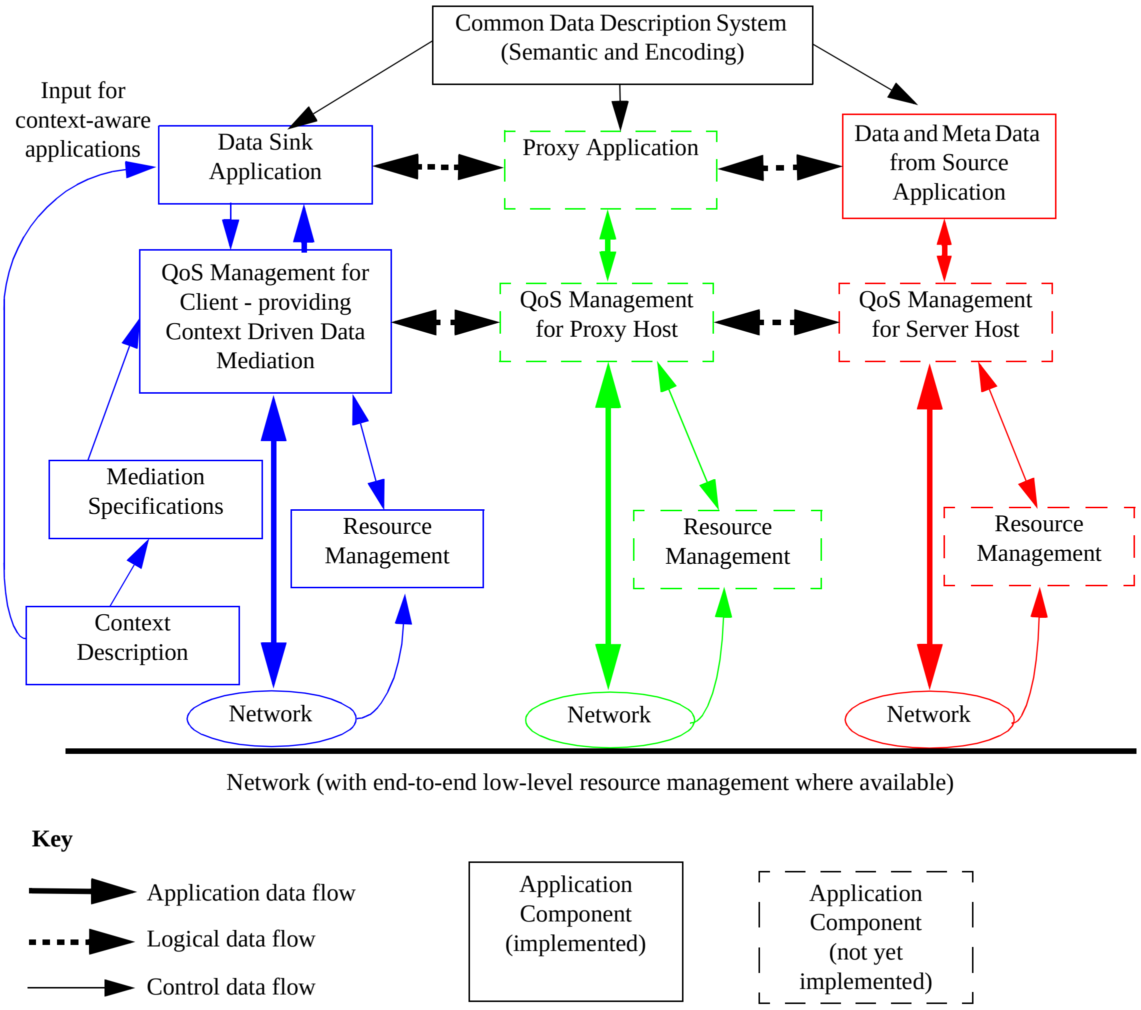


Figure 3-2: System Architecture

The overall architecture with which we are working is shown in figure 3-2. Applications interface to the outside world through a data mediation system, which bases its behaviour on the user's context and resource availability. In server applications this provides application level QoS management functionality, at the client this provides a wider context aware mediation. Applications (and their QoS management subsystems) interact with each other over networks, these may include within them active nodes providing transcoding, caching and other proxy services. These active nodes in turn interface to their own local resource management.

Resource management is provided locally at each node, enabling constraints on usage through admission control, based on predicting future resource availability. Where the resource is local (such as CPU time, memory, disk throughput) this may be a highly accurate management system. Where the resource is shared with nodes and applications which do not co-operate with this control then the resource management cannot provide a guaranteed level of service. However, if one can assume fairly responsive monitoring and accept the weakness of the admission control based on it, this provides an improvement over no management. Where underlying resource management facilities exist, the local resource management should layer on top of this. So, where end-to-end protocols such as RSVP are available, local network throughput reservation may mirror reservations established through RSVP.

Where many active nodes may exist the QoS management system is then responsible for aggregating the data and selecting between these wider options. Advanced active elements may provide services such as dynamic hyperlinking [100], where different active nodes provide links to different sorts of information. Traffic monitoring services may add congestion information to road features, by modifying the feature data. These additional data services may stimulate new applications for mobile computing. This wide range of data may easily lead to information overload: for the display, for the volume of data to be transferred over the network, or for the user's ability to digest it. Dynamic data services may also compete. A selection mechanism must therefore be able to differentiate between interesting and unnecessary data (and many degrees in between). This must be balanced with timely delivery, screen space availability, cost, etc. Different users in different situations will have widely varying preferences. Successful management of these issues, in a predictable and unobtrusive manner will be key to the success of advanced applications.

Our focus has been on client side applications and, as figure 3-2 shows, most of our implementation has been on the client side. However, while our work addresses data mediation at the client to enhance the user's experience, similar techniques may be applied at servers and proxies to assist them in managing load [2]. These adaptations will interact with the client side management. A client may find its choices for adaptation limited if a server is limiting the data it will serve, or a transcoding proxy is overloaded and not offering a full set of services. The choice of client-side mediation, as in Zenel and Duchamp [127], allows us to reflect users' needs, while maintaining privacy with regard to *why* choices are made.

In order for the data mediation process to be possible, context awareness and specifications of required behaviour which respond to context are needed. The servers and proxies will describe data offered through meta data. This is interpreted through shared knowledge of the meta data encoding and the descriptions it provides. For our approach a common semantic type system which allows the meta data to describe properties of the data beyond the attributes due to its encoding is important.

3.4 INTERACTION WITHIN THE ARCHITECTURE

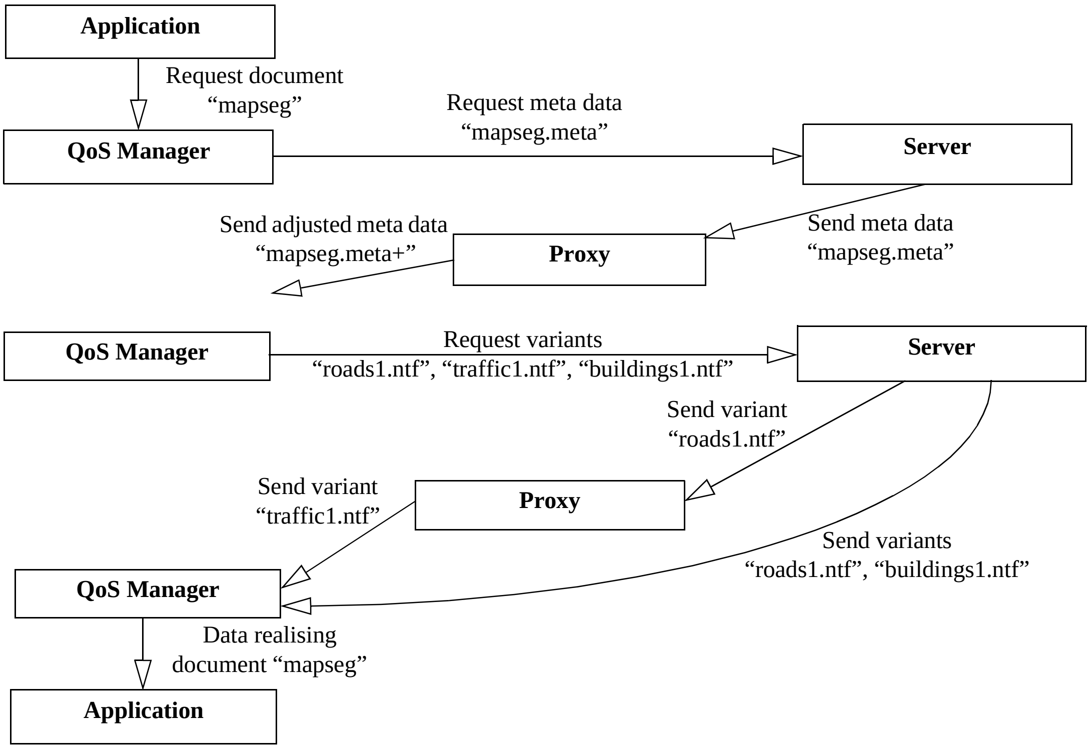


Figure 3-3: Illustration of Interaction with Meta Data

In summary the process of retrieving data, illustrated in figure 3-3, is:

a. The application makes a request to the QoS manager (or mediator), which then requests the meta data.

b. The server's response describes the various elements of the data and the variants of those elements it has to offer. The server, when responding with meta data, may make a provisional reservation of resources based on the variants offered. In cases of high load it may adjust its offer. Any server side QoS manager is omitted from this diagram for clarity, however it would mediate in the data processing in a similar manner to the client side manager.

c. Any intermediate nodes (proxies) append to the meta data and pass it on, this addition advertises their capability to modify the data (e.g. add traffic information, compress data, or text language translation).

d. The client receives the meta data and makes a selection amongst the variants whilst reserving sufficient resources locally to support the request.

e. The request is then passed back, the proxies and any low-level resource management capable network elements also making reservations. The server then adjusts its reservation to reflect the chosen variants.

f. The data requested is returned to the client, passing through any proxies whose services were selected.

3.5 SUMMARY

We stated the needs we have identified for contextual mediation and those parts of the problem which we intend to address. Our focus is on supporting mediation to reflect a wide (and not predefined) range of aspects of context. The key part of this is developing a system of specification which reflect needs and limitations arising from context. To support this we shall also investigate processes of selection, models of context including resource management and meta data. We have given an overview of our map adaptation exemplar application and a general motivation for applying contextual mediation to this. Our main exemplar shall be map data, although examples in terms of web data shall also be presented. A high level structure of the system these applications fit within and its interactions was presented. The various components of this system shall be described in detail in the following chapters. We have introduced the key concepts of *document*, *element*, *variant* and *type* from our meta data and related them to features and data in the map application.

CHAPTER 4 DATA AND META DATA SUPPORTING MEDIATION

In the previous chapter we introduced our model of data as a structured collection of *elements* realised by a structured collection of *variants* and also described our map application. In this chapter we shall explore the characteristics of the map data we seek to use, how these may be modified and examine how it may be subjected to mediation. Data for web pages is also examined and described in similar terms. We describe the meta data we use to enable mediation and the map and meta data server we developed to demonstrate our ideas. Provision for supporting dynamic data is also addressed.

4.1 VECTOR MAP DATA

There are many encodings for geographic data. The Ordnance Survey has (over the period this thesis was being worked on) provided data to us in: NTF v2.0 (BS 7567), DXF (AutoCAD Release 12), ESRI Shape and GML[95] (an XML DTD) formats. Each of these formats encodes data in a different manner and structures its encoding of semantic data differently. These differences reflect common practice and product needs at the time the format standard was laid down, making it more or less suitable in various application domains. It is common for mapping agencies to have come from producing paper based maps and later supporting GIS applications – typically for planning or record keeping. In keeping with this heritage the formats tend to describe data with some expectations of size when rendered (especially the paper oriented formats, such as NTF) and with the expectation that the system being used is relatively highly specified. While many of the common map data formats reflect cartography and GIS history, the data is often stored in databases. The ability to re-encode data and extract meta data is now well established.

We chose to work with NTF data, as this was acceptably easy to parse, extract meta data from, split according to features and modify. The Ordnance Survey provided us with complete coverage of Great Britain in two distributions (Meridian and Strategi) in this format. Its down-side was that we had to produce our own rendering software, as no freely available Java tools could be found for this. DXF had many similar benefits, but required significantly larger files for little perceived benefit to our application. ESRI Shape data was experimented with as free rendering tools were available, but these did not prove easy to use and splitting or modifying the data was also hard. GML was quickly rejected as the available parsers, which used DOM, had memory requirements which made them unsuitable for our desktop systems and completely impractical for mobile devices. A brief example of NTF can be found in appendix 3.4.

Maps in vector format are somewhat larger, in terms of data stored, than the same data rendered as a raster image. Table 4-1 and figure 4-1 illustrate a comparison of vector and raster map data size, for urban and rural data. The vector data is OS Meridian 2, NTF format data, 1:250 - 1:250000 depending on feature type and region. The raster, of the same data, is in JPEG format, with an image size of 600*600 pixels and a quality index of 0.75. Both are shown as compressed (using GNU ZIP (GZip)) and raw. The urban data is the average of 10 10*10km tiles of London and its suburbs. The rural data is the average of 10 10*10km tiles of Cambridgeshire and Hampshire. The number of features column is extracted from the NTF data and is intended to give an intuitive feel for the relative density of information in each map.

Map Type	Number of Features / km^2	Data Size (kBytes) / km^2	Average Data Size (Bytes) / feature / km^2	GZipped Data Size (kBytes) / km^2	Average GZipped Data Size (Bytes) / feature / km^2
Vector (NTF), Urban Area	123.1	19.49	162.1	5.323	44.28
Raster (JPEG), Urban Area	123.1	1.963	16.33	1.959	16.30
Vector (NTF), Rural Area	8.313	1.408	173.4	0.424	52.35
Raster (JPEG), Rural Area	8.313	0.825	101.7	0.796	98.02

Table 4-1: Typical Characteristics of Map Data in Vector and Raster Formats

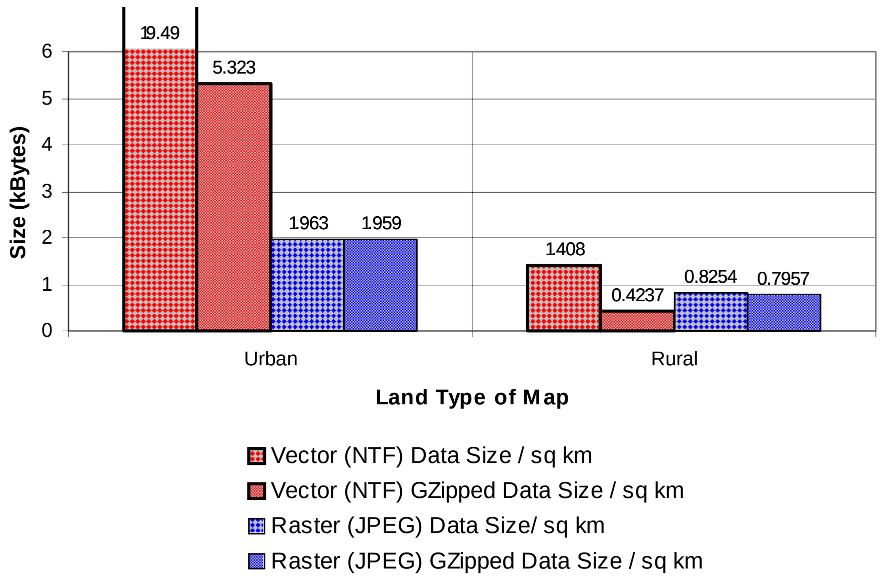


Figure 4-1: Comparison of Vector and Raster Data Sizes

We see in table 4-1 that the JPEG image is substantially smaller than the vector data, although under compression the relative sizes become much closer. Note also that both data formats exhibit a substantial overhead. In the case of NTF we know this to be approximately constant, being due to headers, although there is an apparent 9% difference in the bytes / feature between urban and rural data. In the rather sparse raster images of the rural map the data size for the JPEG does not drop anywhere near as far in size / feature as the vector data does (JPEG feature data is 58% of the size of NTF data for rural data, but 10% for urban data). The NTF format is a text encoding, which is therefore particularly amenable to GZip's lossless compression, while GZip has little effect on the already compressed JPEG. For compressed rural data the NTF in fact takes fewer bytes than the JPEG. From these examples one can conclude that for urban areas NTF (and in fact most vector map formats) are quite inefficient compared to moderately compressed rasters. As the map image becomes sparser the benefit decreases and for rural data the vector format has a size advantage. See figure 4-2 for examples of urban and rural maps.

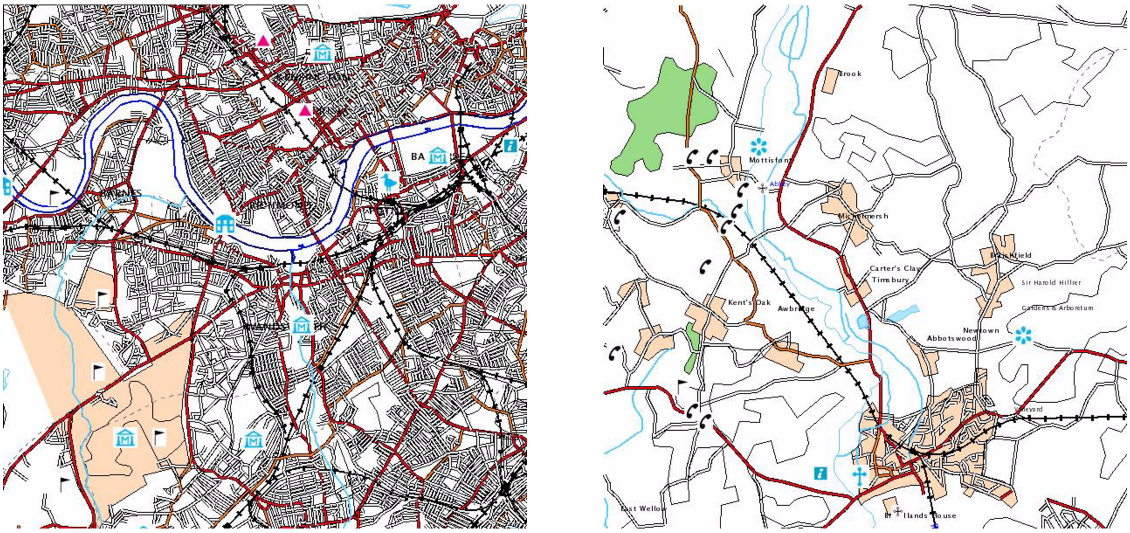


Figure 4-2: Examples of Urban- (left) and Rural- (right) Area Maps

The 600*600 map these examples were created from is approaching the limits of usability for the urban data, however in order to provide a map with a level of detail appropriate to the device a raster needs to be regenerated to include different features – which removes the benefits of simplicity and compactness that are benefits of raster data. With a vector map one can choose to include or omit any given feature and to change the rendering of any given feature as necessary. In the example given, one might imagine that much of the fine detail here could be omitted without overly affecting usability, this may be by omitting features or by zooming in. For vector data a zoom can be arbitrary and is effectively a form of feature omission. For raster data a zoom typically involves generating or storing different images and, as we have seen, is not as efficient a way of dropping data size as feature loss in vector data. Feature omission is very much harder to achieve from a raster source without incurring the overhead of a new image many times. For example, consider figure 4-3. Here we compare vector and raster data with approximately the same size (GZipped vector data). The size reduction for the raster is by scaling the image to fewer pixels, then returning it to its original size for display, coupled with increasing JPEG compression. The size reduction for the vector data is achieved through feature selection. While the vector data contains less information than the raster the data it does present is far more readable than the JPEG, where small features blur into each other.

Even if a client system will not display maps in vector format, in order to provide suitable rendering for the context vector maps are clearly preferable as a data source. Another benefit of vector data is if multiple data sources are used, the data can be combined with no special effort, for instance if both sources contain data which should be rendered in the background and in the foreground, or the different sources contain maps at different scales. Where the client cannot render vector maps a proxy may be used to render the data and provide a dynamic raster image. The overheads associated with this use of a proxy may be more satisfactory for some classes of client.

Our work does not depend on specific data encodings, nor does it propose encodings or enhancements which are more suitable for certain purposes. Raster maps and vector maps each have their advantages. Where throughput is limited raster maps may be more appropriate in some circumstances. However, we have shown that the size disadvantage in vector data is of a manageable order and is compensated for in many cases by its flexibility. We developed the remainder of our approach using vector data, rendered at the client. Where download times and throughputs are discussed in relation to data provided, the flexibility described is more important than absolute times – which are determined in a large part by the data format chosen rather than the approach in principle.

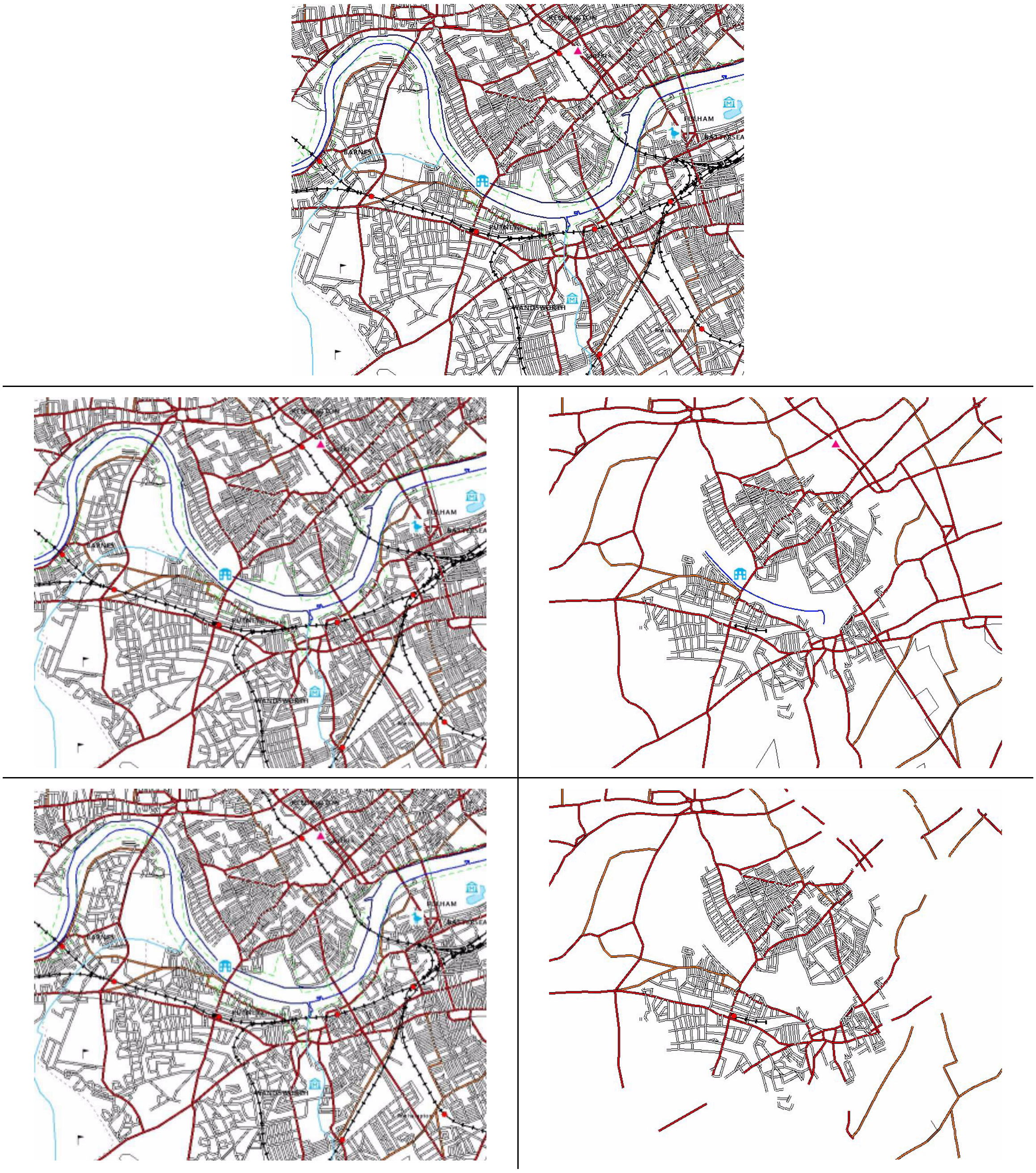


Figure 4-3: Maps Illustrating Data Size Reduction
Top – Full Selection approx. 344kBytes / 790*647 pixel JPEG (q=0.8), 265kBytes
Left Middle – JPEG (q=0.75) scaled to 521 * 427 pixels, 155kBytes
Right Middle – Vector scaled by selection, approx. 163kBytes
Left Bottom – JPEG (q=0.6) scaled to 395 * 324 pixels, 118kBytes
Right Bottom – Vector scaled by selection, approx. 108kBytes

4.1.1 Level of Detail in Vector Data

Vector data is generally encoded with a specific level of detail. For maps this usually corresponds to a scale for a paper map. The level of detail encoded depends firstly on the detail captured by the surveyor and secondly on the detail provided in the encoding, which may be equal to or less than the surveyed data. The encoding corresponds to the scale of a paper map by adjusting the smallest feature which can be drawn or seen on the map. For instance, where a road bends the smallest bend in the encoding will correspond to the smallest change in the line which the user would see. The encoding can drop detail in two ways: firstly, by not recording the least significant digits of the location. This is quite a coarse approach, but can save one digit of encoded location for every point specified. The second approach is to remove intermediate points from lines which encode feature changes smaller than a given threshold from the last point described. This is equivalent to changing the scale on the map drawn. We use this approach in generating variants of map features and to offer reduced quality data for reducing data size.

For instance, in a 1km^2 area of London and for a more rural area, we find all elements and their variants. The variants offered have sizes as presented in table 4-2. As minor detail in features is omitted the size of the data falls. Where no difference was found no extra variant is offered for that element, in this case the next higher level of detail available is used in calculating the size. The first and last point on a line are always retained by our scaling algorithm. We see that there is little benefit to be found in choosing data of lower scale if one is trying to reduce the data size in our example data set. This may be due to two factors: firstly the algorithm used was not the most effective at reducing size; secondly the data set we used was typically composed of many small features, such as road segments. Road data was presented as a new feature at each road junction. With further processing and an improved algorithm the reduction in data saved may be improved. A further description of adapting vector map data for use in mobile systems and appropriate adaptations can be found in [1].

Urban / Rural	Scale	Total Size (Bytes) of Variants of Roads at this Scale	% of Highest Detail Variants' Size
Urban	2500	123220	100.0%
Urban	25000	121924	98.9%
Urban	250000	119391	96.9%
Rural	2500	60345	100.0%
Rural	25000	59514	98.6%
Rural	250000	57321	95.0%

Table 4-2: Example of Data Size Change Due to Omission of Feature Detail

There are other ways to reduce the size of the NTF data we are using, for instance to record differences between points rather than the full difference from the tile origin for each point. However, we have not pursued this further as our interest is not in encoding map data or compression. Our work remains valid for a wide range of data sizes, as the potential data set being addressed is intended to be flexible.

4.2 WEB DATA

Another form of data which is of keen interest are web pages. The data volume for a page varies widely: from simple text-based sites, to highly graphical sites. The latency involved in highly dynamic sites vs. simpler, more static sites is also noticeable. A typical web page's structure is illustrated in figure 4-4.

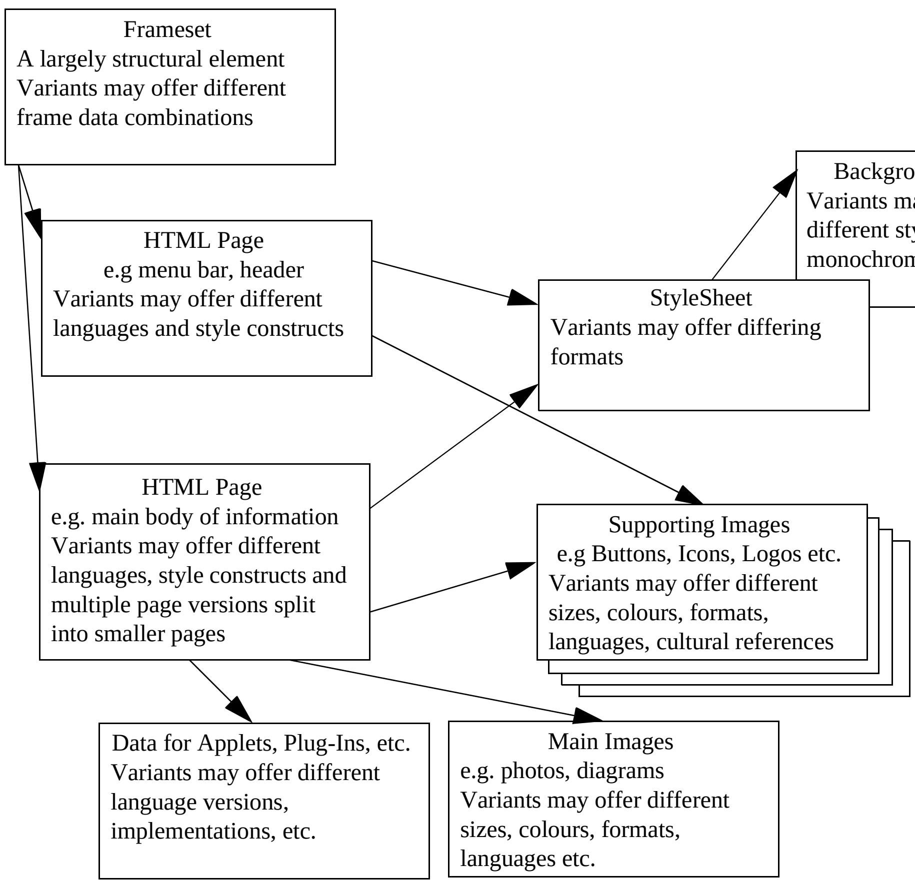


Figure 4-4: Typical Web Page Structure

Web pages are rather coarser than map data, consisting of only ten to one hundred elements in a typical page. Where pages have higher element numbers many are typically small images used for layout, which offer little content or scope for adaptation and in any case should be treated together. Within the structure of the page the various elements will have different semantics such as:

- Icon
- Photo
- Descriptive text

and the variants will have highly varying formats such as:

- HTML
- GIF
- JPEG
- Java applet

In many cases variants may be produced, modified in different ways, such as:

- Different languages for text.
- Different image formats, sizes, colour depths, compression levels on images, sound and video.
- Different encodings, degrees of interaction, etc. in applets and plug-in data.

Again we see that selection according to a range of data attributes of different elements of the data, which encode different semantic parts of the document, is a practical proposition. Providing this facility is not a focus of our research but one which we do not rule out when making design decisions. We return to a case study of this application in chapter 10.

4.3 META DATA

As discussed there is a need for users to select a subset of broadly relevant and technically acceptable information to be displayed. Current solutions suffer from significant limitations, particularly for the class of applications we are studying. Where structured data of one (or a few) format are being, used selection according to the data’s format is unlikely to fully capture the user’s needs. In web browsing it may be sufficient to say that one dislikes images. However where all data are in one format selection based on format is not meaningful. In this case it is desirable to be able to express preferences related to the *semantics* of the information, e.g. display roads but not contour lines where both are given as ESRI Shape data. We address this issue through the use of *elements*, which are instances of a *type,* which defines its semantic content. The Dublin Core meta data [45] also uses the term type to describe a similar concept.

We illustrate the class structure we use for maps in figure 4-5, using the Unified Modelling Language (UML). An *Atlas* is described as being a collection (aggregation) of *Map Tile*, both of which are sub-types of *Document*. The *Map Tile* is an aggregation of *Feature*, of various types. The types describes the semantic type of the element. We show types which have been specialized in the case of *Road*. Clearly other feature types would also sub-type *Feature*. This diagram is abstract and applies over the whole of the atlas. We illustrate an instantiation for a specific map tile in figure 4-6. Each element may be represented by one or more variants, we illustrate variants which offer alternative representations of the same semantic information. The *Variant* type is shown as having a specialisation, *Map Variant*, which is capable of describing information specific to maps. These relationships are close to those described in [70]. Variants may be contained where there is a dependency in the data, in the case of NTF map data we have header and footer data, which gives base-coordinates and other information.

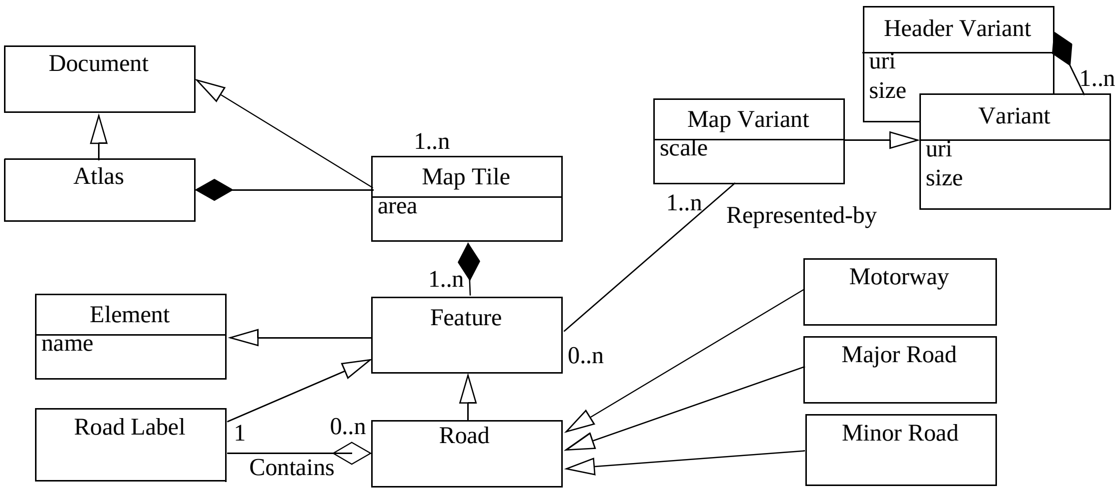


Figure 4-5: Class Model for Representation of Maps

Document authors (or their agents) describe the structure of the document's elements and their variants; and the objectively measurable attributes of the variants. Each element of the document can be represented by one of a set of variants and/or refer to a more specialised element. Variants represent data with a similar purpose, but differing characteristics. For instance, features may be presented with different minimum feature sizes, corresponding to the level of detail presented at different map scales in the paper world. A change in scale can affect both feature inclusion and detail in the description of features. For instance, roads may be presented in variants which show progressively more detail such as small bends. Variants may also be used to encode data in different formats, e.g. maps may be presented using vectors or rasters. In addition meta data will describe system-level properties, such as size (bytes).

Meta data describes the instances and relationships such as those shown in figure 4-6, of the class model, as described in figure 4-5. Figure 4-6 illustrates four variants (roads1v, mways1v, mways2v, m1v), each of which represents the “M1” motorway and in all but one case other feature instances as well. If one were only interested in the “M1” motorway, one of these variants would be selected for retrieval from the server and any possible elimination of unwanted features made locally. Instances of elements may be anonymous, such as for “Minor Road” and “Road Label”, where the name is “*”. This notation describes a group element for all instances of the type, without the need to explicitly identify each one. This is useful where there is no clear identity for many small features or in order to trade smaller meta data for expressive power.

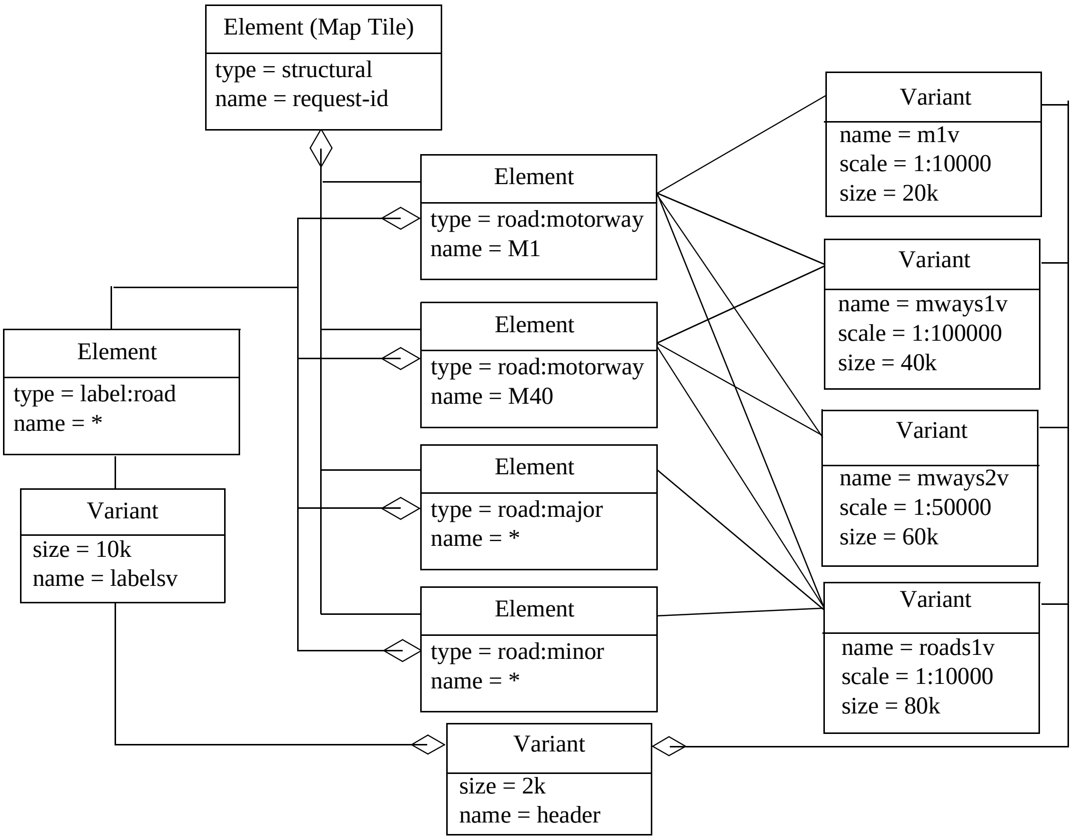


Figure 4-6: Instantiation of Document, Elements, Element Groups and Variants for a Map Tile

The variants may be pre-computed files, e.g. all motorways, or dynamically selected data, such as a particular road from a database. The trade-offs involved in these retrieval mechanisms are not described here. Our techniques can accommodate either, although the overhead in describing very fine grained features may become excessive. Subjective meta-data requiring significant human effort to author is not always popular with content creators and is technically hard to produce where content is dynamically generated. By concentrating on automatically derivable meta data, its use becomes less of a burden to content providers. If a proxy is offering translations on data then the meta data must be modified to reflect this, e.g. a proxy which performs a standard translation on images for PDAs would describe the results of the application of its function over each image. The modified meta data would describe the expected attributes of the generated image, as the proxy does not yet have the image to act on. Caching transcoders may offer greater accuracy and faster delivery.

Web pages may be represented in a similar way. The structure of frames and included images etc. is supported by the containment of elements in variants. A “no-frames” version of the page may be described by the page element being represented either by a frame-set, or by a single-frame HTML file. Different language versions may be described as different variants of each element.

We shall not define the format of the meta data in detail here (see appendix 2.2 for more detail). The element and variant structure and parameters of each, including element type, are encoded in XML. There are a basic set of encodings which are extended to represent meta data for a given domain, such as maps. The basic encoding primitives are:

- `<e n="ename">` – to define an element with a name, “ename”. May be nested.
- `<eg n="egname" eno="n">` – to define an anonymous group of “n” elements with a name “egname”.
- `<t n="tname">` – (possibly multiple) within an element, to show that the element has a type of “tname”.
- `<v n="vname">` – to define a variant with a name, “vname”. May be nested.
- `<r n="ename">` – (possibly multiple) within a variant, to show that the variant realises element “ename”.
- `<a sb="bsize" price="price" date="cdate"/>` – within a variant (not all attributes are required) to describe attributes of the variant, in this case size (in bytes), price and creation date.

These basic structures are extended for maps as follows:

- `<area minX="minX" minY="miny" maxX="maxx" maxY="maxy"/>` – within an element to define a bounding box for it.
- `<a scale="scale">` – the scale attribute for the description of a variant indicates 1:scale.

For example, the data described in figure 4-6 would be encoded as shown below:


```
<metatypedomain="map">
<e n="442100106100443100107100-header"><t n="system:structural"/>
<e n="M1"><t n="road:motorway"/><area minX="442460" minY="104808"
maxX="446972" maxY="108384"/>
<eg n="labelgroup1" eno="28"><t n="road:labels"/><area minX="441421"
minY="104800" maxX="447000" maxY="108400"/></eg>
</e>
  <e n="M40"><t n="road:motorway"/><area minX="441983" minY="107009"
  maxX="442590" maxY="107568"/>
    <eg n="labelgroup1"/>
  </e>
  <eg n="roadgroup1" eno="14"><t n="road:major-road"/><area minX="441988"
  minY="106649" maxX="442565" maxY="107095"/>
    <eg n="labelgroup1"/>
  </eg>
  <eg n="roadgroup2" eno="12"><t n="road:minor-road"/><area minX="441438"
  minY="106486" maxX="442027" maxY="107258"/>
    <eg n="labelgroup1"/>
  </eg>
</e>
<v n="header"><a sb="4404"/>
  <v n="74dc0593"><r n="M1"/><a sb="20000" scale="10000" price="18"
  date="2000-275-0:0"/>
    <v n="16bd059f"><r n="labelgroup1"/><a sb="10000" scale="10000" price="36"
    date="2000-275-0:0"/></v>
  </v>
  <v n="6c9b4588"><r n="M1"/><r n="M40"/><a sb="40000" scale="100000"
  price="48" date="2000-275-0:0"/>
    <v n="16bd059f"/>
  </v>
  <v n="f13459d"><r n="M1"/><r n="M40"/><a sb="60000" scale="50000" price="42"
  date="2000-275-0:0"/>
    <v n="16bd059f"/>
  </v>
  <v n="fd7c59d"><r n="M1"/><r n="M40"/><r n="roadgroup1"/><r n="roadgroup2"/
  ><a sb="80000" scale="10000" price="42" date="2000-275-0:0"/>
    <v n="16bd059f"/>
  </v>
</v>
</meta>
```


The type definitions at all nodes processing the meta data need to be consistent. We have an XML encoding which allows us to define extensible type systems for specific domains. See appendix 2.1 for technical details. The type systems allow sub-typing. An example of the type system for roads is shown below:


```
<type-domain name="map">
  <type name="road">
    <type name="a-road"/>
    <type name="b-road"/>
    <type name="dual-carriageway"/>
    <type name="under-construction"/>
    <type name="motorway"/>
    <type name="minor-road"/>
    <type name="name-number"/>
  </type>
</type-domain>
```

Extracting typing encoded within data is also possible, e.g. NTF uses one of a set of well defined feature codes within a feature definition. We choose to allow mappings between multiple types in our definition and multiple types in the data's definition, for maximum flexibility. These relationships are defined in a simple XML format, an example of which is given below:

```
<type-mapping domain="map" foreignformat="ntf" foreignfield="feature-code">
  <map>
    <type name="building:railway-station"/>
    <type name="railway"/>
    <foreign name="6155"/>
    <foreign name="6551"/>
    <foreign name="5520"/>
  </map>
</type-mapping>
```

Here we see the mapping between the various *feature codes* for a railway station in NTF: 6155 (Meridian railway station), 6551 (Meridian station name), 5520 (Strategi railway station); and the type system definitions “building:railway-station” and “railway” in the type domain “map”.

4.3.1 Impact of Meta Data

It is important to consider the impact of retrieving the meta data on the system. The transfer of this data will take time, which has to be considered against the benefit of using it.

In figure 4-7 we illustrate the impact of meta data, in terms of data volume, in comparison to the vector map data which it represents. The vector map data sizes are taken from the same 20 $10*10km^2$ tiles as in section 4.1, the meta data sizes taken from the map server we have developed, describing those tiles. We can see that while there is a clear overhead it is not prohibitive: 13% for urban data and 43% for rural data, which already presents less of a resource problem. In addition we find that the overhead it presents is more stable than the data itself. This is due to the grouping of features and the fact that a line feature in a map is encoded as a sequence of points, while its meta data is the same size as the data for a point feature. Having said this it would be desirable to reduce the meta data overhead. The language may be further compressed and future work may address data efficiency further.

Possibly of more concern than size is that the meta data imposes an extra request / reply delay on the interaction. With many slow or highly loaded networks this could be seen as a serious problem: both in terms of responsiveness under good conditions and in terms of taking up time which might be better spent transferring data. There are two clear responses to these concerns: first, that the impact will often be small enough that the benefit of more appropriate application behaviour will outweigh the impact of the overhead; second, that if the last-hop network is the source of a large part of this extra delay, then a proxy with better connectivity should be considered to perform the mediation. These concerns are sufficient that we shall examine the practical impact and make comparisons with unmanaged systems, under experimental conditions, in chapter 9.

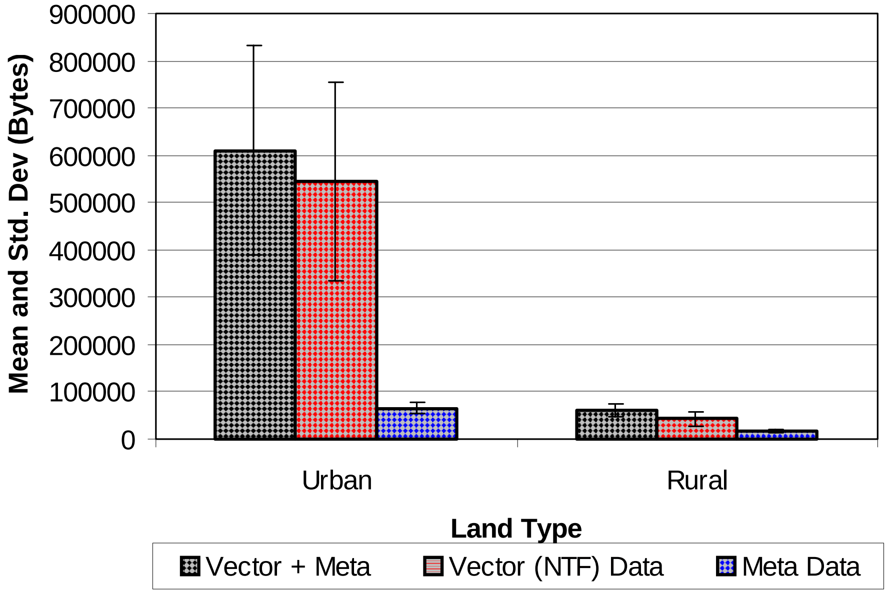


Figure 4-7: Size Range (Mean and Standard Deviation) for GZipped Data and Meta Data

4.3.2 Relationship with Existing Work

One approach to satisfying the needs of context awareness already sufficiently developed to be present in standards is the provision of meta data and meta data directed adaptability. We use meta data to describe the alternative presentations of information, but not to guide the adaptation explicitly. We shall discuss some of the work in this area next – meta data is a key input to our system and adaptation through meta data has limitations which help to motivate our user-centric approach.

Unlike many other approaches [59,61,73,88] no quality rating is provided by the authors of the data or meta data in our approach. The meta data provides only an objective description of the data. The QoS management applies its own utility rating to the data. Our approach places a burden of complexity on the user to specify their QoS requirements in detail. However, we believe that this complexity may be hidden from the average user through the use of parameterised specifications authored by specialists for particular classes of users and tasks. The content provider's loss of control is balanced by their ability to specify the type of data represented (which is often the underlying theme in justifications for abstract quality ratings by the provider) and to limit the variants provided. The benefit is in enabling users of diverse devices with diverse interests to access data for a small overhead in the data provision process. The effort required to produce meta-data is significantly lower than that required to hand-tailor data.

One example of previous work which simply describes data and includes the notion of at least somewhat structured multimedia documents is [17,70] (v. Bochmann, Kerhervé et. al.), which has a similar notion of mono-media elements being available in multiple variants. The variants are described according to their attributes with quality determined by comparison with requirements, as in our system. Their approach is based on a “news on demand” application, using streamed video as a main source of data. The data used is explicitly “voluminous and unstructured” and so the variants offered are coarse grained. The negotiation of which combination of variants is to be delivered is based on limitations expressed over cost, coding attributes, etc. and dynamically measured resource use. The specification is a described as a monolithic expression of acceptable levels, e.g “less than 50 cost units”.

Hori et. al. from IBM [61] include the use of a “role” in the context of adapting web pages along with an “importance value”. They suggest that the client may use these in allocating resource limits for data volume and screen area. Their motivation is the ability to describe data for modification (by proxies) to provide for display on restricted devices, in particular by splitting web pages into sub-pages. Our approach is more general, in that the relationships between the elements of the maps are not generally a matter of sequence which need to be satisfied in their display. Also the rating of the elements is not included in the meta data. The issue of sequence in temporal document structure, discussed in work such as [25,61,62], is necessary to consider when applying these techniques to other application domains and media encoding formats. Smith, Mohan and Li [106] describes a technique for extracting “purpose” from images automatically. This and similar techniques may be used to aid meta data generation where authors do not wish to provide detailed type information.

IETF RFC 2295 [59] describes a protocol for content negotiation for HTTP. The transparent content negotiation scheme described in RFC 2295 was designed to improve on the limited scalability that the accept header content negotiation of HTTP1.1 offered. In this case the server describes (per element) the available variants to the user agent, which then performs the selection and requests the chosen variant. Our approach is designed around meta data which is specified for at least a page at a time – multiple data elements are described in one meta data file and their relationship is also described. This enables our negotiation scheme to take account of the combined requirements of multi-part documents in one selection and negotiation round. This does impose a greater overhead at the start of each page load, although the overall overhead due to the provision of meta data should be less as only one additional interaction is required per page, rather than one per element. In addition this approach generally impacts caching policy, as the negotiation should pass through the cache to be fully effective.

The IETF have also approached this problem in the content negotiation working group. RFC 2533 [73] describes a syntax for describing media which bears some similarity to that used in the HTTP protocol. Where selection according to limits in resolution or language, etc. are required, their specification format is clear and easy to define. However, where selection is to be amongst a set of acceptable alternatives which vary across several attributes which have a (more or less) sliding scale (such as image size, as opposed to language) the use of their language starts to appear cumbersome. As for the HTTP protocol we claim an advantage in having meta data which is designed to describe multi-part multimedia documents.

Another suggestion from the W3C and the WAP forum which comes out of interests in mobile use is the “Composite Capability / Preference Profiles” (CC/PP) negotiation framework [99]. This work has some similarities to the IETF CONNEG work [73] and efforts are being made to ensure that they interwork. The CC/PP language describes the hardware and software capabilities of devices and preferences of users. The user preferences catered for are primarily binary switches or exact matches, e.g. for selecting language. The suggested interactions for applications such as web browsing are that the client sends a minimal specification at the start of a session (with a given server). As in the HTTP accept header negotiation (above), this suffers from a potential scalability problem, in that the server must hold a description of all recent clients and perform selection, in addition to delivering data. This protocol seems limited with respect to describing perceived quality over attributes which may take a wide range of values, although it provides a clear means to express limitations, particularly with respect to devices.

We do not see our approach as being conflicting to that of specialisations based on XML, such as the Wireless Mark-up Language (WML) [117]. Where WML encoded versions of documents are available they can be described in meta data and made available for the appropriate devices. WML and other specialised approaches have the advantage of tailored presentation and syntax appropriate to the mode of use. However, our approach has the advantage that is not tied to particular classes of devices and does not require separate author effort to produce. This facilitates the use of emergent devices, whilst allowing specialised data to be used where the effort has been taken to provide this.

There is also an emerging range of standard meta data formats based on XML, which is a likely direction for the next version of our meta data. Of particular interest is the Dublin Core [45] and OpenGIS [95] work. However, the focus of these standards tends to be on meta data to aid complex searches across data - describing the meaning in the data. The description of the structure of data and the attributes of the data is not their main purpose. The OpenGIS standard however has put a lot of work into vocabularies and syntax to describe geographical entities and we may at least partially reuse these definitions in future work. RDF [77] is a highly general syntax for meta data and may also be applied to our future work. There is very little description embodied within the RDF standard, it essentially allows for the definition of resources and properties of those objects, which allows the description of relationships between resources. Our approach to meta data could be represented in this way.

4.4 Dynamic Data

4.4.1 Link Services

In addition to "traditional" map data, surveyed at intervals usually measured in years, one might imagine maps that contain representations of rather more dynamic aspects of the world around us. Providing hyperlinks from spatial data to other multimedia sources is not a new idea. It does however provide an additional motivation for mediating map data. One can imagine that for any given area there are many information sites describing the place in question, such as:

- Traffic speed, accident and roadworks information, etc.
- Web sites for shops and offices. The most useful page to access might be opening times, price lists, advice on transport, etc. depending on the context of access.
- Web sites for parks and nature reserves. These may also describe opening times, details of migratory birds recently seen, autumnal tree colours of interest to photographers, snow conditions and avalanche warnings, etc.
- Contact details for nearby services. When looking for medical assistance rapid identification of the nearby hospitals, open pharmacies, etc. may be of interest.

The potential for information overload is clearly present in such a diverse range of data. Due to objects linked-from, or information linked-to, a large part of the data will not be of interest in any given context and may be omitted.

4.4.2 Transcoding Data

One popular approach to limited resources is connection to the wider networked world through a transcoding mechanism, as seen in Fox et. al.'s work [53]. These commonly provide a fixed adaptation to known device characteristics. This approach is included in our model through the use of intermediate nodes which advertise their capability through meta data: meta data from servers passes through proxies who return an estimate of their ability to modify the data. Where the proxies also cache data the estimate can reflect a cached transcoding or more complex analysis. The advertisement has the advantage that use is not tied to particular classes of devices and does not require separate effort to produce documents tailored to specific devices. This approach facilitates the use of emergent devices, while allowing specialised data to be used where the effort has been taken to provide it.

4.5 Summary

We have motivated our use of vector map data as offering a benefit in flexibility compared to raster maps. Benefits accrue from flexibility in the data to be downloaded and presented, flexibility in the level of detail used per feature and flexibility in the data sources used to provide the data presented. This is expected to enable more effective mediation than could be achieved with raster maps, although placing some burden where a full rendering is expected.

We structure data as semantic *elements* realised by *variants*. The elements and variants can be structured and together describe alternative representations of a coherent document being requested. This description enables mediation while placing what we expect to be an acceptably small overhead on the system. The benefits of describing the whole document in advance include a more sophisticated level of trade-offs available to the mediation mechanism and a single overhead load rather than a series of negotiations for multiple elements. The use of semantic *types* in describing elements offers a level of differentiation which is not available where all data are treated as equal.

Further details of our map server, type system, map data and meta data formats can be found in the appendices.

Chapter 5 Context Awareness and Resource Management

A central requirement for contextual mediation is a way to describe context in a clear manner. In this chapter we describe the *model of context* we are using. We model both abstract contextual states and dynamic resources which may be reserved. A model of contextual state, which we describe first, will enable appropriate specification to the mediation process. A model of available resources, described second, will constrain the mediation through the satisfaction of goals.

5.1 Aspects of Context

Before we address the issue of a model of context we shall extend our definition of context. We discussed what constitutes context in chapter 2. As we have stated we consider context to be wide-ranging and what concepts are relevant or detectable can be expected to vary and evolve. We wish to be able to react to many different circumstances independently. We start by re-stating the definition of *context* which we arrived at earlier:

> *Context is the circumstances relevant to the interaction between a user and their computing environment.*

In order to handle the different circumstances of context we have a notion of *context aspect*:

> *An aspect of context is a conceptually separate characteristic, whose value describes a single circumstance from the overall context.*

Aspects of context and the values the aspect may take (in the map application example) may include:

- Location, including descriptions as:
  - absolute coordinates in space
  - abstract coordinates, such as “at work”, “at home”
- Speed
- Screen size (in pixels)

- Screen readability (due to lighting conditions etc.)
- Task which the application is supporting, e.g.:
    - navigating, with sub-tasks of being a tourist, being at work etc.
    - researching an area
- Other activities engaged in, e.g.:
    - walking
    - driving a car
    - observing and recording surroundings in detail
- Mode of transport, e.g.:
    - foot
    - cycle
    - car
    - train
- Social context, e.g.:
    - in private
    - with customers
    - with colleagues
    - with friends

The aspects, in practical terms, may reflect single sensor inputs or the fusion of sensors. However, it may be that some sensors describe more than one aspect, e.g. Global Positioning System (GPS) receivers commonly describe time, location and speed. These may be separated out from the single reading into three aspects. We have attempted to encode requirements due to mode of transport, speed, a task of navigating with sub-tasks of tourist and work for the application developed.

Sensors need not be on the device the user is interacting with. A feature of the ubiquitous computing vision is that many devices will be present in the environment and may be interacted with. Temperature, lighting and noise may be sensed within a room and the information provided to other devices in the room. Research on active badges by Want et. al. [116] describes an infrastructure which can locate objects in a room. The room then knows the location of the active badge and can provide this upon request. Some forms of context such as co-location may use techniques such as this, or broadcast requests to user’s devices, to discover who is nearby. Social context may then be established by interaction with peers. In many cases the user’s device need not carry a full range of sensors as the environment can provide many of these.

We have discussed that we would like to maintain a separation between the various aspects of context. We expect to be able to specify mediation to reflect a small number of aspect-conditions, rather than a complete context. Each aspect value shall therefore be annotated with an aspect name.

The range of contextual aspects leads us to require that each aspect's value can be interpreted correctly, we shall discuss this in the next section.

5.2 CONTEXT STATE DESCRIPTION

We maintain a separation between the context aspects for a number of reasons:

- The separation allows the aspects which are sensed to be varied as need and/or capability arises. Ubiquitous computing by its very nature has to address heterogeneous needs. Any assumption about context sensors being available and providing a valid signal is likely to be false.
- A context aspect depends on a minimum number of sensors. By representing needs as reflecting only the aspects which give rise to it the aspects of context sensed can be varied with minimum impact on the specification of needs. For instance needs arising from "driving a car" can be described independently from those describing the device in use. Many devices may be used in a car and many devices may be used while engaged in other activities.
- Where a combination of aspects is required to describe a state this can always be produced from separate aspects, while the extraction of a single aspect's value may not always be possible where context is described as an overall state.

The value for each aspect of context may vary widely along with the variation in aspects. Each aspect must be typed to enable interpretation. Some shall represent numerical values, possibly with units, e.g. speed. Others may be more complex data types, e.g. locations described as areas. Aspects may also be represented with abstract values, possibly in a hierarchy, e.g. "at work", "at home", "in kitchen". Standard models are required for numeric contexts, such as coordinate systems and for abstract contexts, such as conceptual locations, in order to make comparisons. In some cases translation functions may be available.

There may also be a degree of certainty attached to an aspect's value. For instance, the location system may be 100% certain that we are "at home", but only 60% sure that we are in the kitchen, with a 20% chance that we are in the garden. This certainty mechanism can also embody context aspects whose state cannot be determined at the time of the interaction being considered. We have not implemented this in our context model, to date, although we revisit this issue in chapter 6.

Context may also be expected to change over time, each aspect changing independently of the others. Changes may include the network conditions, the device being used or the location. All of these are amenable to automatic sensing. Context may also reflect changes in the user: the task being undertaken may change without another event being detected, the sequence of contexts leading to a context may be considered to change the context, even the simple repetition of an action may have an effect on the desirable behaviour of a system. History therefore may form a part of an aspect's description of state.

5.2.1 A Model of Context

We use a total context, *C*, as a combination of context aspects. The values each of these aspects take will be determined by the aspect they describe. In eqn.5.1 c_i is the value for aspect *i*.

$$C = \bigcup_{\forall \text{aspects, i}} c_i \qquad \textbf{(Equation 5.1)}$$

Each aspect's value is a tuple or optionally, if not 100% confident, a list of tuples, of the form in eqn.5.2:

$$c_{\text{aspect}} = \langle \text{value, type, [units], [reference to abstract value model], [confidence]} \rangle \qquad \textbf{(Equation 5.2)}$$

This definition of an aspect value requires a value and type (numerical type or abstract). For some aspects units may be appropriate, e.g. speed in miles/h, km/h, m/s etc. Where an abstract type model is used this may be referenced for interpretation. A confidence (normalised) indication should be used where confidence is understood by the aspect's sensor interpreter.

The types used to describe context values should be amenable to testing to identify whether the current value meets given conditions. An equality operation is clearly needed and is simple to provide in most cases. Ordering operators are also needed in most cases. For numeric values the definition of these is as normal. This enables behaviour to be defined for ranges of values, e.g. where "20 <= speed < 40". Where context takes more abstract models we will still require order. We have used abstract contexts which form trees, as shown in figure 5-1. The tree hierarchy allows us to define a *more general than* operator >, e.g. "at workplace > in meeting room" is true. More complex type structures (such as trees where nodes may have multiple parents) are possible, but we have not investigated their use.

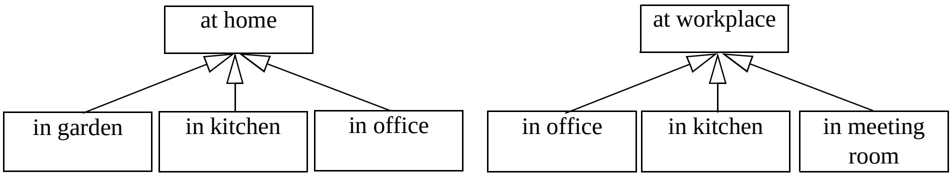


Figure 5-1: Abstract Locations

5.2.2 Relationship with Existing Work

In Dey, van Laerhoven, Schmidt et. al.'s work [44,76,105] contextual data from sensors are combined by matching combinations of sensed conditions, in order to describe overall contexts. While this is desirable behaviour when using context to annotate data or trigger actions, it is less useful in our approach: the mediation specifications which derive from one aspect (or the combination of a few aspects) can stand apart from those for other aspects. This enables extension of context aspects and flexibility in the combination of aspects which would be complex to achieve with monolithic specifications reflecting the overall context. We take the various aspects of context and produce a specification which reflects their combination rather than directly using the complete context. Rather than employ unwieldy monolithic specifications, the specifications can now reflect the context in a detailed and flexible manner. Some combination of input from different sensors may be used to improve accuracy, or confidence in correctness, for an aspect. This fusion of sensed data is described for location in [80] (Leonhardt and Magee).

An implementation where data are rendered in a suitable manner for the context, e.g. modifying text display due to movement, lighting and social conditions is described by Schmidt in [104] and the idea is widely used as motivation in the literature. However in this case and in general, the adaptation to context is presented as a tailored adaptation to specific contexts. This requires that in addition to sensing each context the response to that context be encoded. Our approach is to provide a more generic framework for describing how to adapt data to context. The effort per sensed context is then in adding or modifying a specification. This also has the benefit that conflicting specifications, which may well result from evolving specifications, will be resolved in a consistent manner.

5.3 RESOURCE MANAGEMENT – NETWORK ADMISSION CONTROL

We require that our system choose between an arbitrary number of alternative representations of a compound document, each element of which may vary, e.g. a map page may consist of roads, contours, water features, etc. Each of these may be built up from basic information and additional detail. It is likely that more than one of the offered variants for any element would be *acceptable* in principle. It is desirable to provide the best possible presentation, which would be context dependant. The selected variants should maximise overall utility and present data in a *consistent and predictable* manner. In addition, other *goals* may have to be met such as a deadline for the map download. We may derive the deadline from the estimated utility of the data: how “interesting” it is may determine how long we are prepared to wait for it to download.

Initially the highest utility variant representing each element is selected. Where multiple variants have the same utility the one which makes the smallest resource load is chosen. The selection function makes a note of the selections on the meta data graph and provides a summary of the selected variants. From this selection the total resource requirements are calculated and requested. The resource manager’s admission function will take the resource requirements and goals (e.g. deadline) and test its resource model to see if the request can be admitted. If the admission request is accepted then the reservation model is updated and the data requested. If rejected the resource shortfall is returned and incorporated into the goals. e.g. reduce data volume by 10kB. Selection functions will be discussed in more depth in chapter 7.

The network resource model maintains an estimate of throughput and round trip time. We measure throughput to the application, after any compression etc., rather than network bandwidth, although the two are inter-dependant. Throughput may be consumed over time by a request. This is represented using a simple time-slice scheme, as seen in Degermark et. al. [42]. Time for the network is broken into a series of slices, of a fixed length, e.g. 100ms. A reservation request is of the form of a start time, t_1, a latest end time, t_2, with a size, s. If the data / time is greater than the model of available throughput, the size difference required to satisfy this basic constraint is calculated and returned as a rejection. The network throughput may be reserved between these two times until:

$$s = \sum_{\forall \text{slices } t_1 \dots t_2} \text{slice throughput reserved} \times \text{slice time}$$

(Equation 5.3)

This we refer to as a *greedy reservation*, as illustrated in figure 5-2. This model assumes flows are responded to sequentially and in a best-effort manner. Alternatively, s may be divided by the number of time slices between t_1 and t_2, to give s_s. Throughput of s_s is reserved at each slice s, any shortfall is distributed across the other slices. We refer to this as a *deadline reservation model*. This model allows for parallel treatment of concurrent flows, with a just-in-time rate control and is illustrated in figure 5-3. Both these models are in common use for resource management and scheduling to meet deadlines.

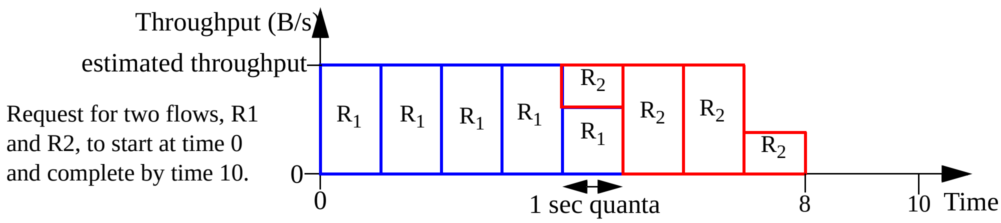


Figure 5-2: Two Greedy Reservations

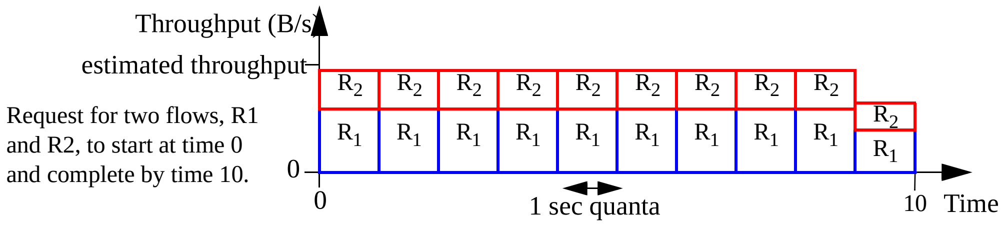


Figure 5-3: Two Deadline Reservations

If the request can fit into the model, admission is granted, otherwise the shortfall is returned and the prospective reservation removed. At this time we have not implemented any interfacing from this application level model into lower level resource models, or network QoS control mechanisms such as RSVP.

5.4 Network Monitoring

The performance of most networks is somewhat variable. For wireless networks the underlying QoS is highly variable, so monitoring is required to feed into the network resource models. The admission function can provide more accurate control as a model of effective parameters is built up. These models of resources will never provide a guarantee of resource parameters. However, more predicatable application behaviour can be achieved by improving the accuracy of the resource model.

Resource monitoring is provided by the application: each request (element request or meta data request) is supported by a thread which handles the connection establishment, reading from the buffer and timing. Once a minimum size of buffered data (or end of file) has been reached, the buffer is passed to the application. In the case of map data, partial data may be parsed during data loading, while any data which the parser cannot use yet is pushed back to the loader. The application measures round trip time and effective throughput for the meta data load and the full collection of elements to be loaded. The interaction between arrival time, data size and throughput are managed in the text book manner, as shown in the equations below. The round trip time is measured as:

$$\text{rtt} = (\text{1st reply time} - \text{1st request time}) - \left(\frac{\text{1st reply size}}{\text{throughput}}\right) \qquad \textbf{(Equation 5.4)}$$

Request and reply times are measured as the application calls the HTTP API's "GET" method and the data is read from the buffer. The reply size is considered in calculating the round trip time, as a significant volume of data may be delivered in the first reply to the application. The consideration of 1st reply size will be inaccurate where network bandwidth is not symmetrical. As a multipart document may be loaded, the interaction of the various requests is hidden by considering all of a group of requests. The use of first request, first reply and last reply achieves this combination. The throughput is measured as:

$$\text{throughput} = \frac{\text{data size} - \text{1st reply size}}{\text{last reply time} - \text{1st reply time}} \qquad \textbf{(Equation 5.5)}$$

Any requests where the timing difference between first and last arrival is too small (<= 2 ms), or the data transferred is too small (<128 Bytes) are ignored in calculating the round trip time as the timing in Java is made to 1ms resolution, so inaccuracies and overheads become too significant for accurate measurement. The technique presented relies on sufficiently large data loads and small timing intervals, to make accurate measurements. In this application domain this is usually the case. Further accuracy could be gained by making use of the measurements commonly performed within the network protocol stack, however our results are satisfactory at this time and benefit from greater portability for not being so close to the lower levels of the network stack. Where applications compete for network resources we assume that either they make use of our network monitoring and admission control as a common system function, or that their impact is steady (on average) and so can be treated as a limitation on the network which is reflected in the measured throughput to the application.

These values are reported to the resource manager and the models updated:

a. A recent history is updated. We typically keep the values of the last three loads. A recent average is calculated.

b. An exponentially weighted moving average (EWMA) model is updated at a given ratio of new to old value.

c. If the recent average is found to be very different (by a set ratio, jw) from the decaying average value, the decaying average is set to this recent average value.

This model is used for both round trip time and throughput. The test configuration has EWMA ratio (w) = 0.25, the jump ratio (jw) = 1.094 and the recent history has 3 slots. This mechanism and its place in the negotiation process are illustrated in figure 5-4. The model update algorithm as used for the throughput model (with some class casting removed) is also shown in code, below:

```
public synchronized void reportThroughput(int bw) {
int    recentAvg, currentVal;

    // Add this reading to recent history
    lastBWReport.add(bw);
    if (lastBWReport.size() > bwHistoryLength)
        lastBWReport.remove(0);
    // Find recent average
    recentAvg = 0;
    for (int i=0; i<lastBWReport.size(); i++) {
        recentAvg = recentAvg + lastBWReport.elementAt(i);
    }
    recentAvg = recentAvg/lastBWReport.size();

    // update throughput moving average
    currentVal = ((1-bwUpdateRatio)*getThroughput()) + (bwUpdateRatio*bw));
    setThroughput(currentVal);

    // Is recent average more than jump sensitivity away from general average?
    // If yes, force throughput to be recent average
    if ((recentAvg > (currentVal*bwJumpSensitivity)) ||
         (recentAvg < (currentVal/bwJumpSensitivity)))
        setThroughput(recentAvg);
}
```

The EWMA (after any forced change) is used in the network model. The jump test aims to provide a faster response to large changes in monitored values, or a large deviation from some initial assumption. We present tests of the responsiveness of this monitoring mechanism in chapter 9.

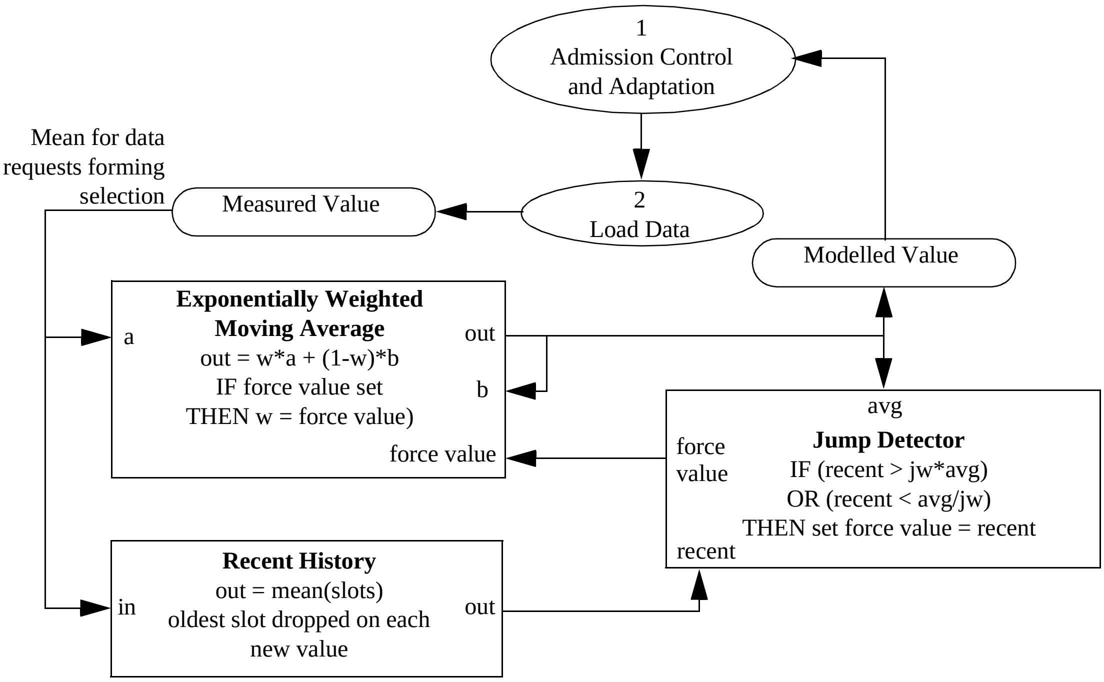


Figure 5-4: Resource Model and its Interaction with Loading and Selection

This two-average technique which aims to provide both stability in the face of noise and agile response to change is similar to that described by Kim and Noble in [72] as being better than simple single average EWMA filters. Their model uses two differently weighted EWMA filters, again using passive readings from actual network use. The more stable output is used in most cases. Spot readings of network bandwidth are taken (from two readings) and where these are different from the stable output by more than a certain factor, the agile filter's output is used. The effect is similar to our system with a recent history of two, although the recent history is replaced with an EWMA filter. The value jumped to in our system may be somewhat less accurate under large changes as all values have an equal weight in our system while newer values dominate in the EWMA filter.

The update values were chosen to trigger a jump after two readings of 150% of the prior level, e.g. ...1000, 1000, 1500, 1500 causes a jump at the second 1500, while maintaining a stable model in the face of minor fluctuations. For a given history length the jump sensitivity required for this can be found automatically. Within these criteria a rapid move towards the actual value is to be preferred. Using a spread-sheet implementation of the model we examined a range of history lengths and update ratios, over a sequence of network conditions. The ratio of actual condition to modelled value were taken. We chose an update ratio of 0.25 and a recent history of 3 measurements and jump ratio of 1.094. These values showed a mean actual value / model close to 1 and a low standard deviation. In our test application the meta data load was reported as one load and the various loads for data are averaged and reported as another load.

We present in figure 5-5 a comparison of various update ratios for a history of length 3. The modelled value steps first from 1000 to 1500, which was the condition designed to cause a jump after two readings. The next up-down step was to a lower value and then one was taken to two higher values. We then added 20% of nominal value “noise” to the readings inserted two large spikes, followed by a larger than jump condition step. We see that the behaviour of the three test ratios was very similar. The lower, 0.2, ratio took visibly longer to adjust to change in the noisy cases. The higher ratio tracks the used value closely but the jump was activated due to noise. This can be seen in the latter part of the trace, where the 0.25 ratio line remains flatter. For our implementation we felt this instability did not justify the slightly faster adjustment to larger changes and so was not used.

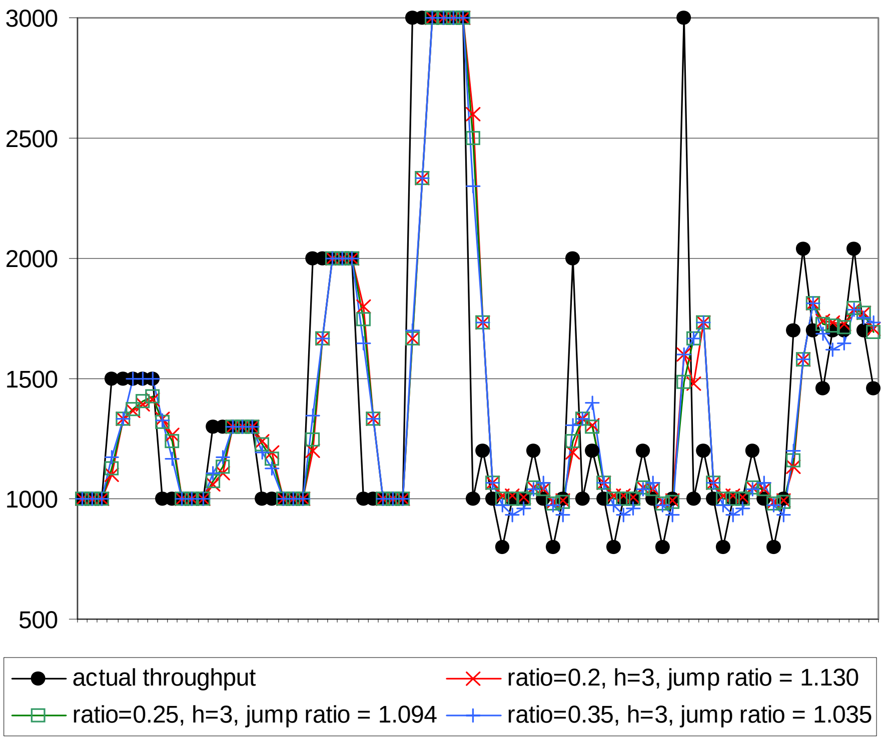


Figure 5-5: Comparison of Throughput Update Ratios

In figure 5-6 we present the same sequence of modelled throughputs but examine the difference between history lengths. For each history length we present the lowest mean and standard deviation that we found (using 0.05 or smaller update ratio changes). We see that with a history of length 2 the jumps were performed more quickly, indicating a more agile response. However, the response to smaller changes was very slow. With a longer history we see that the response to larger changes was slower.

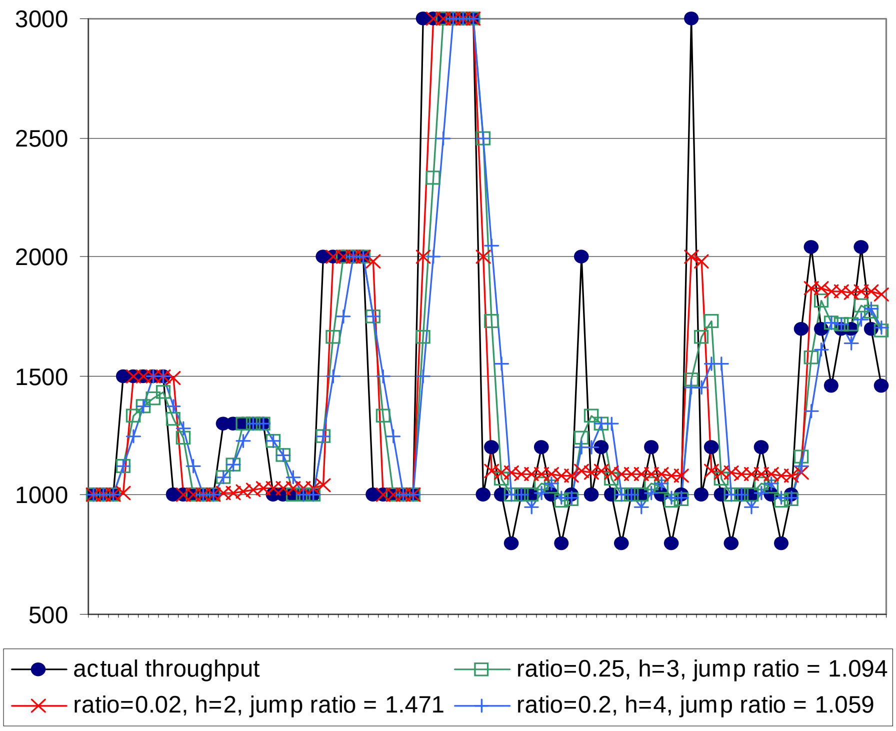


Figure 5-6: Comparison of Throughput History Lengths

We maintain a separate resource model for various (partial) URLs. This allows us to model the different networks and responsiveness of different sites as well as our local characteristics. With our meta data / map data application we maintain separate models for the meta data and map data. While this slows our response to changes in the network it allows a more accurate model of the requests for data, which differ from the requests for meta data. The meta data requests involve traversing an R-Tree to identify suitable data before commencing a response. The data requests access the data through a list of offered variants associated with a request object. The request object is kept in a hash-table and so is retrieved faster. In addition the meta data and map data allow different degrees of compression. The compression affects the perceived throughput, as we measure the size after compression. We do not measure the compressed size as the compression achieved varies with the data volume and so cannot be calculated accurately on a per-variant basis by the server when producing the meta data.

5.4.1 Issues in Monitoring Multiple Flows

The calculation of throughput we have shown is applicable to either a single flow which fully uses the limiting hop on the network, or over the aggregation of multiple flows. It is not applicable to individual flows sharing the network unless the sharing can be expected to be constant between requests.

In our application it is likely that multiple flows will be required. The map data load will be split into multiple requests, for data with different headers and in order to manage the GET request size. When issuing multiple flows it is desirable that they be staggered in time in order to avoid a rise in processing time at the server as a result of contention for computation time. In general network use it is also desirable to have requests pipe-lining or at least starting at different times in order that the flows do not compete with each other and cause a loss of performance.

The network throughput reservation used assumes that the data request is made as one request and data transmission is evenly spread across the time. This is inaccurate as multiple requests are made. This is for two reasons: Firstly the map server can only handle one header variant per request. Where data spans multiple tiles multiple requests are required. Secondly, the GET request's URL is limited in size to 1024 characters. The list of variants required for many requests is substantially longer than this. The request is then split, with multiple data requests with the same header variant being issued. (This causes some difference between the actual data size and that reserved but this is a small enough proportion that it has been ignored.) Issuing multiple requests at once is well known to be bad practice, as the various TCP connections compete and the server gets a single large computational load. We do not assume HTTP/1.1 and so open multiple requests. Use of pipelined requests would be preferable to the following method, but is left for future work. While we cannot ensure a smooth response we can attempt to construct a steady flow of requests which ensure the connection is always occupied by at least one data flow. Our network performance estimation would suffer substantially if there were significant breaks in the data flow. We achieve this by issuing the requests over time. Rather than make precise predictions of each request's behaviour (which duplicates a lot of counting) we make some basic assumptions about how long requests will take and desirable levels of concurrency. The spacing of loads is determined by this. Having some concurrent requests prevents improvements in the network going undetected. Requests which are expected to be larger are dispatched first as these will be longest running. Our metric for this is URL length (the length of the URL provided to the HTTP GET), which is at best an approximate measure. In practice however this is sufficient as a typical requests consists of: one or two requests for the main map data tiles, which have been broken into similar (URL length) parts. These requests are likely to have approximately the same size data as each other. There are also usually several smaller (URL length) requests for data from smaller parts of neighbouring tiles or additional data sources (we have an overlay of tourist data with a different header for instance). These smaller requests are issued later and can be expected to finish faster. The degree of concurrency is not critical – maintaining a flow of data is the main intention. If these assumptions are found not to hold as the data evolves the actual size of the variants for each request could be calculated with some reworking of the request issuing mechanism.

Our request dispatch algorithm is shown in (simplified) Java below.

```
protected synchronized void requestData(Vector dataRequestList) {
int              requestCount;
int              rtt; //ms
double           expectedTotalTime, expectedReqTransmitTime; //ms
int              concurrentLoads; // target concurrency not enforced
DataRequest      req;

    // sort requests by URL length - a very basic indication of data load
    // helping longest requests be dispatched first and small requests later
    dataRequests = sortRequestList(dataRequestList);
    requestCount = dataRequestList.size();

    // to limit flow conflict
    concurrentLoads = 2 + (requestCount/4)

    // time of flight for each request, assuming even sized
    expectedTotalTime = 1000*getByteSize(variants) / throughput(req.siteName);
    expectedReqTransmitTime =  expectedTotalTime / requestCount;
    interLoadGap = Math.min(rtt/concurrentLoads,
        expectedReqTransmitTime/concurrentLoads);
    requestNextData();
}

protected synchronized void requestNextData() {
DataRequest      req;
Thread           reqRunner

    for (int i=0; i<dataRequests.size(); i++) {
        // dispatch a new request
        req = (DataRequest)dataRequests.elementAt(i);
        reqRunner = new Thread(req);
        newLoadForSite(req.siteName);
        // Get this loader going
        loaders++;// used as a tally
        reqRunner.start();// actually start the instrumented load
        Thread.currentThread().sleep(interLoadGap);
    }
}
```

After some experimentation we elected to issue requests at

`min((round trip time), (estimated request time))/concurrency limit`

in order to stagger requests without having gaps in the network usage. The absence of gaps is important in meeting deadline goals and in our approach to measuring network characteristics. Our concurrency limit of `2 + (requestCount/4)` is a trade-off between reducing competition and maintaining a flow of data with an incomplete understanding of the expected duration of each request.

Round trip times will not substantially interact with each other, assuming that the time measured is not substantially due to initial data transfer. To measure round trip time we therefore calculate this for each of multiple requests and report the minimum into the network manager for the given request.

Throughput cannot be calculated from the individual request's throughput reliably, as we do not know in detail how the available network bandwidth was shared by the various flows. We therefore calculate throughput by taking the first response for all flows and the last response for all flows as the overall time and the total size of data delivered between first and last response for all flows as the overall size. Throughput is calculated for the combination of flows and reported to the network manager for the request.

5.5 Screen Space

Another resource which may vary with context and can be treated as a per-request consumable resource is screen space. A screen will have a finite number of pixels and so can render a finite number of different points of information. Map symbols generally take more than one pixel to impart any useful information, require some separation from other features to be read clearly and may overlap with other features in places. The clarity of the screen may also vary with context. A model of screen space is therefore not as simple as a model of pixels. We describe our model below.

5.5.1 Introduction

In a map features take a varying number of pixels to render. The number of pixels used by a feature in a vector map is determined by the feature's path, the rendering applied and any space required around the feature to ensure it is distinct from surrounding features. Until the full vector data for a feature is available its path is not completely known. However from its semantic type we may know whether it is a line, area, point or text feature. Meta data may also describe the area which encloses the feature, from which an estimate of its path length may be made, e.g. a road will be estimated to take at least the length of the diagonal through the enclosing area, a minor road is more likely to make turns within the area than a motorway and so may be expected to have a longer path. The path for an area may be thought of as being the boundary line of the area plus sufficient rendering of any fill to determine what sort of area it is. The semantic type may also be used to infer the rendering to be applied. Motorways are commonly drawn as wider lines than minor roads, point features are rarely represented by a single pixel symbol. In many cases, while a map may be to scale, various features will be drawn with symbols larger than a precise scale drawing would give. The space required to separate features may be considered to be part of the rendering size for many purposes.

In a web page the space taken by features is rather different. SMIL [62] approaches some of the issues here in its consideration of how layout may be rearranged in a multimedia presentation. However its approach is that of a produced production, with alternative presentations being prescribed by the author. In a more general sense we again consider that the various elements of the web page may take screen space, depending on their variants and/or on their rendering by the final application. Images may well have their dimensions described in meta data. Text may be described as a number of words, from which an estimate of rendered size may be drawn.

While the space taken on a screen by features may be estimated we need a framework for making use of this knowledge. Our specifications can already describe preferences for image sizes using utility functions over image dimensions (see chapter 10). However the trade-offs in space may depend on how the various elements in a document interact with each other. In a map with many overlapping features one may prefer to select the more important features in order to render them clearly. As the user moves towards a less dense area of the map the system can adjust feature inclusion while maintaining a clear display, without a change in the context triggering a change in specifications. In a web page an index page with few words may be better able to include large navigational buttons, logos etc. than a page with more information. The choice of text volume to include on the current page, rather than have linked to but shown on another page, may be determined by the number of times it is desirable to scroll within a given page.

Where a device is being used in a vehicle it may be vibrating. This will seriously impact readability on any screen if the information is too dense. In addition the device may not gain its user's full attention, e.g. where driving a car. A clear presentation of data which may be glanced at will be more desirable than one which requires greater concentration to use. Poor lighting may also affect the ease with which the screen may be read. All these conditions have received some work in the area of context sensing and so their consideration is timely. In each case we may require, overall, that features be less dense. This is similar to having a smaller screen, but not entirely the same. The screen size may affect what data is appropriate and when features actually overlap.

We model the screen as a resource with given dimensions. Languages such as Java make finding screen (or more importantly application display panel) dimensions a trivial task. From these we can form a reservation model. The inclusion of a feature in the map will cause space on the screen to be reserved. This will not be precise pixels, but an estimate of the number of pixels and possibly which approximate area of the screen they shall be in. As variants realising elements are changed the reservation of pixels will change to reflect differing parameters. The context will give rise to a goal of *maximum feature density*. This is given as a ratio of feature pixels to screen pixels and is applied in the negotiation process alongside a deadline. In a map to be rendered without scrolling this may be less than one. In a web page to fit within three screens it may be around three. In practice density for maps may be higher than one as some feature overlap is to be expected and in our application we tend to overestimate an element's impact from the meta data having bounding boxes rather than exact dimensions.

It may not always be desirable to model the whole screen at once. The reality of feature distribution may result in some areas being significantly more dense than others without the average density failing the goal test. The screen area can then be broken into smaller areas. This can reflect the average density for each area and the goal be applied independently of the number of sub-areas used. Where it is not certain where an element will be in the screen the estimate can be divided across all potential areas. For instance a road may have a bounding box covering one half of one screen area and one quarter of another. We would apply two-thirds of the estimated screen space required in the first area and one-third in the second.

Note that the action of utility functions describing preference due to screen size is independent of this resource management function. Limiting selected variant size due to screen size simply provides a starting point in selection, as describing size preference provides guidance for selecting the most appropriate variant. The screen density resource considers the combination of all elements' selected variants.

Context may also affect the way in which the features are rendered. For instance, for users with poor eyesight or in cases of reduced screen visibility thicker lines and larger text may be used. The rendering of maps may also be affected by preferences for features, for instance brighter and/or bolder drawing of the most important data and dimmer drawing of other information. This kind of change will impact the screen space reservation, but may occur separately from changes to the screen density limit. We have not yet considered modified rendering in our implementation. Inclusion of this facility would require that the rendering engine provides current metrics on drawing to the screen space reservation system.

5.5.2 Implementation

The first part of making a reservation is to define the zones of the screen which shall be used for localising the density limit. We found that a single reservation treating the whole screen was prone to clutter as open areas reduced the average density while allowing overcrowding in other areas. We divide the screen into a grid of `3+user interface size/400` in each dimension (taking the largest dimension of the user interface, in pixels). Each of these zones is allocated an area, by querying the location to be displayed from the application.

To make an initial reservation the best selection is applied, as for network throughput reservation. The element tree is traversed and its impact noted. Different metrics are applied for different element types. An estimate of the number of pixels required is made, which depends on the area the element(s) cover, the number of features in an anonymous group and the type of the element. The meta data may describe elements which fall partly off the screen. The whole of the pixel count for the element will not be applied in this case, just those parts which overlap one of the defined screen zones. An illustration of an element reserving space on a four zone screen is shown in figure 5-7. The algorithm for applying the use of an element is show below, with simplifications for clarity.

```
protected void applyToZones(LocationI element, int pixelCount) {
double    elementArea, intersectionArea;
LocationI intersectionLoc;
int       pixelUseForZone;
int       currentZone;

    // compare element with zone locations,
    // add to pixels for pixelCount / (intersection area / element area)
    elementArea = element.getArea();

    for (int i=0; i<zoneDim; i++) {
      for (int j=0; j<zoneDim; j++) {
       currentZone = j*zoneDim + i;
       intersectionLoc= element.getIntersectArea(zoneLocations[currentZone]);
       if (intersectionLoc != null) {
          // if it does interset, then its got an effect on the screen
          intersectionArea = intersectionLoc.getArea();
          pixelUseForZone = pixelCount*(intersectionArea/elementArea);
          zoneUseCounts[currentZone] += pixelUseForZone;
       }
      }
    }
}
```

Once all elements have been considered the density is calculated. This is a simple comparison of pixels used in zone / pixels in zone against allowed density. Any one zone failing the test causes the reservation to be false.

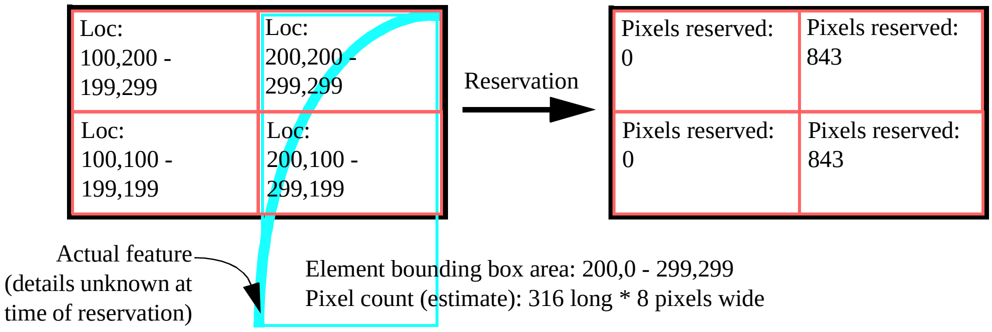


Figure 5-7: Reserving Screen Space for an Element

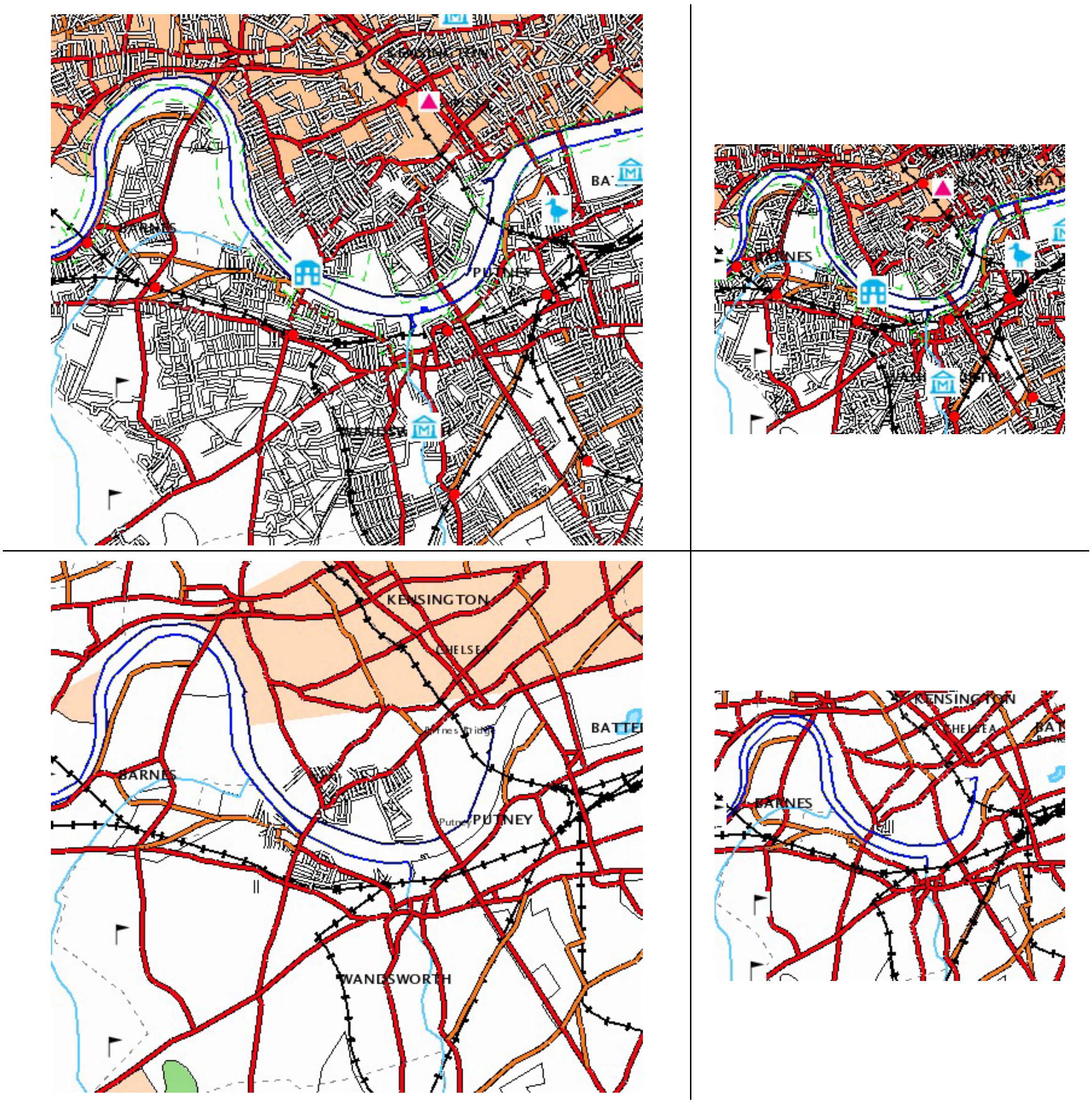


Figure 5-8: Meeting Screen Size Limits Through a Density Model
Top – Unmediated Data, Bottom – Mediated Screen Density=2.5
Left – Application Window 500*500 pixels, Right – Application Window 300*300 pixels

At this point the reservation process has to differ from that for network throughput. It would be inappropriate for the selection mechanism to know the effect of any given element and which screen zones the element contributes to. As the selection is degraded the removal and/or addition of elements (or their variants) is described at each step to the screen resource manager. Where an element is removed it has a negative pixel count and the same algorithm can be used as for addition.

We illustrate the selection of map data limited by allowed screen density for different screen sizes in figure 5-8. The location shown is 7.4*6.1km. The context for the mediated maps is driving in a car, on business, at 30mph. In the upper row one map is shown at two sizes where the data has undergone no mediation. The features, small roads in particular, are cluttered so as to make them unreadable. In the lower row the same map area is shown at the same two sizes, this time with mediation applied. The context-based specifications have driven the omission of data to result in a clear map at both sizes. The inclusion of more detail in the centre is a result of a preference for data near the point of interest (assumed to be the centre of the map here). In order to regain the extra detail in the centre for navigation down smaller roads it would be necessary to zoom in on the area of interest in order that each feature competes with fewer other features for space on the screen and so may be selected (or allow a greater screen density). It can be seen that the difference in the maps in the lower row makes them both readable, although the only change in context is the screen size.

5.6 OTHER RESOURCES

We have not examined other resources in detail. A simple implementation of a money resource, with cost associated with data has been implemented. This follows a similar model to the throughput resource, with precisely known costs for data requested and money available. It verified that the negotiation algorithm was amenable to multiple resources, but only demonstrates a trivial constraint and so will not be presented in detail.

5.7 SUMMARY

We have described a model to hold contextual data, which will allow the current context to be tested against conditions. This requires an identification of context aspects, the type of the aspect’s value and a means for comparing *equal to* and *between / contains* on the context and a given value.

A model of network resources has been described. This enables reservations against expected network characteristics. The expected network throughput and round trip time are derived from measurements of the (recent) past. These measurements take averages over whole flows and possibly multiple flows, for previous interactions. Fluctuations occurring over shorter times than this are not modelled directly. We can give no guarantee that the characteristics expected will actually exist. However, we can expect that large scale changes in characteristics will not happen between every flow, although some change may occur. Small changes may often be reversed, being due to temporary effects. Changes of factors of 2 or greater are expected to be less frequent. The network model is tuned to respond quickly to these large changes, so that their impact can be accommodated rapidly. The model of screen density described has similarities to other recent work, such as [72] in its provision for both agility and stability. Our model operates at an application level, which remains appropriate for HTTP style interactions where per-packet models are not required, nor is any overhead placed on the network. This level of monitoring has the benefit that no ties to the operating system protocol stack are required. The modelling of both round trip time and throughput is crucial in mobile networks and where HTTP traffic is being considered as both have a substantial impact.

We have also described a model of screen density. We use a system of pixel counting in screen zones for locations shown to avoid localised clutter, with reservations by elements which describe the area they cover. Estimates for the screen use of different types of features are applied in the reservation. This allows applications to respond to different screen sizes and effects which impact their readability, such as vibration. The model treats areas of the screen to avoid localised clutter while accepting that some overlap of features is inevitable. The model requires input from the application on how it will render the data, which may also be varied according to context but this is not treated here. This model requires application specific extensions to consider elements representing map data and an application screen for map display, as location drives layout and map feature screen use is controlled by the map rendering engine. However much of the functionality is generic and could easily be applied to the layout of data such as web pages.

Tests of the network resource model are presented in chapter 9. Tests of the screen space manager are described in chapter 9 and discussion of its effects included in chapter 8 and chapter 10.

CHAPTER 6 SPECIFYING MEDIATION

In this chapter we shall describe how we specify preferences for mediation and how they may be enabled by and reflect context. We extend our descriptions in a somewhat formal manner to enable reasoning about this process.

6.1 CONTEXT AND PARTIAL SPECIFICATIONS

We seek to enable contextual mediation, which is a form of context awareness:

> *A system is context-aware if it uses contextual information to provide relevant information and/or services to the user, or to enhance the provision of services.*
>
> *Contextual mediation is the use of context to modify a service, in particular to describe limits and preferences over a large range of offered data, in order to display the most appropriate parts.*

Context is applied to the specifications in order to find which are to be applied in mediating data to be requested at any given selection. As the specifications can encode preferences and needs arising from context they enable the process of contextual mediation.

6.2 ADAPTATION SPACE

Applications incorporate an adaptation engine which addresses issues of perception of quality [93,120]. This could be generalised to implement adaptation functions for media types and/or application classes. An adaptation engine must aim to provide the best *achievable and acceptable* delivery of the data possible. By using media and/or application specific adaptation engines, the task of making the adaptations can be broken down and an intelligent selection between alternatives where a trade-off or selection between adaptation strategies must be made. Note that this adaptation may be in conjunction with system-level adaptation and QoS management functions, which are designed to manage more minor variations in resources, while the application is responsible for coarse grained QoS management.

The specification of preferences and system limitations describes an area of acceptable performance as an *n*-dimensional model for *n* QoS managed parameters, which in turn may be combined into an *e*-dimensional model of *e* elements. In figure 6-1 a 2D space of detail against data size is shown. The parameters for the variants realising an element will generally be interrelated and have boundaries of acceptable performance. The available variants, with their various parameter combinations, form points in the performance space. A parameter must be describable as some directed continuous or discrete value series.

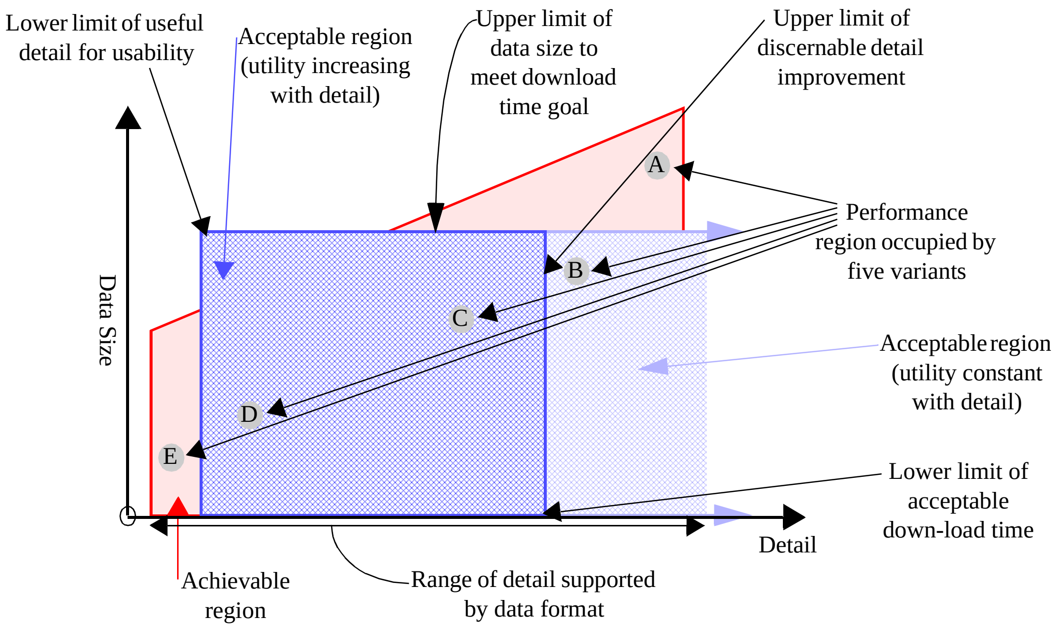


Figure 6-1: Parameter Spaces for Selection

An *achievable performance region* defines the capability of a resource in terms of a region which variants might occupy, defined by the limits of the encoding (see figure 6-1). An *acceptable performance region* defines the application requirements in terms of the same QoS parameters. The intersecting space then describes the *achievable and acceptable* region in the performance space, which the selected variants should occupy. A *variant* defines a performance region for a specific level of compression, data resolution etc. This can further be divided into two regions: a region in which the utility of all variants is one – there is no discernable difference in quality to the user (this may be a factor of the user or their equipment, rather than a technical measure of the variants) and a region in which a difference can be found between variants. In the first region any selection amongst variants is due to the resource use of that variant, while in the second the resource use can be traded-off with the relative utility of the variants.

It should be noted that the bound on the *best* end of acceptable may not be immediately obvious. A degradation in quality may not be consciously perceived, but still affect the user, as Wilson and Sasse reported in an experiment on video QoS in [122].

A data element may be provided in several variants, e.g. a map segment may be available at various scales, show various types of feature and cover differing areas. These changes in the data will affect its resource usage, e.g. by changing the size of the data, the display size or resolution required to show the map segment. One of those flow variants which is within the intersection of the acceptable and achievable performance regions should be selected to implement the flow based on goals such as: maximising QoS, maximising overall resource availability or minimising cost. [57,74,114] show examples of similar models. A change in resource availability would shift the edges of the achievable region. Changes in context might alter the boundaries of the acceptable region.

Figure 6-1 shows an example of this for a single element, e.g. a map segment image. Here we see five variants: A, B, C, D and E. A is outside the acceptable region (its resource use is too great). B is in the ideal zone. C and D are acceptable, but offer different points in the data size / detail space. E is unacceptable (its detail is too low). A selection must then be made amongst B, C and D, according to their properties, the system's resource limits and the user's preferences.

The description of the user's preferences and the system's limits is therefore crucial in managing the experienced QoS. This chapter describes our system of describing the acceptable region and making the trade-offs within it. Chapter 5 describes the process of managing the bounds of the acceptable region where these are due to varying resources. Chapter 7 describes the process of which point in the selection space to pick and the more general case of which combination of points to pick where multiple elements each have variants within the selection space.

6.3 USER SPECIFICATIONS

Users specify the importance they perceive for element types, e.g., road, vegetation and administrative boundaries, through *weights*; how they perceive the quality of the presented data to vary according to its attributes, through *utility functions*; and constraints in the use of resources they wish to be enforced, through *goals,* e.g., download time.

6.3.1 Weight Functions

Users specify their *preference* for elements of a particular type by associating a *weight* with the type, e.g. to describe a preference for displaying information about roads rather than rivers. The weight is normalised between zero and one. The function over the type takes the form:

```
weight: Type → 0..1
```

The following value assignments define the default weight for any type ("*") to be 0.4, for roads to be 0.85 and for motorways to be 0.8. Higher numbers indicate stronger preference.

```
weight(*) = 0.4
weight(road) = 0.85
weight(motorway) = 0.8
```

The types are structured through specialisation, such that a general type will be superseded by a more specific type, e.g. a weight for "motorway" would be applied rather than a weight for "road".

6.3.2 Selecting Weight Functions

The description above illustrates the use of a static value assigned to elements of a given type. In many cases this is sufficient and clear. However, it is also true that for some cases the preference a user will express for an element may depend on parameters of that element. For instance, where features over a large area are being described, a railway station closer to the centre of interest may be deemed more interesting than one farther away. In this case it is useful to be able to modify the weight of the elements to reflect this.

We assign weights as a function. The parameters the function can take reflect the parameters of the element. Where a fixed value is required we pass no parameter to the function. The functions are selected and applied in a similar way to utility functions, described below. The effect of a parameter on different types of element can be reflected by the use of different functions, e.g.:

```
select-wf: (Attribute, Type) →
(named-weight-function: Attribute.value-type → 0..1)
```

Different attributes and types may give rise to different utility functions, e.g.:

```
select-wf(distance, road) = pref-nearer(distance.value)
select-wf(distance, house-numbers) = strong-pref-nearer(distance.value)
select-wf(area, land-use/park) = pref-larger(area.value)
select-wf(<null>, road) = fixed-weight-road = 0.85
```

6.3.3 Utility Functions

Preferences for different variant representations of an element are defined using *utility functions* applied to the attributes of the variants, similar to that of Lee et. al. and Walpole et. al. in [78,115]. This is a common concept in multimedia.

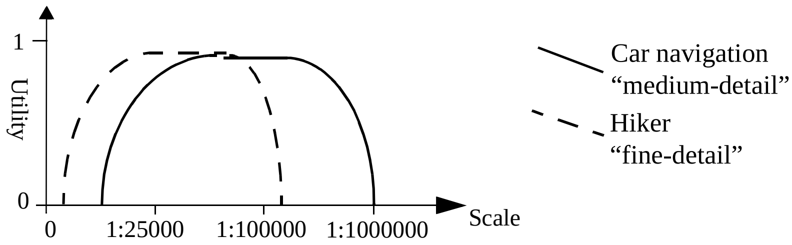


Figure 6-2: Example Utility Functions for Scale

To illustrate the use of utility functions based on types of data, we shall consider possible variations in the use of map data. For in-car navigation various features are relevant, we shall consider roads, where one might define utility functions relating to scale as depicted by the solid line in figure 6-2. Here a lot of detail makes the map harder to read, while a large scale is often still useful. For hikers, there might be less of a tolerance for the loss of detail of bigger scales, but more time to read and use the greater detail of the smaller scales. The dashed line in figure 6-2 illustrates this different perceived utility.

The utility function may assign specific values, e.g. to the language of the tourist information associated with a map. This could be represented as a series of exact matches returning values. This use of utility has a similar effect to the explicit weightings found in some other approaches, although our mechanism is more general.

The variant may have a utility function applied to each attribute described (in meta data). The utility values for the various attributes are combined to derive an *overall utility*, *u*, for the variant. As each attribute’s utility value, *u(a)*, has been normalised to between zero and one, we simply multiply all utility values together for each variant. The use of a product function has the benefit that any variant which is unacceptable in some aspect (zero utility) registers as a wholly unacceptable variant. The comparison may need adjustment when comparing between variants which have different numbers of attributes. For example:

```
UtilityFn: Attribute.value-type → 0..1
medium-detail(scale-value) = // the function as in figure 6-2
pref-recent(age-value) = 1 - ((year-today - survey-date)/20)
u      = u(scale) * u(survey-date)
       = medium-detail(scale.value) * pref-recent(survey-date.value)
```

The attributes utility functions are defined over will depend in part on the media type in question. For utility functions based on attributes which are not universal, the format must be at least partly known, e.g., “survey scale” is only a meaningful attribute for map data. In contrast, “sample rate” has a different meaning for music data and temperature sensor. Resolution, colour depth and quality (compression) may be applicable for both still images and video. Universal attributes include “mime type” and “creation date”. When defining utility functions applicable to a general format, e.g., attribute “scale” for format “map”, it is not necessary to re-specify that definition for more specialized formats, e.g. “ESRI-Shape”, unless different behaviour is required. The number of utility functions can be minimised by operating at the most general level available.

6.3.4 Selecting Utility Functions

It should be clear that these utility functions do not say anything about the perceived importance of roads within a map. If one were navigating it is likely that the “road” data would have a high weighting. However, a tourist might be less interested in the detail of the road than in historical monuments and restaurants, so loading unnecessary detail would delay the loading of more important information over a wireless network and clutter the display. Utility functions over attributes can aid in describing these preferences. When defining user preferences it is important to be clear whether one is describing the importance of a feature (type weighting) or the effect of the representation quality on the perception of the data (utility function over attributes of a variant). The *types* of data a variant represents may be used to select amongst different utility functions to be applied to each attribute, which reflect different ways in which the perceived utility of the data varies. The key difference between weight functions and utility functions is that weight functions reflect parameters of the element experienced in the realisations of all variants, while utility functions reflect attributes of a specific variant.

For example, one can match conditions to get a utility function with a selector of the form:

```
select-uf: (Attribute, Format, Type) →
(named-utility-function: Attribute.value-type → 0..1)
```

Different attributes, formats and types may give rise to different utility functions. A particular utility function may be used for different types. For example:

```
select-uf(scale, map, road) = medium-detail(scale.value)
select-uf(scale, map, house-numbers) = fine-detail(scale.value)
select-uf(scale, map, traffic) = fine-detail(scale.value)
select-uf(survey-date, map, *) = pref-recent(survey-date.value)
```

The utility functions over scale may describe a perception that the loss of utility with increasing scale is greater for house numbers and traffic than for roads. Similarly a hiker will want considerable detail to navigate safely through an area with cliffs, whereas a loss of precision in the description of a road is less likely to be dangerous. The utility function over the survey date would indicate that up-to-date data are to be preferred.

6.3.5 Goals

The final part of the specification is rather simpler. This part describes goals, or constraints, on resources to be applied over the whole document retrieval. We have used goals of download deadline, maximum data density on screen and maximum cost in the map application. Where throughput is limited with respect to the download deadline this has the effect of constraining the variants selected. A goal may be represented in a similar way to weights and utility functions:

```
goal: Resource → Value
goal(deadline) = 15
goal(cost) = 50
```

It is also possible to define goals as functions in terms of other values known during the selection process. For example, we know the maximum weight for any element which has a variant selected at the point when the selection process negotiates resources. If the goal is a function of maximum weight then a selection containing more interesting data may be given longer to download:

```
goal: Resource → function(selection-parameter)
goal(deadline) = (10 + 10*max-weight)
```

6.3.6 An Example Specification

We will discuss selection mechanisms in detail in chapter 7, but for now we assume that a meta data description can be queried from the data sources, before requesting the actual data. This meta data will provide: a description of the data *types*, attributes of variants, cost of download and size of data (bytes). The selection process follows an algorithm which treats differing types of data according to their weights and selects amongst the variants of each element according to their utilities. The algorithm is specified to maximise the perceived utility to the user within the resources available due to the goals.

In addition to meta data we assume that near the device, or proxy, making the selection there will be some model of available resources which the goals may be applied to. In the case of deadline, we assume that a model of network throughput and round trip time to the various sources is maintained and that network throughput may be reserved. While this reservation may not be end-to-end, or provide guarantees (and will certainly only be an estimate in the case of wireless networks), we have found that this provides sufficient limitation to provide a reasonable restriction of download time and variance. We have verified this using simulated links with characteristics similar to wireless networks(6.6kb/s to 1Mb/s) and reasonable deadline ranges (10s to 60s).

In figure 6-3 we illustrate a user specification, giving type weights, a deadline goal, utility functions to be applied over the scale parameter for different types and a single utility function to be applied over the survey date. The type weight specifications describe a preference for road features over water features. Here the most specific weight for a motorway is for road, which is a super-type of motorway. The utility functions over scale may describe a perception that the loss of utility with increasing scale is greater for topographical details than for roads. For instance, a hiker will want considerable detail of rock features to navigate safely through an area with cliffs, whereas a loss of precision in the description of a road is less likely to be dangerous. Similarly the utility function over the survey date would indicate that up-to-date data are to be preferred.

Our use of weights, utility functions and goals may lead to a somewhat complex description of requirements, which are inappropriate for “average” users to create or edit, however we believe that having specialist rule creators, with editing or feedback facilities, would be an appropriate solution in most cases. A restricted problem domain assists in the specification of effective preferences.

6.3.7 Relationship with Existing Work

Much of the work on adaptation relies on resource models and models of trade-off. We shall now discuss some of the work in this area which has particular relevance to our work. The basics of resource management and utility functions are sufficiently well established not to require a discussion here.

The use of utility functions across multiple media dimensions to control QoS adaptation is described by Walpole, Krasic et. al. in [115]. The system of utility functions with thresholds we describe is similar to the utility function they describe. The model they describe concentrates on providing QoS for a video stream, whereas our approach concentrates on multi-element media. This introduces another level of potential trade-off between those elements, in addition to changes in quality along the various dimensions in any one element.

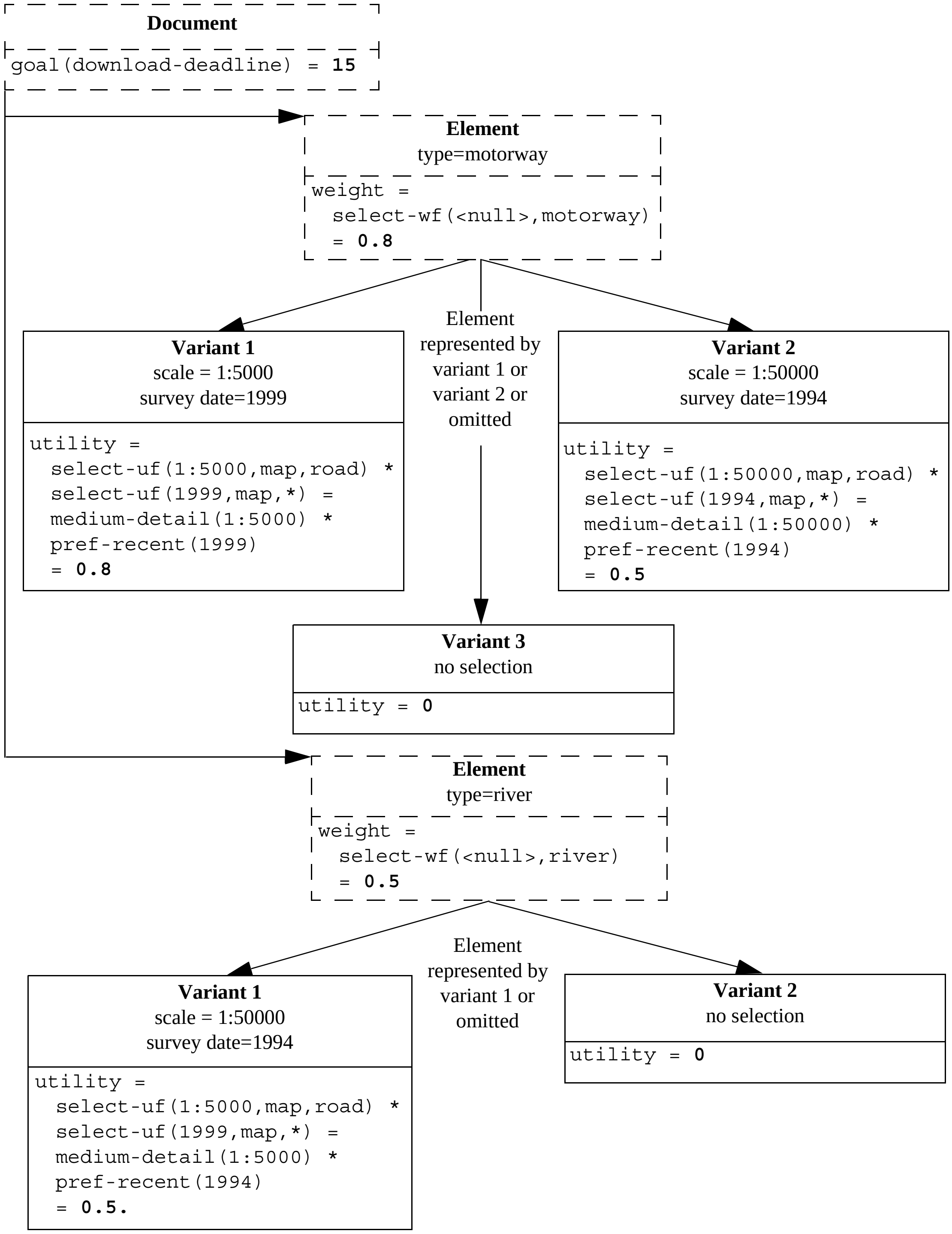


Figure 6-3: An Example of the Application of a Specification

Along with [12,57,115] we are using a performance space rather than a single degradation path in our adaptation. Our approach is to select a variant among a set of discrete points in a performance space, the combination of attributes described for a variant matches a point in the space. Others have described the more general case of freely varying attributes of the media within a performance space. The use of a degradation path appears often in the literature. Where media quality (or resource use) is only varied in one dimension a degradation path is the simplistic case of a 1-dimensional performance space, e.g. Witana, Fry and Antoniades [125].

In [85] an overview of approaches to adaptability is presented by Ma et. al. and a suggestion for linking perceived quality to deadline specification for web browsing is described. Their approach to deciding how to adapt is based on allocating percentages of the data the user is willing to use, for a given element. We believe our approach of rating the utility of variants by measurable attributes provides a finer degree of control than their assumption that any given proportion of data volume corresponds to a known (and acceptable) degradation in quality.

In both weight and utility there need be no meaning attached to any value taken. The only meanings we attach, to both weight and utility, are that zero indicates “outside the acceptable region”; and that one indicates the “upper limit of discernable improvement”. As noted by Watson and Sasse in [118] ratings given in language attach a cultural interpretation to the values. We desire to indicate only “better than” or “worse than”, with some degree indicated by the difference in value. Some selection algorithms will interpret a large difference in value differently to a small difference, others will ignore this and simply interpret an order.

6.4 BUILDING CONTEXT DRIVEN SPECIFICATIONS

Various parts of the specification required at any time will be due to the many factors associated with context, such as: location, co-location, social situation, task and screen size. Each of these *context aspects* will have some effect on some part of the overall specification. However there may be a large number of combinations of these context aspects which may be met by any given user, e.g., a user may use several devices each day and undertake various tasks. Clearly anticipating all combinations of context aspect which might be met and defining specifications tailored for each is not practical. Instead we seek to describe the specifications due to each of these aspects of the context. So, a single specification would be written for each device commonly used, e.g., desktop, laptop, PDA, mobile phone. Another specification for each task engaged in, e.g. various jobs at work, shopping, commuting. Yet another for co-location with groups of people, e.g., colleagues, customers, in-public, alone. Specifications may also be written for contexts which are defined by combining context information from different aspects, each set of specifications reflecting a subset of context. The specifications would be combined to reflect the total limitations and preferences in any given overall context. We now describe our approach to building specifications due to context.

Further examples of how we envisage context to affect utility functions, weightings and goals are given below:

- When the cost of the link is great, e.g. for mobile phones, goals can be defined to reduce cost, possibly correlating cost to overall utility. The assumption being that one is prepared to pay for more interesting data.
- Location and proximity information may cause changes in behaviour. For instance, “in the car” may be identified as placing some urgency about fetching map data. Utility functions, encoding format and type weightings may also be varied to achieve more suitable data presentation.
- When using a screen of restricted colour capability, the utility functions over colour depth may reflect this. Rather than taking the user’s limit for perception of colour depth, the screen’s ability to render colour defines the limit.
- When driving a car, audio data (or text which can be rendered as speech) may be preferred over visual data. Where data has to be graphical, key information will be rendered as highly important but detail will quickly be described as unimportant, to avoid distraction.

6.4.1 Specifications for Context Aspects

We wish to make our specifications context driven, which requires that we are able to describe context. We use a total context, *C*, as a combination of *context aspects*. The values each of these aspects take will be determined by the aspect they describe, e.g., location may be an area which one is within, screen size a rather more precise pair of values.

We wish to associate a specification with certain conditions in one or more aspects of context. A specification will be applicable when one or more context aspects have certain values, or are within or beyond some value. We use a set of matching functions, *M*, to achieve this. *M* may contain a matching function for each of a subset of those aspects in the description of the total (known) prevailing context, *C*.

A specification, *S*, is then the combination of its context conditions, *M*; and the three parts of the specification: the weight function selectors, *W*; utility function selectors, *U*; and goals, *G*; which are applicable under those context conditions.

Whether the specification is applicable depends on the context conditions being met. This may not be as simple as an equality relation as the aspect may not take a simple numeric type – it may describe abstract notions, or geographic spaces. Specifications are applied if *all* the context conditions are met according to their matching function by some context aspect in the overall context. The matching function may optionally define whether an unknown value matches true or false (false being default).

```
Spec: ([c-match], [select-wf], [select-uf], [goal])
c-match: (Context-Aspect, Context-Value) → Boolean
```

As an example, consider the following specifications:

a. Here we describe some preferences which may be due to driving a car at a moderate speed. Units for speed and deadline are assumed here for simplicity:

```
M:      c-match(speed, c-speed) = (30 <= c-speed <= 100)
        c-match(location, c-location) = (c-location==in-car)
W:      select-wf(<null>, road) = 0.6
        select-wf(<null>, motorway) = 0.8
        select-wf(<null>, footpath) = 0
U:      select-uf(scale, map, road) = medium-detail(scale.value)
G:      goal(deadline) = 15
```

b. Next, we illustrate a specification reflecting different preferences when moving slowly:

```
M:      c-match(speed, c-speed) = (0 <= c-speed < 30)
W:      select-wf(<null>, *) = 0.4     // a default for any type
        select-wf(<null>, road) = 0.85
        select-wf(<null>, motorway) = 0.2
        select-wf(<null>, footpath) = 0.5
U:      select-uf(scale, map, road) = high-detail(scale.value)
G:      goal(deadline) = 20
```

c. The screen size of the device in use may also affect the presentation of data, as highly detailed map data would be lost on a small screen, which might be encoded as below. Note also that a specification need only include one part (weight, utility function selector or goal) and the context-match.

```
M:      c-match(screen-size, c-ssize-x) = (60 < c-ssize-x < 320)
        c-match(screen-size, c-ssize-y) = (60 < c-ssize-y < 240)
U:      select-uf(scale, map, *) = general-detail(scale.value)
```

d. A general specification relating to work may be defined:

```
M:      c-match(task, c-task) = (c-task == work)
W:      select-wf(<null>, *) = 0.4
U:      select-uf(age, map, *) = pref-recent(age.value)
G:      goal(cost) = 50
```

e. This specification may be specialised for sub-tasks of work. For instance for a driver making deliveries:

```
M:      c-match(task, c-task) = (c-task == delivery-driver)
W:      select-wf(<null>, road) = 0.85
        select-wf(<null>, traffic) = 0.5
        select-wf(<null>, house-numbers) = 0.6
        select-wf(distance, house-numbers) = strong-pref-near(distance.value)
U:      select-uf(scale, map, house-numbers) = fine-detail(scale.value)
        select-uf(scale, map, traffic) = fine-detail(scale.value)
G:      goal(deadline) = 20
```

f. Match functions may be defined to compare aspects of context. For instance if the screen is square one may prefer square icons:

```
M:    c-match(screen-size, (c-ssize-x, c-ssize-y)) = (c-ssize-x == c-ssize-y)
U:    select-uf(isize, image, icon) = pref-square(isize.x, isize.y)
```

6.4.2 Relationship with Existing Work

We believe there is a need for a more flexible solution than the provision of standard translations or restrictions according to device parameters, as described in [53,59,73,88,99] (Fox, Holtman, Klyne, Mohan, Reynolds et. al). Our approach has the advantage that use is not tied to particular classes of devices and does not require separate effort to tailor documents to specific devices. Being more general this approach facilitates the use of emergent devices, while allowing specialised data to be used where the effort has been made to provide it. These specialised versions may be part of the solution, particularly for very small devices. However, when using devices of the next level of capability (PDA to laptop, kiosks, in-car systems, etc.), the device itself ceases to be an absolute limitation. The user interface and network connection are often both sufficient to support more than the most basic data presentation, although still being limited in comparison to desktop PCs. Maintaining a user-driven specification, while offering any available dynamically generated assistance, can better support device and user centric adaptation of the data presentation.

A discussion of the use of utility functions across multiple media dimensions to control QoS adaptation is described by Walpole et. al. [115]. The model they describe concentrates on providing QoS for a video stream. Our approach contrasts with this in concentrating on multi-element media. In both [78,115] the use of utility functions to describe user preference and system limitations over parameters of media is similar to our own.

Our approach of selecting data according to its properties has similarities to the Ma et. al's approach in [85], they describe an interplay of willingness to degrade data against urgency. Their approach assumes that the quality of the data can be inferred from its size. While this relationship may hold in the case of images, it is not true for all data. We do not regard the selection process as being solely about degrading data, which is a common starting point, e.g. [53,93]. In many cases presenting all offered data is likely to be overwhelming and contain much irrelevant information, in this situation omission does not constitute a degradation, from the user's perspective. Similarly we measure utility across the user's perception, rather than in relation to some ideal or original version or as a function of data size, as seen in [88] (Mohan et. al.).

6.5 COMBINING SPECIFICATIONS

Given that one has several specifications which are relevant at any given time, how does one combine them? We shall examine our approach in this section.

6.5.1 Unrelated Parts

Where terms in two specifications would not be applied to the same attributes of a variant both can clearly form part of the final specification. e.g: `goal(download-deadline)` and `goal(cost)` are both used in the overall specification as there is no overlap between the two definitions' scope. However, there are many cases where two specifications may overlap. Resolving whether these specifications can be combined, or the conflict should be resolved one way or the other, is then necessary.

6.5.2 Specialisation of Specifications

Two specifications may operate such that one is more specific, in terms of context, than the other. That is if all context aspects are that are matched in the more general specification are matched in the specialised one. Where the context takes a range of numerical values the value range must be equal to or within the general one. Where the context takes a value from a tree of conceptual values the value in the specialised specification must be equal to or a descendant of the one in the general specification. In at least one value the specialised specification must be more specific than the general one, this may be achieved by matching on a context aspect which is not used in the general specification. In this case it is sufficient to take the most specific which defines each part. Where a general specification is specialised, e.g. “delivery” specialises “work”, any terms in the more specialised specification override where there is a conflict.

6.5.3 Overlapping Specifications

In addition to the cases above, there may well be specifications deriving from different aspects of context, referring to specifications against the same part, e.g., different weights for the same type, different utility functions for the same attribute, with no difference in specificity, but a difference in the outcome of their application. Which should be chosen?

Placing an order over the context aspects is somewhat arbitrary and may be hard to specify: each specification may be due to a number of context aspects. We take a combination of generous and conservative selection amongst options and combination of options for the different parts of the specification, once elimination of specifications handling a specialisation of context and a few special cases are addressed.

a. For type weights, for each weight function parameter:

- Where one weight refers to a sub-type of another weight, e.g., motorway and road, both are used in the overall specification. The weight for the most specific type is applied when evaluating weights for elements.
- Where the types are equal, a zero rating is honoured, otherwise the best is taken. As type weight controls element inclusion and degradation order, the generous approach seems most valid – a preference is to be honoured. The only exception is a weight of zero, which specified “absolutely unwanted”.
- Where weight functions for several different parameters (including no parameter) which come from different specifications that exist for an element, they are combined as a product.

Given a moderate “default” value, positive and negative preferences can both be described for more specific types. For example, consider the following weights, from different specifications:

```
select-wf(<null>, road) = 0.6
select-wf(<null>, road) = 0.85
select-wf(<null>, motorway) = 0.8
```

When applied to a road, the weight used would be 0.85 (taking the maximum). When applied to a motorway, 0.8 would be used (taking the most specific type).

b. For utility functions, all the applicable functions for an attribute are combined by product. Any zero utility result will clearly override all other values. The lower value that a greater number of products (of values between 0 and 1) tends to give is not a problem as the utility is used to differentiate between variants within a type. All variants will have the same specifications applied to them, so the effect will be consistent within each type. Where the variants have differing numbers of attributes described, some normalisation may be necessary.

For example, consider the following utility functions selectors, which derive from unrelated specifications and which can be used to derive *u(scale)*:

```
select-uf(scale, map, *) = general-detail(scale.value)
select-uf(scale, map, road) = medium-detail(scale.value)
select-uf(scale, map, motorway) = low-detail(scale.value)
```

When applied to an element of type “road”,

```
select-uf(scale, map, road) = medium-detail(scale.value)
```

as type “*” is not equal or a super-type of “road” and “motorway” is a sub-type.

When applied to an element of type “motorway”,

```
select-uf(scale, map, motorway) =
  medium-detail(scale.value) * low-detail(scale.value)
```

as type “*” is not used, as before, however “road” is a super-type of “motorway” and so is relevant. If the use of medium-detail for road and low-detail for motorway derived from the same specification the result would be:

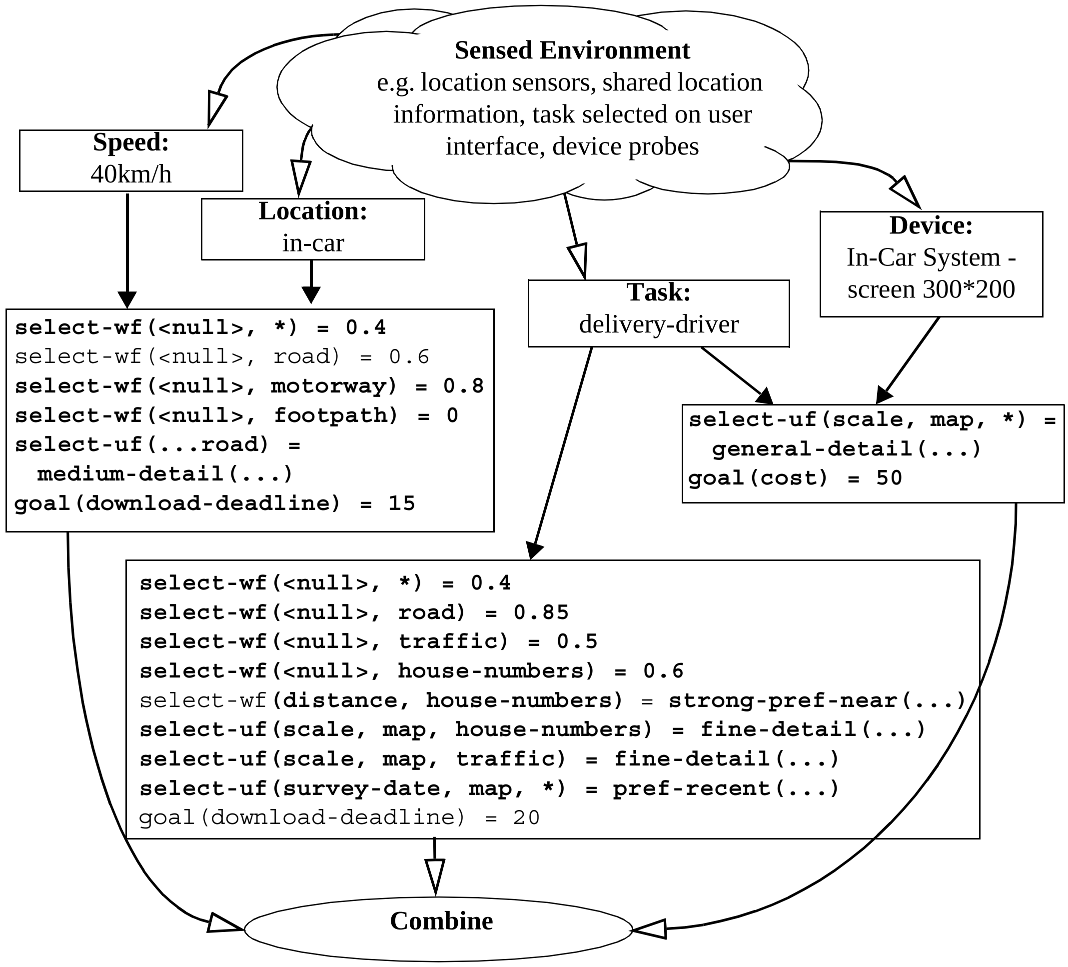
```
select-uf(scale, map, motorway) = low-detail(scale.value)
```

c. For goals, we take the lowest value for each resource and parameter, such as deadline-fixed and deadline-due to max. weight, price-fixed, etc. This will then reflect any limitations in time, cost, screen space etc. which are being described. Where goals are functions the final evaluation of the selection of which goal to apply may have to be delayed until the parameters of those functions are known.

Where multiple goal functions are available for a parameter we sum the goals for each different parameter. Where multiple goal functions are available, we combine them as follows:

a. Select the minimum value for a given function parameter. This ensures that limits being described are adhered to, even if more generous alternatives are available.

b. Sum the goals according to different parameters. This is largely a matter of convenience in writing specifications. It allows a base limit to be set separately from any flexible addition due to maximum weight, or other parameters. Any function giving a variable result does not then override or assume anything about basic limits set.




```
select-wf(<null>, *) = 0.4
select-wf(<null>, road) = 0.85
select-wf(<null>, motorway) = 0.8
select-wf(<null>, traffic) = 0.5
select-wf(<null>, house-numbers) = 0.6
select-wf(distance, house-numbers) = strong-pref-nearer(distance.value)
select-wf(<null>, footpath) = 0
select-uf(scale, map, *) = general-detail(scale.value)
select-uf(scale, map, road) = medium-detail(scale.value)
select-uf(scale, map, house-numbers) = fine-detail(scale.value)
select-uf(scale, map, traffic) = fine-detail(scale.value)
select-uf(survey-date, map, *) = pref-recent(survey-date.value)
goal(download-deadline) = 15
goal(cost) = 50
```

Figure 6-4: Selection and Combination of Specifications Due to Context
Bold in specifications indicates function used after combination

Figure 6-4 illustrates the translation of sensed data into context attribute definitions and how selection then combines the relevant specifications. The specifications extend those given as examples throughout this thesis. The combination of specifications has resulted in a specification which can be applied as described in section 6.3. The bold specifications highlight those which are used.

We do not expect non-technical end-users to write these specifications. They could be provided by specialists, from user studies and a user might tune parameters, especially goals. Initially some aspects of context would require user input, however eventually we assume sensors could detect most of these automatically. Similarly we have not considered specifications which respond to user feedback. The acquisition of user reactions to maps was beyond the scope of our work and is of questionable applicability where the use of the device is not the user's main focus of attention. In any case, our models of specification could be modified by user feedback if this was desired.

6.5.4 Uncertainty in Context

In chapter 5 we discussed the possibility that the sensed context may not be certain. This may reflect the accuracy of sensors, age of readings or our trust in the sensors. Corroboration between sensors may give rise to greater belief in a reading, although often a decrease in accuracy. A greater problem exists for us where two possible values are provided for a context aspect, each of which would cause different specifications to be active. There is a good chance however that one will be inappropriate. We have not addressed these issues in our prototype application, but our specifications should be able to accommodate this somehow. As a starting position for future research we would suggest allowing both active specifications to be combined as for other active specifications. Refinements may include preventing aspect values with much lower confidence than others offered activating specifications; and assigning "safety conditions" to context aspects, to indicate that certain values should override alternative values where detection is inconclusive. Whether a person in a car is a driver or a passenger is a possible case for this – the case of the driver should dominate for safety reasons. This issue highlights the need for accurate sensors of context and provision of user override.

6.6 Formal Presentation of Combining Specifications

We now present a more formal description of how specifications are defined and how their combination may be achieved.

6.6.1 Specifications

We use a total context, *C*, as a combination of context aspects. The values each of these aspects take will be determined by the aspect they describe. In eqn.6.1 c_i is the value for aspect *i*.

$$C = \bigcup_{\forall \text{aspects, i}} c_i \qquad \textbf{(Equation 6.1)}$$

A specification will be applicable when one or more context aspects each satisfy some matching functions. The aspects which shall be considered when assessing a specification's applicability, will be a subset of those in the description of the total (known) prevailing context, *C*. A specification, *S*, as in eqn.6.2, is then the combination of its context conditions, *M*; and the weights, *W*, utility functions, *U* and goals, *G*, which are applicable in that context.

$$\langle M, W, U, G \rangle \qquad \text{where} \begin{pmatrix} M = \bigcup c - match_{aspect} \\ W = \bigcup weight - selector_{attribute,\, type} \\ U = \bigcup uf - selector_{attribute,\, format,\, type} \\ G = \bigcup goal_{resource} \end{pmatrix} \qquad \textbf{(Equation 6.2)}$$

Whether the specification is applicable depends on the context conditions being met. Specifications are applied if *all* the matching functions return true given the context conditions prevailing, eqn.6.3.

$$\text{specification S is valid} \leftrightarrow (\forall j \text{ in M}), \exists i \text{ in C} \rightarrow m_j(c_i) \qquad \textbf{(Equation 6.3)}$$

6.6.2 Combining Specifications

6.6.2.1 Unrelated Parts

Where terms in two specifications would not be applied to the same variants, or attributes of the same variant, both can clearly form part of the final specification. These can clearly be combined by union, without further consideration, as in eqn.6.4. We require that: weights refer to types which are not equal to or derived from each other, utility functions (applied to the same type) refer to different attributes and that goals define limits over different resources such as deadline or cost.

$$\begin{aligned}&\text{include}(w_{ti} \cup w_{tj}) \leftrightarrow ti \not\subset tj\\&\text{include}(u_{fi,\,ti,\,pi} \cup u_{fj,\,tj,\,pj}) \leftrightarrow ti \not\subset tj \vee pi \neq pj\\&\text{include}(g_{pi} \cup g_{pj}) \leftrightarrow pi \neq pj\end{aligned} \qquad \textbf{(Equation 6.4)}$$

6.6.2.2 Specialisation of Specifications

Two specifications may operate such that one is more specific in terms of context than the other. In this case it is sufficient to take the most specific which defines each part. Where a general specification is specialised, e.g. “work to “delivery”, any terms in the more specialised specification override where there is a conflict.

Specialisation is the case where (line by line in eqn.6.5):

- S_x<S_y is used to denote “S_x is a specialisation of S_y” and is the case iff:
- All common aspects of context take an equal or more specific value.
- Any aspects in the more specialised specification which are not equal or more specific take no value in the less specialised specification.
- The more specific context must take equal or more specific values for all aspects in the less specific specification.
- At least one aspect in the more specific specification must take a more specific value than in the more general specification (including aspect not used).

Note, the definition of order between matching functions is optional. Where the match is on a numeric value range “equal-or-more-specific” may be inferred. Where the order between matching functions cannot be derived the order may be given explicitly, or simple “equality” used, rather than “equal-or-more-specific”.

$$\begin{aligned}&(S_x < S_y \leftrightarrow\\&(\forall m_i \in M_x)(\exists m_j \in M_y, \text{equal-or-more-specific}(m_i, m_j) \vee\\&\neg\exists m_j \in M_y, (i = j)) \wedge\\&(\forall m_j \in M_y)((\exists m_i \in M_x, \text{equal-or-more-specific}(m_i, m_j)) \wedge\\&(\exists m_i \in M_x)((\exists m_j \in M_y, \text{more-specific}(m_i, m_j)) \vee \neg\exists m_j \in M_y, (i = j))))\end{aligned} \qquad \textbf{(Equation 6.5)}$$

6.6.2.3 Overlapping Specifications

There are many cases where two specifications may overlap. Resolving whether these specifications can be combined, or the conflict should be resolved is then necessary.

a. For type weights (line by line of eqn.6.6):

- For an attribute, where specifications specialise, the more specific is used, as defined in eqn.6.5. This is the first test for utility functions and goals too.
- For an attribute, where one weight refers to a sub-type of another weight, e.g., motorway and road, the weight for the most specific type should be applied.
- For an attribute, where the types are equal a zero rating is honoured, otherwise the best is taken.
- Then, take the product of the weights for all attributes.

$$\text{include}((w_{ai,\,ti} \in S_i), \neg(w_{ai,\,tj} \in S_j)) \leftrightarrow S_i < S_j, \text{ otherwise}$$
$$(\text{include}(w_{ai,\,ti}, \neg w_{ai,\,tj}) \leftrightarrow ti \subset tj)$$
$$\text{include}(w_{ai,\,ti}, \neg w_{ai,\,tj}), (ti = tj) \leftrightarrow \begin{pmatrix} w_{ai,\,ti} = 0 \\ \text{otherwise}(w_{ai,\,ti} > w_{ai,\,tj}) \end{pmatrix}$$
$$\left(weight = \prod_{\forall \text{attributes}\, i} w_{ai,\,t} \right), \text{ where } w_{ai} \text{included only once from each specification} \quad \textbf{(Equation 6.6)}$$

b. For utility functions, all the applicable functions for an attribute are combined by product. Any zero utility result will clearly override all other values.

$$\text{include}((u_{fi,\,ti,\,pi} \in S_i), \neg(u_{fj,\,tj,\,pj} \in S_j)) \leftrightarrow S_i < S_j, \text{ otherwise}$$
$$utility = \left(\prod_{\forall \text{u where}((tj \supseteq te), (pi = pj), \text{ where te is type of element})} u_{fj,\,tj,\,pj} \right) \quad \textbf{(Equation 6.7)}$$

c. For goals, we take the lowest value for each resource, such as deadline. This will then reflect any limitations in time, screen space etc. which are being described.

$$\text{include}((g_{pi} \in S_i), \neg(g_{pj} \in S_j)) \leftrightarrow S_i < S_j, \text{ otherwise}$$
$$\text{include}(g_{pi}, \neg g_{pj}) \leftrightarrow g_{pi} \le g_{pj} \quad \textbf{(Equation 6.8)}$$

6.7 SUMMARY

In this chapter we presented our techniques for describing needs and preferences which arise from context. The generality of functions over semantic types, properties of encodings and limits in resources enables a rich specification of requirements. The specifications result in a *selection space* in which the data offered can be considered. The approach is equally applicable in domains where data are in many formats as it is where all data are in one format. Similarly, by retaining an abstract form of context our approach is designed to allow a flexible description of context. Where aspects of context are not relevant they need not be considered. Where aspects of context emerge as relevant factors, or aspects evolve to take new ranges of values, the specifications may be extended to reflect these without excess effort.

The specification described makes few assumptions regarding the data offered: that the data will be described using a known semantic type system and that the parameters described will conform to those over which utility is specified. The meta data language may be extended to describe format and data-class specific concepts, such as map scale or JPEG quality. Where this data is not available a best effort application of specifications is possible. The use of parameters of elements in weight functions increases our ability to reflect the benefit of data to the user. The use of type information in selecting utility functions offers great flexibility in ordering variants of different elements.

While we do not have the degree of control typing specifications to a specific document might offer, the result of this is that a set of specifications may be applied over a wide range of data without detailed prior knowledge of what data will be offered. This removes a substantial burden from the author of the data. They need only describe the offered document and any pre-generated variants. The example specifications shown can easily be applied over many map tiles and still encode needs and limitations appropriately and accurately. The transfer of the burden of control onto specification writers is the payoff for this. It is to be hoped that specifications can be reusable and so the total effort is no greater than for authoring by the data provider. The selection space offered data is placed in and the trade-offs made in the selection shall be covered next, in chapter 7.

A description of the XML encoding for our specifications and the larger example of specifications used in testing can be found in appendix 2.3.

CHAPTER 7 PERFORMING MEDIATION

In this chapter we describe selection techniques which we have used to perform the mediation. Again, we extend our descriptions in a somewhat formal manner.

Our specifications provide an ordering over the various elements of a document and an ordering over the various variants which may describe that document. A best selection is the initial stage in the selection process: all elements with non zero weight have their highest utility variants selected and resources to satisfy this selection are applied for. If this reservation succeeds then the data are requested. However, it will often be the case that the total resource requirement of the best version of all somewhat interesting data will exceed the system's ability to satisfy the stated goals, in this case a selection process is required. We consider two algorithms below which offer variations on a theme. In both cases the algorithms form a degradation path which models degradation of the selection. Degradation is defined in terms of the weights and utilities derived from the specifications. The process of degradation will generally move towards satisfying resource limits. The issues in these algorithms are described below.

7.1 SELECTION ALGORITHMS

We have worked with two *selection definitions*. One we shall refer to as the *sequential* definition and the other as the *cost based* definition. To illustrate these two cases, consider the two elements and their variants, as described in figure 7-1.

7.1.1 Sequential Selection Algorithm

The sequential definition requires that the order between weights describes the order in which degradation is applied across elements and that the utility order of variants describes the order in which variants are selected for an element. Allowance is made so that unnecessary degradation is not made.

In table 7-1 we consider the basic case of two elements with four variants each, as shown in figure 7-1. The reasoning can be generalised from this to other more complex situations: the element with the lower weight is degraded first, each having variants selected in order of decreasing utility. Step 0 shows the best variant of each element selected, which is the position for the initial application for resources.

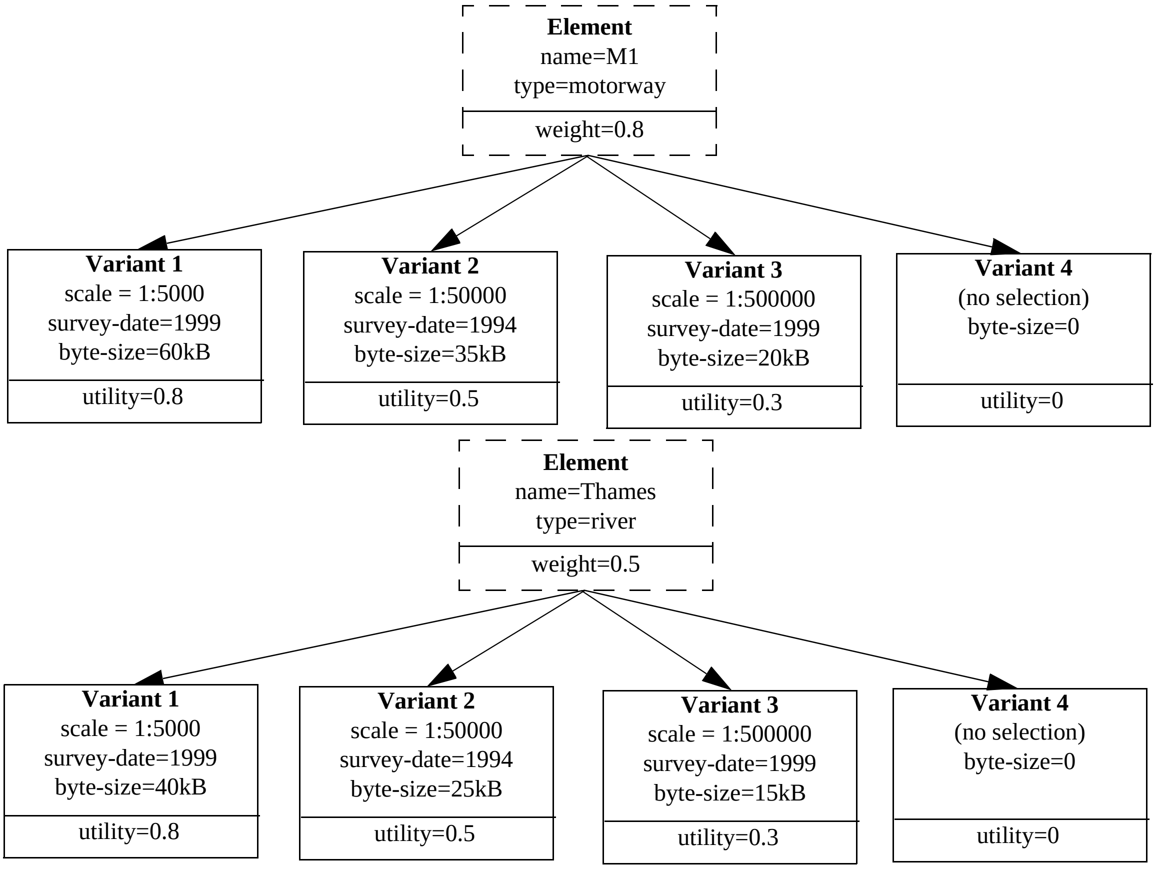


Figure 7-1: Example Elements and Variants for Selection Algorithms

Step	Element “M1” Variant Selected	Element “Thames” Variant Selected	Total Size
0	Variant 1	Variant 1	100kB
1	Variant 1	Variant 2	85kB
2	Variant 1	Variant 3	75kB
3	Variant 1	Variant 4	60kB
4	Variant 2	Variant 4	35kB
5	Variant 3	Variant 4	20kB
6	Variant 4	Variant 4	0

Table 7-1: Degradation Path, Sequential Selection

We shall now consider a few selection scenarios:

- If the maximum size allowed (e.g. due to network characteristics and deadline) is 80kB, the selection would return at step 2, where “M1” is realised with variant 1 and “Thames” realised by variant 3. No improvement is possible within the unused 5kB.

- If the maximum size allowed is 50kB, the selection would return at step 4, where “M1” is realised with variant 2 and “Thames” is not realised. However, there are 15kB unused. “M1” cannot be improved without breaking the goal, but “Thames” can be improved to variant 3. We address this improvement in section 7.1.7.
- If the maximum size allowed in 30kB, the selection would return at step 5, where “M1” is realised with variant 3 and “Thames” is not realised. No improvement is possible within the unused 10kB.

7.1.2 Cost Based Selection Algorithm

The cost based definition performs degradation based on the perceived cost of the degradation. Simply put, the cost is the product of the weight of the element and the loss of utility from the best variant for that element:

$$\text{cost}(\text{variant}_n, \text{element}) = \text{weight}(\text{element}) \times (\text{utility}(\text{variant}_1) - \text{utility}(\text{variant}_n))$$
$$\text{where variant}_1 \text{ is the best variant for element}$$ **(Equation 7.1)**

A degradation step, even of the most important type of element, need not be regarded as a great loss if the utility in the representation changes only slightly. Where two degradations of the same cost occur, the degradation with the lower weight is performed first. Where both the cost and the weight are equal, the one with the lower initial utility is degraded first, if a tie still exists the choice is arbitrary.

Taking the same example as before, we show the sequence of selections in table 7-2. Step 0 is still the case where the highest utility variant for each element is selected. The degradations are taken in order of rising cost.

Step	Element “M1” Variant Selected	Element “Thames” Variant Selected	Cost of Degradation	Total Size
0	Variant 1	Variant 1	-	100kB
1	Variant 1	Variant 2	0.15	85kB
2	Variant 2	Variant 2	0.24	60kB
3	Variant 2	Variant 3	0.25	50kB
4	Variant 2	Variant 4	0.4 (lower weight)	35kB
5	Variant 3	Variant 4	0.4 (higher weight)	20kB
6	Variant 4	Variant 4	0.64	0

Table 7-2: Degradation Path, Cost Selection

Reconsidering the previous selection scenarios:

- If the maximum size allowed is 80kB, the selection would return at step 2, where “M1” is realised with variant 2 and “Thames” realised by variant 2. “Thames” may be improved to variant 1 with the unused 20kB.
- If the maximum size allowed is 50kB, the selection would return at step 3, where “M1” is realised with variant 2 and “Thames” is realised with variant 3. No data is unused.
- If the maximum size allowed in 30kB, the selection would return at step 5, where “M1” is realised with variant 3 and “Thames” is not realised, as before. No improvement is possible within the unused 10kB.

In the case of the 50kB limit, if the time taken to make the selection had affected the data size allowed, or in any case if the selection had affected the goal (e.g. where goal is a function of the selected data) the selection process can be re-run, continuing from the current selection.

7.1.3 Degradation Paths

In both algorithms the degradation is performed over a sequence of steps. Each step represents the progression of an element from one variant to another. The path is formed once the need to degrade from a best selection has been identified. Each element is considered in turn. Its variants are taken as pairs from the ordered sequence of variants which realise the element. In the sequential algorithm the degradation path is traversed until the weight position of the element is found. Then the steps are inserted in order of the lower utility in the step. In the cost based algorithm the degradation path is traversed until the cost position of the step is found. Then the steps are inserted in order of weight and for a given weight in order of the lower utility in the step. The sequence of degradation steps is illustrated in figure 7-2.

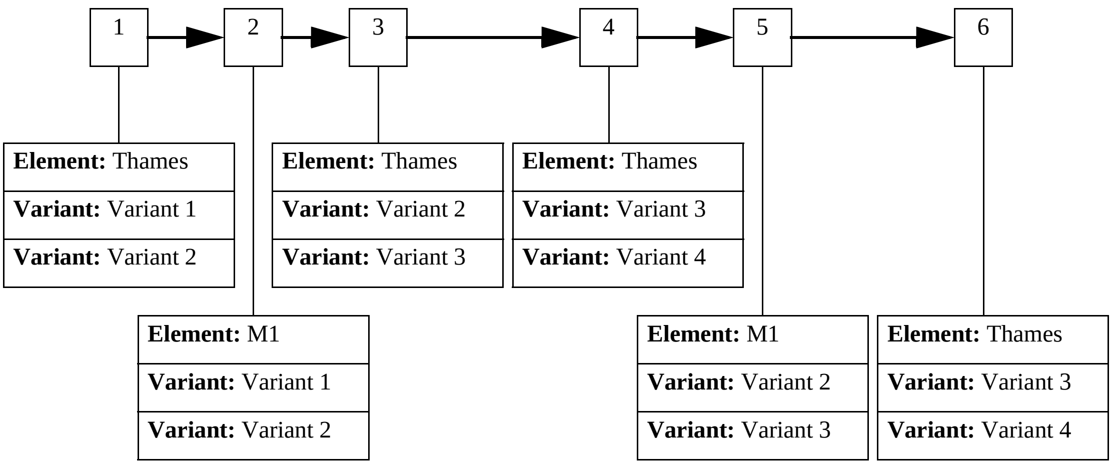


Figure 7-2: Degradation Path Example, Cost Based Algorithm

7.1.4 Grouping Elements for Degradation

A refinement of these algorithms may consider degradation steps formed of more than one element changing variant. For instance, it would be natural to consider all variants of elements with the same weight being degraded as a group to provide a consistent quality rendering.

The map application developed implements this grouping. The degradation path is formed of groups of variant steps, to be applied together rather than from single variant changes. In the sequential algorithm the groups are formed from elements of the same weight progressing to the same utility. In the cost based algorithm the groups are formed from elements of the same weight, making the same cost progression to variants of the same utility.

The grouping has two main effects:

a. The presentation of elements of the same type becomes much more consistent. This will improve user perception as the behaviour becomes much more predictable and feature inclusion or omission becomes more consistent.
b. The selection algorithm becomes less likely to fill the available resources. The granularity of resource use in the available selections becomes larger as steps are combined.

The benefit of consistency is expected to outweigh the extra degradation which may be applied to meet limits. Where the data offered displays sufficient differences then specifications will draw sufficient differences that under-use of resources will not be a significant problem. For instance in a map there may be many minor roads. These may be differentiated by their distance from the area of interest. A potentially large resource-use and user-perception step is thus broken into several smaller steps. The difference reflects a perceived change in the benefit in the data, so this is not simply a mechanism to subvert the grouping in the selection algorithm.

7.1.5 Weight Inversion

In both cases the ability of variants to support the realisation of other variants, so that elements may contain other elements, impacts the weight given to elements at some degradation stages. Where the transition from variant-a1 to variant-a2 of element-a causes element-b, contained by element-a to be lost the weight of element-b is assumed by element-a at this point. Hence loss of data through the loss of its container only occurs when the most important part of the structure is being lost naturally. See figure 7-3 for an illustration of this handling of *weight inversion*.

In figure 7-3 and table 7-3, variant 1 to variant "n" for each element are shown in order of decreasing utility. Initially the highest utility variant is chosen for each element: variant1 for each. The element "tile-header" would be selected for degradation first based on either its weight relative to the element "M1", or the product of weight and utility lost (0.0001 vs 0.32). However, as "tile-header" contains "M1" its loss would also cause the loss of "M1". So, it is promoted to be degraded at the point all its contained elements with higher weights have no variant realising them. The first stage in degradation is now for "M1" to be realised by variant 2. In the next stage "M1" is degraded to the zero utility "no selection" variant, variant 3. At this point "tile-header may also be degraded to its no selection variant.

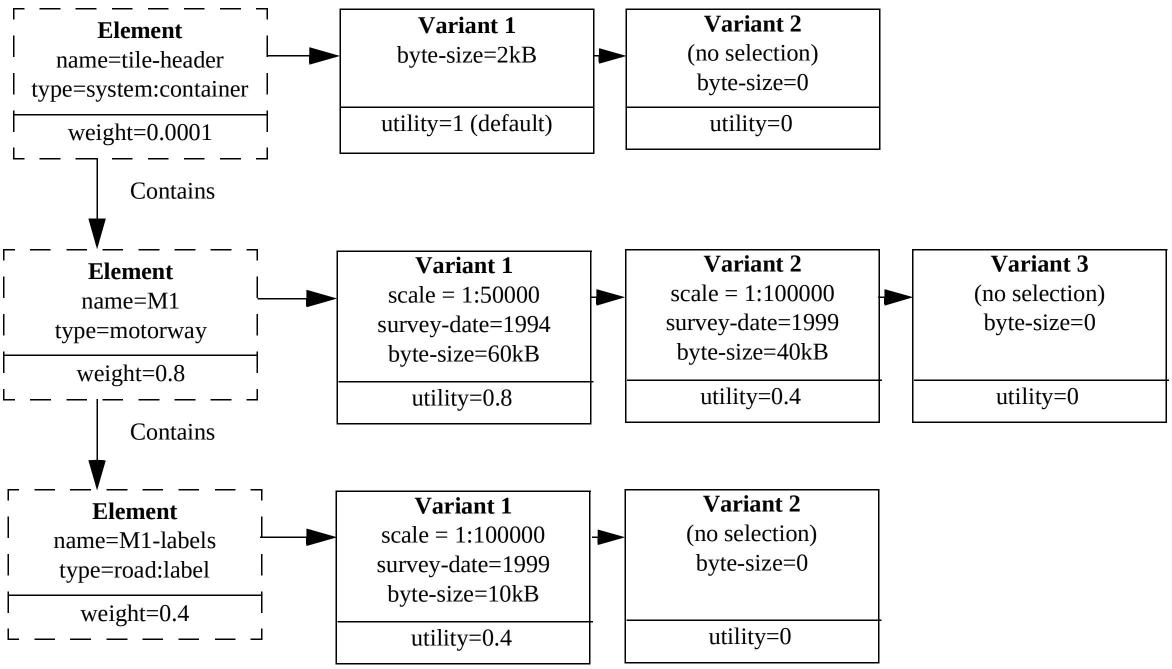


Figure 7-3: Degradation with Weight Inversion

Step	Element "tile-header" Variant Selected	Element "M1" Variant Selected	Element "M1-labels" Variant Selected	Notes
0	Variant 1	Variant 1	Variant 1	Initial best selection
1	Variant 1	Variant 1	Variant 2	"Tile-header" promoted above weight of "M1". Degrading lowest type element first
2	Variant 1	Variant 2	Variant 2	"M1" is next lowest type.
3	Variant 1	Variant 3	Variant 2	
4	Variant 2	Variant 3	Variant 2	Now no contained elements are realised, tile-header may be degraded.

Table 7-3: Degradation Path with Weight Inversion

7.1.6 Meeting Multiple Resource Constraints

As the degradation path used in selection is based on the weights and utilities of the data, meeting the resource needs occurs as a side effect. In this way multiple resource constraints can be met through one pass of the selection algorithm. The degradation path does not seek to solve any resource constraint directly, in fact a degradation step may cause an increase in resource consumption. However, all elements can eventually degrade to no selection, which has zero resource use. All degradation paths eventually lead to no data to request and so solve any resource constraint. Moving away from the extreme case, at each degradation step the change in resource requirements is applied to the total requirements in each resource dimension.

7.1.7 Reversing Degradation

Once all resource savings have been achieved resources may be re-applied for. However, in order to provide maximum utility to the user and reverse degradations which may have caused unnecessary resource loss, we perform an improvement step. Moving down the degradation path from last applied to first applied, improvements are tested against the resource requirements. Where an improvement does not break the resource limits it is applied. Once this is complete, a new set of resource reservations are attempted, to reflect the current selection.

7.2 A Formal Definition of Selections

7.2.1 Basic Definitions

A selection, A, is the definition of exactly one variant, v_i, to realise each and every element, e_i, of a document, D:

$$\begin{aligned} A &= \{v_0, v_1, \ldots, v_n\} \\ &\quad \text{where}(v_i \in \text{ variants which realise}(e_i)) \\ &\quad \forall(e_i \in \text{ elements in document}(D)) \end{aligned} \qquad \textbf{(Equation 7.2)}$$

However a valid specification should reflect the goals defined over all parameters:

$$\sum_{\forall i \text{ such that}(e_i \in D)} \text{resource use}_p(v_i) \le goal_p \qquad \forall \text{parameters}, p \qquad \textbf{(Equation 7.3)}$$

This definition of an allowable selection leaves many possible degradations of the document. We have discussed two selection algorithms, which we shall address again soon. First we shall simplify our notation a little. Eqn.7.4 gives us a shorthand, $v_{i,s}$, to describe a selected variant for element e_i. Eqn.7.5 describes a no-selection variant, $v_{i,0}$, from the set in eqn.7.4, which has no resource use of utility, u. This variant is inserted into each element's variant set by the system. The variants in the set are ordered, with $v_{i,0}$ the no selection variant and $v_{i,b}$ the best variant, as described in eqn.7.6.

$$\text{selected variant } v_{i,s} \in \{v_{i,0}, v_{i,1}, \ldots, v_{i,b}\} \quad \text{such that } (v_{i,a} \in \text{ variants which realise}(e_i))\forall(a, 0 \le a \le b) \qquad \textbf{(Equation 7.4)}$$

$$\text{no selection variant } v_{i,0} \in \{v_{i,0}, v_{i,1}, \ldots, v_{i,b}\} \text{ such that: resource use}_p(v_{i,0}) = 0 \ \forall \text{parameters}, p \text{ and}(u(v_{i,0}) = 0) \qquad \textbf{(Equation 7.5)}$$

$$u(v_{i,a-1}) \le u(v_{i,a}) \le u(v_{i,a+1}) \quad \text{such that } (v_{i,a} \in \text{ variants which realise}(e_i))\forall(a, 1 \le a \le b-1) \qquad \textbf{(Equation 7.6)}$$

We are now prepared to describe selections which are more useful than simply being valid. As with the earlier descriptive approach we shall start with the best selection, in eqn.7.7. Here we define that a zero weight element will have a no-selection variant and all other elements will be realised by the highest utility variant, $v_{i,b}$.

$$\forall(e_i \in D) \text{ where } ((w(e_i) = 0), (v_{i,s} = v_{i,0})) \text{ otherwise } (v_{i,s} = v_{i,m}) \qquad \textbf{(Equation 7.7)}$$

7.2.2 Cost Based Selection Algorithm

To define cost based selection we shall start with the definition of a step in eqn.7.8, the cost of a step in eqn.7.9 and the order between steps arising from their cost and weight in eqn.7.10

$step_x$ defines the transition of $v_{i,s}$ such that $s = s - 1$ **(Equation 7.8)**

$$cost(step_x) = w(e_i) \times (u(v_{i,b}) - u(v_{i,a}))$$ **(Equation 7.9)**

$$\begin{aligned} &cost(step_{x-1}) \le cost(step_x) \le cost(step_{x+1}) \\ &\qquad \text{where } (cost(step_{x-1}) = cost(step_x)) \\ &\qquad w(step_{x-1}) < w(step_x) \end{aligned}$$ **(Equation 7.10)**

Given this ordering of steps we can define that they are taken in order, as in eqn.7.11. Here we take steps (whether then make a progression towards any goal or not) in increasing order of cost, until all goals are satisfied. Further to this we can define the improvement process, as in eqn.7.12. Here we define that if a step can be reversed without breaking our parameters; that all costlier steps are not taken or reversed or cannot be reversed; and the variant to be selected in that step is one better than the variant currently selected for the variant then the step can be reversed. We shall not describe the grouped algorithm implemented.

$$\begin{aligned} &\forall(step_x, step_y), x < y \\ &\text{while } (\text{resource use}_p \le goal_p, \forall \text{parameters}, p) \\ &\text{take}(step_y) \leftrightarrow \text{taken}(step_x) \end{aligned}$$ **(Equation 7.11)**

$$\begin{aligned} &\forall(step_x, step_y), x < y \\ &\text{reverse}(step_y) \leftrightarrow \begin{bmatrix} \text{resource use}_p(step_y) \le spare(goal_p), \forall \text{parameters}, p \\ \text{and not taken, reversed or unable to reverse}(step_x) \\ \text{and where } v_{i,a} \text{ taken in } step_y, (v_{i,a-1} = v_{i,s}) \end{bmatrix} \end{aligned}$$ **(Equation 7.12)**

7.2.3 Sequential Selection Algorithm

The sequential selection algorithm works in a similar manner, with a different ordering of steps, as in eqn.7.13.

$$\begin{aligned} &pos(step_{x-1}) \le pos(step_x) \\ &\qquad \text{where } (w(step_{x-1}) < w(step_x)) \\ &\qquad \text{or } w(step_{x-1}) = w(step_x) \wedge u(step_{x-1}) < u(step_x) \end{aligned}$$ **(Equation 7.13)**

7.2.4 A Note on the Selection Process

The algorithms described offer a simplified solution to a case of the *knapsack* problem. The problem definition is slightly more open than usual:

- The knapsack has multiple dimensions (resources).
- Each item may be realised by a number of alternatives. Each of these may have a different size in each dimension.
- The selection of items may affect the size of the knapsack in any combination of dimensions.
- Where one of the dimensions is deadline, the longer spent picking the selection the smaller the knapsack gets in the time dimension.

In our case the simplification also offers a selection with greater consistency in the presented data than algorithms which seek to maximise resource use.

Termination of the selection is guaranteed by the *no selection variant* introduced by the system for each element. All selection paths proceed towards a selection in which no variant is realised. The no selection variant has no resource use in any resource and so all goals will be met. Earlier termination may occur if a deadline goal meets the deadline time during selection, in which case the algorithm also terminates.

However, no drop in resource use is guaranteed at any step in the selection path, except those where an element ceases to be selected. In many cases one can imagine (byte size, cost) perceived quality and resource use will tend to change together for much of the time. However, this is a property of data and user perception and not the specifications or algorithm per-se. Resource use is guaranteed to remain under the goals stated at each pass of the selection algorithm, including the improvement phase. Multiple passes may be required where goals change, or resources are consumed during the selection process. Goals may change as a result of a change in the selection, e.g. the implemented system allows for goals to be a function of the maximum weight realised. Resources can be consumed during the selection process by the effect of outside consumption, e.g. another application paying for data and so reducing the money available; or in the case of time by the execution time of the selection algorithm.

The complexity of the selection algorithm impacts the selection where a deadline is one of the dimensions. Time overhead in searching for an optimal selection (optimal being defined from weights and utilities) must be traded-off with the quality of the selection. By defining an order over the elements and their variants we avoid the worst of the complexity the problem poses. Our approach has four phases, the overall worst-case complexity being of the order of 3n + complexity of the sorting algorithm.

- Rate the weight of elements, utility of variants and define goals.
- Sort the variants of each element and the elements in the document into a degradation path according to the appropriate scheme.
- Pass over the path performing degradations until the goals are met.
- Reverse over the degraded part of the path making improvements where possible.

As each element's variant sequence is terminated with a no selection variant, the algorithm will always terminate with finite variants. In practice, the number of variants offered per element is unlikely to be more than a few tens. In the map application as developed there would typically be between two and five variants for an element.

7.3 RELATIONSHIP WITH EXISTING WORK

The approach described supports a more general mechanism for selecting "important" data than the layered approach often adopted in media scaling, e.g. MPEG and Ye, Jacobsen and Katz [126]. By addressing structured data where different semantic data or levels of representation are explicitly separated out, the improvement / degradation path can offer rather more choice than a sequence of layers. In the case of maps, it is also possible that a layer definition would not suit all users of the map. As in Noble [94] our degradation is performed with an understanding of the application semantics and the use of the data within the application. The processing of structured data can be modified to reflect application interaction and the design of our application specific and general components to support degradation is similar to that described for the Odyssey system.

There is work on content selection being undertaken in various standards [59,73,99] from the IETF and W3C. These approaches generally describe the hardware and software capabilities of devices and limited preferences of users. The user preferences catered for are primarily binary switches or exact matches, e.g. for selecting language. The suggested interactions for applications such as web browsing are that the client sends a minimal specification at the start of a session (with a given server). These techniques suffer from a potential scalability problem, in that the server must hold a description of all recent clients and perform selection, in addition to delivering data. These protocols also seem limited with respect to allowing a rich description of user preference in addition to device limitations, or describing perceived quality over attributes which may take a wide range of values, although it provides a clear means to express limitations, particularly with respect to devices.

CMIF [25] and SMIL [62] describe temporal and spatial behaviour of a presentation and have similar constructs for describing multiple variants of media elements in a structured presentation. The selection may be subject to a test on various parameters of the system in SMIL. CMIF focuses on providing alternative content for various contexts or user abilities, as defined by the author. In both cases the result is that the selection of alternatives is limited by the authored selection support. Our approach takes similar approaches to whole document degradation and supporting a range of alternatives for elements. We believe that our method offers a more flexible approach for emerging system and user classes.

The process of selection over many media elements, possibly which variants of each, to meet resource use is similar to the process described by Mohan et. al. in [88]. In both cases there is a measure of the utility (or value) of the variant being selected which should be maximised, the total resource use of all elements in all resources is considered and any element may be realised by just one variant. The solution in [88] simply maximises the sum of the utilities, while we take into account the weight of the element as well as the utility of the variant. When building our degradation paths the lower value elements are presented as candidates for degradation first. This avoids the problem of many low value elements appearing more important to the selection process. This problem is exacerbated in their work as the utility is always one for an original variant of an element, regardless of its value to the user.

Work has been performed which illustrates the applicability of adaptation in web type applications, such as Abdelzaher and Bhatti’s in [2]. In this case the adaptation was at a web server and was due to the server load. Their results demonstrated that there was sufficient potential for adaptation to allow a substantial limit on the latency of page delivery under a wide range of server loads (factor of 6 request rate).

Koistinen and Seetharaman [74] describe a process of negotiating QoS between a client and server, over the “worth” of various characteristics of the service to be provided. They describe restrictions over parameters, particularly those concerned with reliability and delay. Within those restrictions they present a negotiation protocol, which is based around the assignment of weights to parameter values: they illustrate this in a similar manner to the utility functions we use. Their approach uses a language based specification of contract requirements and is intended for negotiating deterministic QoS for ongoing service provision.

7.4 SUMMARY

In this chapter we have completed the description of our approach to contextual mediation. In earlier chapters we discussed data, how it may be adapted and its description with meta data. We described context, both in states and consumable resources and how it may impact the perception of data. In the last chapter we discussed how contextual states can be used to control the selection of specifications for mediation which reflect those contextual states. The specifications describe preferences amongst data and limits over resource use. In this chapter we have described a process for moving from offered data to selected data, while reflecting the preferences and resource limits in the specifications.

Our process for performing the mediation takes the weights and utility functions and uses these to form a degradation path for the various elements in the document to be presented. The degradation path is a view on degradation perceived by the user. Data of no interest is eliminated, which is a base-line improvement compared to no mediation or uniform degradation of all data. Data of most importance is degraded last. The selection is passed through admission control to reflect the various goals against each managed resource. This process is repeated where the selection affects the goals or the resources available to the request have changed during the selection process. The level of differentiation between elements and the detail encoded in the specifications in this approach offers greater flexibility and user sensitivity than many commonly used approached.

We have offered two algorithms, which interpret the specifications in slightly different ways. In both cases the definition of a best selection is consistent and the meaning of the weights and utilities arrived at is similar. Whether either of these is an ideal algorithm is not clear and in tests little distinction is perceived. However it is reassuring that our system of specification and the descriptions of data and context are sufficiently distinct from the process of selection that flexibility in this is possible.

CHAPTER 8 CASE STUDY - MAP ADAPTATION

In this chapter we will review the map application used as an example throughout this thesis as a whole application, bringing the various components of the mediation together with illustrations from the developed prototype.

8.1 WALK THROUGH MAP LOAD

We shall now illustrate the application of our techniques in loading a map segment. The interaction is similar to that described in figure 3-3. The full range of data and structuring is necessarily reduced to aid comprehension. We shall consider four types of feature: roads, driving restrictions (one-way systems, turning prohibitions etc.), traffic conditions and contours. Our user assigns “roads” a type weight of 1, driving restrictions and traffic conditions a type weight of 0.8 and contours a type weight of 0. Utility functions over detail indicate a preference for map information detail at between 1:50000 and 1:250000 scale. For driving restrictions this function is overridden to indicate a preference for detail between 1:2500 and 1:25000, as small restrictions still affect the route taken. The utility function for restrictions contrasts to the function applied to traffic conditions, where a small queue is not so significant and so fine detail is less necessary. Outside the stated preferences the functions’ value falls off gradually.

The location service, taking information from a GPS receiver and the route plan, indicates to the application an area of map which is expected to contain the driver’s location in two minutes. The application confirms that it does not already hold a recent copy of the requested data and passes the request to its QoS manager. The QoS manager takes the request and requests the meta data for that map segment from the map server, a second request to the local traffic service and a third to the local council for up-to-date traffic restrictions, including planned road works.

The three responses are processed as the data arrives, to build the element and variant trees. The selection algorithm then walks over the tree calculating the utility of each variant. For each element the variant with the highest utility is marked as selected. Contour data are all marked with a utility of 0 and so are not selected. Road information is offered at 1:10000, 1:50000 and 1:100000 for the area. 1:50000 returns the highest utility and so is chosen. Driving restrictions are offered at 1:10000 and 1:50000, so 1:10000 is chosen. Traffic information is only offered in 1:10000, so is selected.

The total resource requirements for these three data elements is 100kB. If five seconds have passed since the request came in from the application and the map is needed a minute before arrival in the area of interest, then the selection should be made to download by 55 seconds from the current time (given the available network throughput). If the network connection available is only 1kB/s, the resource manager then returns a shortfall of 45kB to the QoS manager. A revision of the selection is then initiated.

Two degradation paths are built: one for road information, with a type weight of 1; and one for restrictions and traffic data, with a type weight of 0.8. The less important type is degraded first. The driving restriction information is degraded to 1:50000 detail. The traffic data remains at 1:10000 as no other alternative is available. The shortfall is now 30kB. The next stage of degradation removes both elements and so is passed over at this stage. The road data are treated next and degraded to 1:100000. The data size is now 20kB. under the available resources. The goal has then been reached.

The 0.8 type weight information is tested to see if improvement is possible without breaking the goal. If it is found that it is, then the driving restriction is returned to its original level. The resource requirement would then be 50kB. On reapplication to the resource manager the request is accepted. A reservation is put on the network by the resource manager and acceptance returned to the QoS manager. The QoS manager then returns to the application a list of the data variants to request. The application requests these from the appropriate sources and displays the data returned.

One can imagine a second user, who is a cyclist, defining different preferences: Traffic information is of less concern, as traffic jams are easier to pass. However, hills have a greater impact and so contour information would have a non-zero type weighting. A similar process would be followed to select the required data, but the resulting display would provide information relevant to that user.

8.2 SPECIFICATIONS TO REFLECT CONTEXT

In this section we shall describe, in part, the specifications we have developed in developing the map application and how they reflect different contexts. These specifications are used in chapter 9 for the tests performed. The specifications inform the results of the user study. The full specifications, in their XML encoding, are shown in appendix 2.3.

The first, basic, set of specifications cover base cases and system issues. These specifications come with no matching functions and so are always valid. They include weights for undefined type elements and structural elements and a general utility function over age, as shown below:

```
W:     select-wf(<null>, system:*) = 0
       select-wf(<null>, system:structural) = 0.000001
       select-wf(<null>, map:structural) = 0.0001
U:     select-uf(age, map, *) =
          age = 0 → 1
          0 < age <= 525600 → 0.7 + (0.3/525600) * (525600-age)
          525600 < age <= 5256000 → 0.4 + (0.3/4730400) * (5256000-age)
          5256000 < age → 0
```

The utility function over age takes minutes as the unit of its parameter. The function returns 1 for no minutes old, 0.7 for 1 year old, 0.4 for 10 years old and 0 for older than 10 years. Between these ages the values change linearly with age.

The next group of specifications we consider are those which vary with task. Task can take a values of work or recreation, recreation specialises to tourist, shopping and hiking. We concentrate on tourist and work in our tests. The tasks result in preferences for a range of types and other limitations. We present a comparison of work and tourist (including definitions for recreation) in table 8-1. This table does not show all functions defined, but illustrates the choices encoded. Preferences for churches are not defined for work. This indicates that an opinion on churches is not needed. Navigation specifications may give them some weight as they can act as useful navigation aids – with a weight arising either as a generic building or as a special class.

Function	Work	Tourist
select-wf(<null>, tourist-feature)	0	1
select-wf(<null>, building:airport)	0.7	0.7
select-wf(<null>, building:church)	<undefined>	0.75
select-wf(<null>, building:phone)	0.5	0.7
goal(deadline)	45	45
goal(price)	500	400

Table 8-1: Comparison of Specifications for Work and Tourist

The next group of specifications reflect the mode of transport. In a car, walking and cycling were considered, although we concentrated on car and walking. A selection of the preferences encoded are presented in table 8-2. Again, the context gives rise to levels of interest in different types of feature. The required detail for different classes of element are also defined. Those on foot are assumed to be more interested in fine detail, especially on roads and paths which they may be following. Car users are also defined as having some interest, generally, in railway stations. Not as features to use (although driving to a station is a possible task and would have weights reflecting different needs), but as navigational handles – features which are easily identified on the ground and so used in navigation.

Function	In-car	Walking
select-wf(<null>, building)	0.73	0.75
select-wf(<null>, building:railway-station)	0.001	0.85
select-wf(<null>, road)	1	0.8
select-wf(<null>, road:byway)	0.15	1
select-uf(scale, map, *)	0 → 0.5 10000 → 0.5 50000 → 1 100000 → 1 1000000 → 0.8	0 → 0.5 1000 → 0.7 25000 → 1 200000 → 1 1000000 → 0.6
select-uf(scale, map, road)	0 → 0.6 25000 → 0.8 50000 → 1 100000 → 1 1000000 → 0.6	0 → 0.6 5000 → 1 50000 → 1 100000 → 0.7 1000000 → 0.5
select-uf(scale, map, path)	<undefined>	0 → 0.6 5000 → 1 25000 → 1 50000 → 0.8 100000 → 0.5 1000000 → 0.3

Table 8-2: Comparison of Specifications for In-Car and Walking

The final set of specifications relate to speed (given in miles per hour here). These describe some basic speed related specifications and also some speed plus mode of transport combinations. We show some of the specifications for diving slowly and driving fast in table 8-3. Different preferences for classes of road and building numbering are shown. Details such as building numbers are generally not needed when driving fast. Weight functions for distance from the point of interest are also shown for a-roads and minor-roads. Interest in major roads falls off more slowly than for minor roads, as detailed navigation on small roads tends to be more appropriate for short distances and major roads are used for longer distance journeys. Distance functions are given with units of metres. Goals of deadline and screen density also become more limited as speed increases, reflecting urgency in loading before in the predicted location and increased limitations on screen visibility, even for a passenger performing the navigation.

Function	under 21mph	over 51mph
select-wf(<null>, road:minor-road)	1	0.6
select-wf(<null>, road:b-road)	1	0.95
select-wf(<null>, road:a-road)	1	1
select-wf(<null>, road:motorway)	0.9	1
select-wf(<null>, building:number-name)	0.8	0
select-wf(distance, road:a-road)	500 → 1 1000 → 0.91 2000 → 0.71 5000 → 0.2 10000 → 0	2000 → 1 5000 → 0.92 10000 → 0.82 20000 → 0.72 50000 → 0
select-wf(distance, road:minor-road)	300 → 1 1000 → 0.9 2000 → 0.7 5000 → 0	200 → 1 1000 → 0.8 2000 → 0.4 5000 → 0.05 10000 → 0
goal(deadline)	30	20
goal(screen density)	2	1.1

Table 8-3: Comparison of Specifications for In-Car Below 21mph and Over 51 m.p.h.

When combined these specifications offer what we believe to be a good generic set of preferences for simple navigation contexts. Extensions to reflect a more diverse range of contexts and to support more diverse data types are left for future work and are subject to the availability of suitable data to define preferences over.

8.3 SELECTIONS ILLUSTRATED

From the application developed we now present a graphical illustration of selections which modify the information presented according to a deadline interacting with a network throughput and context. In figure 8-1 and figure 8-2 we illustrate four maps covering the same area. The centre of the map is considered to be the area of most interest. In figure 8-1 the basic selection is similar: a context indicating fairly slow driving has caused a preference for detailed road information nearby, while larger roads are shown over a greater distance. Some other features, such as woodland a river and a railway are included as navigational aids which are relevant in the context. The difference between the two maps is in the task context. The left-hand map is for use “at work”, the specification arising from this included a disinterest in most tourist attractions. The “tourist” task however results in a strong interest in these features, which are the blue symbols around the map. The treatment of urban areas (yellow / brown) changes between the two contexts. The rendering of these areas suffers from a problem in connecting the fill seed with the area outline in the rendering. As developing a map rendering engine was a side issue a large amount of time was not spent solving this.

In figure 8-2 we see the same area again, here both maps are for a car driver at work. In this case the speed of the car is changed. The faster car's map, on the right, shows less small roads and other features but more complete drawing of the distant portions of the major roads. This is in response to a change in the weights when driving fast such that minor roads become less important and the weight assigned to roads falls off more slowly. When driving slowly, nearby minor roads are considered important and some other features which may aid detailed navigation gain a higher weight.

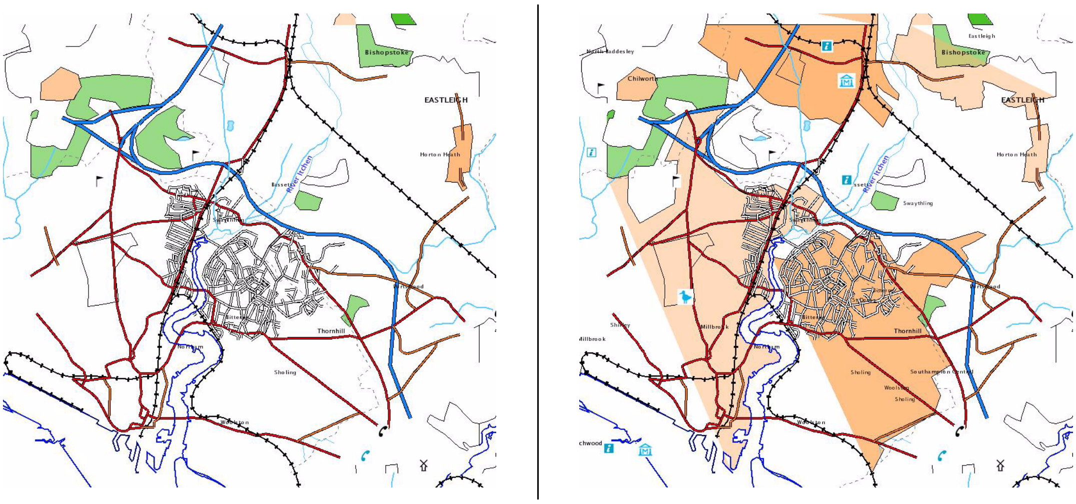


Figure 8-1: Example Maps Subject to Change of Task
Maps of same area, deadlines, network, mode of transport ("in car") and "speed (30mph), left hand task "work", right hand task "tourist"

Figure 8-2: Example Maps Subject to Change of Speed
Maps of same area, deadlines, network, task ("work") and mode of transport ("in car"), left hand speed = 40mph, right hand speed = 70mph

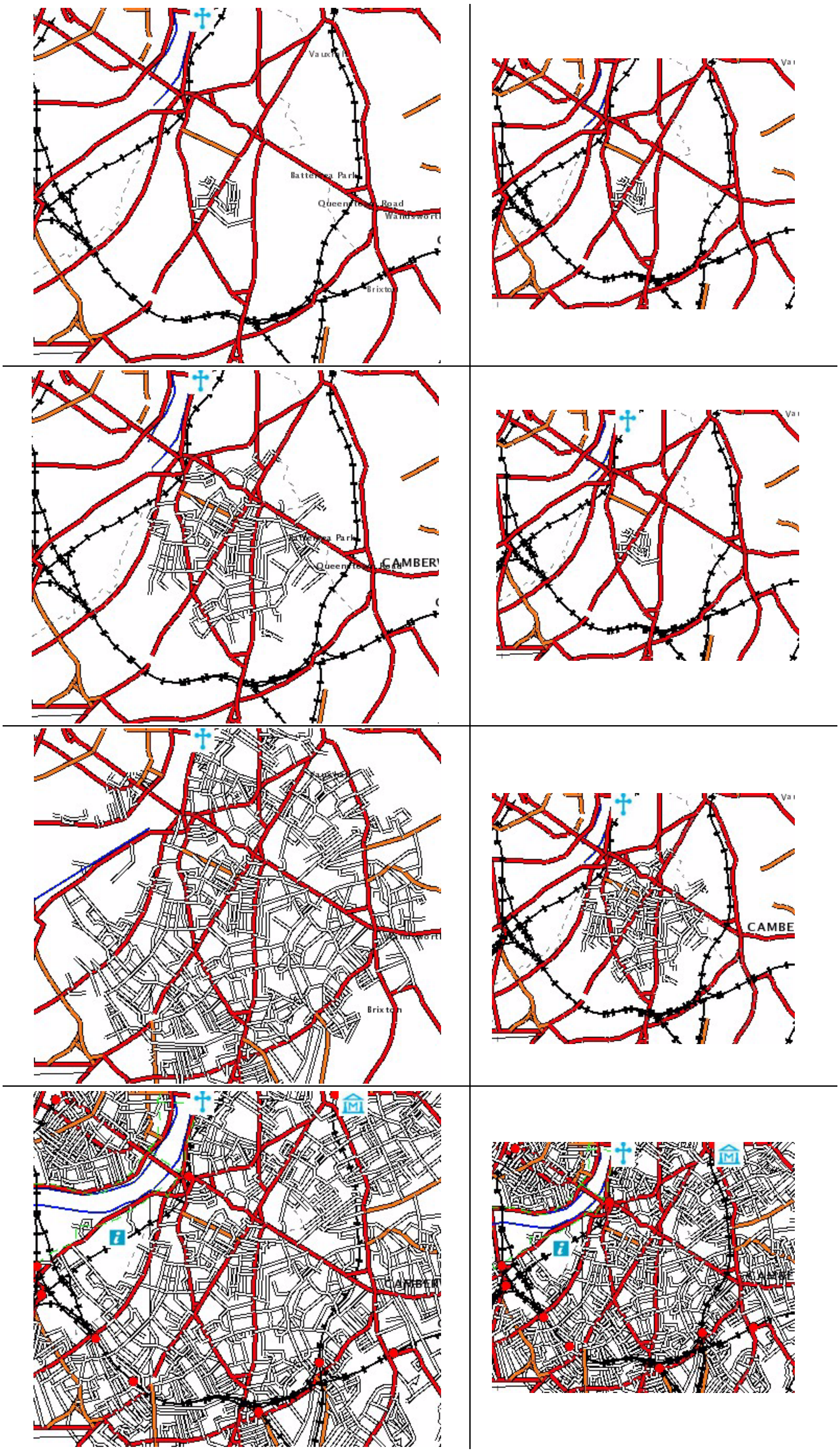


Figure 8-3: Urban Maps Selected to Meet Screen Densities

Densities: row 1 = 1.0, row 2 = 1.4, row 3 = 2.0, row 4 no-mediation

In figure 8-3 we consider a different location, a 4*4km area of London, which shows a high density of roads. The two columns illustrate screens of different sizes (400*400 and 300*300 pixels), while the rows illustrate treatment of different drawing densities on these screens. The last row illustrates an unmediated selection. The context given was for a tourist driving slowly. All maps have are basically acceptable for navigation, although the unmediated maps require much more attention to extract the useful information from the mass of detail. The lower density maps contain very little detail besides the major roads and railways and could be expected to remain usable under difficult viewing conditions. The medium density maps contain rather more detail in the larger screen size, reflecting the increased ability to render detail clearly. These maps would be usable under many conditions.

The higher density maps contain substantially more information in both sizes. Under difficult conditions the larger map would be hard to use, under good conditions however it provides a detailed rendering which is suitable for more detailed navigation. Note that the railways are no longer rendered. They took a lower weight than the minor roads (and so their loss is deemed acceptable, if visually significant – if railways are found to be more important their weight should be adjusted). However, in the more restrictive drawings the drawing density of minor roads would not have been acceptable. The railways would have been selected at the improvement stage in the selection process. With the higher density allowed, their removal would have been sufficient to pass the screen density admission control while the minor roads remained selected. The smaller map now shows a level of detail similar to that which the larger map showed under the medium screen density limit. Being shown in a smaller area the density of data will be greater for a given selection as the features are drawn in the same way on both maps. As differences in screen size become more pronounced the application may also consider modifying the rendering used.

In the last row of maps we see all the available data rendered. In the larger map there is extra data away from the centre and some extra features which had been eliminated by the mediation. However, in general the result is similar. For the smaller screen the unmediated map is noticeably denser and features have started to become confused or overlaid. A different rendering may help, although the roads are already drawn too thinly to include road names. The map data used is also quite basic. Road names, contours, one way systems, house numbers, shop logos and other potentially interesting data were not available to us. All this data exists in location-correlated form for drawing on maps.

We can see that our basic data set is capable of providing maps which are cluttered to the point of being hard to read. The potential benefit of mediation only really begins here – as long as some data can be eliminated without making the map unusable, diversity of data need not be limited by screen size.

8.4 RELATED WORK

Map adaptation for mobile systems is considered by Abdelsalam in [1]. They also take the approach that semantically different data may be selected according to the benefit to the user and that vector map data may be presented with different levels of detail. Their selection algorithm with respect to inclusion of types is binary and does not allow for improvement of lower weighted layers once resource needs have been met. This simple sequential selection may become inappropriate if the load imposed by some layers is substantially greater than others. While this may not be an immediate problem for map data, it is limiting in the range of application and data domains the technique can be applied to. Both these techniques allow for data size and screen clutter to be treated. Their approach is based on GIS work, the semantic division of map features being taken from the feature layers in GIS. Our work is similar in this respect, although we also consider location to be a semantic attribute and we allow for a finer grained treatment of the types. The resource limits considered in our system are also defined in substantially more detail and applied in a manner which should give greater control. Their approach seeks to limit download size, but detailed treatment of network characteristics is not given. We provide an explicit deadline to meet, while they seek to reduce size (in some cases in quite an extreme manner). Our treatment of screen density is an approach they leave as future work. Their response to screen size is to eliminate features which are small in comparison to the screen size. The elimination of detail in their system is performed by a filter which is either applied or not. As they do not aim to meet resource goals the trade-off between adaptations we provide is not required. While our system may provide data which is noticeably degraded, their system will provide good-enough data while ignoring other needs such as deadlines. The preferences for different features are hard-coded in their system, while we derive them dynamically from a context multi-aspect view of the context. The ability to modify preferences (weight and utility functions and goals in our system, weights and resource-use limiting metrics in theirs) to reflect the needs and situation of the user is vital in properly addressing the needs of a widely varying community of users using a diverse set of devices.

The use of the location in determining the area of interest allows the application to constrain its meta data request to the server. When rating the elements in order to build a degradation tree, the location can again be applied to assist in differentiating between elements. In our case the weight may vary with distance from the area of interest. A similar approach may be applied in virtual- or augmented-reality, where decisions regarding feature inclusion or rendering quality can depend on the user's direction of view and location relative to the feature, as seen in work at Nokia [79].

Initial steps in providing maps for tourists tailored to device, task, cultural influences etc. is described by Zipf in [129]. Much of the focus of their work is on the rendering of maps, which would operate in tandem with the selection of data to be rendered. In the same way as one would wish to have sparse features for clarity on a handheld device, enlarged rendering would also be desirable (indeed this would reinforce the limitation on data presented due to screen density).

Rather than presenting a to-scale map, when describing the whole of a route to be navigated, a schematic map may be more useful. This eliminates all unnecessary features and ensures the roads to be used are all presented at a size which can be read. The whole route can be presented at one time, with smaller roads still visible. An example of this approach to improving map usability for navigation can be found in [5]. The problem we are addressing is similar, but we are also interested in cases where the route is not known, or there is no route to describe. The highly simplified route maps may note roads crossed or not to be taken at junctions by examining intersections. To some users, greater detail of the area travelled through is desirable, particularly if that extra detail is relevant to them. A map which contains no information to aid recovery may also be a problem if directions are not followed precisely, or unforeseen obstacles are encountered. Where the route to be followed is known, a large scale map which shows the whole of the route (and little else) can be provided by our system. The ability to zoom in and see more detail around the path is still present and useful.

The different needs of users engaged in different navigational tasks is well provided for by our system. A discussion of the differing needs of car drivers and pedestrians is given by Corona and Winter in [37]. They find that the needs of the two groups are substantially different. In particular that there are many levels of distinctions between features which a pedestrian uses and a driver typically does not, as well as classes of feature which a pedestrian is interested in and a car driver is not. The topographical detail pedestrians use was found to be greater than car drivers, who typically view roads as lines rather than areas on the ground. They also found that the range of maps scales applicable to each group is quite different, although overlapping. These needs can be represented in our system as differing utility functions for levels of detail in feature rendering and classification and differentiation between feature classes. The findings also have an impact on map drawing, pedestrians being receptive to different levels of detail, which is not a direct result of our work.

8.5 SUMMARY

In this chapter we have reviewed our selection process, collecting the key points of the previous chapters into a single example, drawing from our map mediation example application. We have then illustrated the results of this process with images from the application developed. While these do not constitute a proof of the approach's applicability, we hope they illustrate to the reader that contextual mediation can usefully reflect a diverse range of contexts. These examples can be read in conjunction with those in figure 4-3 in chapter 4, which illustrates the scalability of the approach to a range of network conditions.

Chapter 9 Map Application - Evaluation

9.1 Review of Aspects to Evaluate

To test our ideas an application was developed which implemented the contextual mediation described for map data. The specification / mediation mechanism, a model of network throughput and round-trip-time and a model of screen density were all implemented. The application provides mediation of vector map data and is coupled with a simple map- and meta-data server. An automatic sensing of context was not implemented so context states are described by manual selection. Network simulation can be configured for testing purposes. The network simulation and locations to be requested can be controlled from files describing a timed series of cases. Further engineering details of the application (software architecture and file formats) are provided in the appendices.

In order to examine our solution's effectiveness we subjected it to a series of tests. These were designed to measure various aspects of the response to context which we had addressed in the test-bed application. The effectiveness of the application was measured through:

a. Ability to meet deadlines. This includes the ability of the network monitor to model link characteristics and to respond change in those characteristics.

b. Scalability across a range of networks may be examined by applying a wide range of link characteristics and comparing the managed system with the unmanaged; and also examining the quality of the data produced.

c. Overhead of adaptation system. This requires an examination of the proportion of the deadline time spent loading meta data and performing selection.

d. A verification that the selections made are reasonable to actual users. This test was separated from the usability of the application, so as to examine the selections made, rather than other aspects of the application.

Quality of specifications was not tested, as the application's HCI would have impeded the level of user testing required to gain an effective view of this. However the user tests for the selected maps were presented in the context of a user navigating for a car.

9.2 TEST ENVIRONMENT

We chose to simulate a series of network environments, rather than take results from actual network tests. While removing some sense of realism, it provides clearer results without a large scale deployment. The range of simulations also allowed us to separate our tests from available technology, so results are not tied to certain real-life cases. The tests were sufficiently diverse that behaviour in real cases may be predicted.

Our simulation environment used a group of 350-500MHz Pentium based systems with 128MB RAM, running Linux (SuSE with 2.4.4 kernel) and X v4.1 as our client platform. While these are more powerful than many portable or embedded devices they illustrate that high-end computing power is not required for the application. A single 800MHz Pentium server with 1GB RAM running Linux (RedHat with a 2.2.16 kernel) was our server platform. The server ran Apache (v1.3.14) as a general web server, which for the map server passed requests onto the map servlet, run under Jakarta Tomcat (v3.2.1). Sun’s 1.3 JDK was used for implementing both the client and the server. The systems were connected to the departmental network with 100Base-T Ethernet. There will have been some background effects from other traffic, but on a scale which would not invalidate the results (although it may be visible in a few anomalies). The client and server system loads from other applications were sufficiently light-weight that no significant interference occurred, although some low level variation is to be expected (as on a real deployment). Further details of the map servlet can be found in appendix 3.

Poor quality networks were then simulated over this, by creating both an initial delay and returning the data slowly. We wished to avoid the complexity of detailed network simulators such as ns and did not require a simulation of a specific network to demonstrate the applicability of our approach, so chose to implement a simple simulator as a component in our application. Both the network simulation and client application were run on the same system, by embedding the simulation within the instrumented network streams. This also shielded the results from some of the variation due to other use of the network connections. The client-based approach is appropriate for simulating networks which are limited by a dedicated media last-hop. For other cases the simulation would have to approximate the overall effect of other traffic. The simulated network was assumed to be using slow start / congestion control type variable transmission windows (although not explicitly modelling TCP). See [38] for an in-network approach to simulating wireless links for UDP and a discussion of other approaches to this by Davies et. al. The simulation was intended to be appropriate to our monitoring of whole, quite large (10kB+) requests. A detailed simulation of network behaviour was not required. The network simulation process is described below. The data rates are given in bytes per second here, as there is no explicit serial stream of data or model of the protocol overheads usually encountered.

The simulation formed a pipeline between the instrumented stream reader and the server. A request would first have a “round-trip-time” delay applied. This was found from a fixed time plus the time to transmit the GET URL (the number of inter-packet delays at 128 bytes per packet required for the URL string). This is a substantial limiting factor in many wireless networks, especially those based on cellular telephone networks. Once the initial delay was satisfied the HTTP request to the server was actually opened and an object handling the returning data added to a “live requests” list. The opening of the HTTP connection and initial servlet processing made the initial delay slightly longer than the setting given.

A single thread in the client configuration ran a cycle of “delay, send data”. The delay applied was designed to simulate the required bandwidth for 512 byte data “packets” passed to the application at each pass of the loop. The delays were defined to millisecond resolution using Java threads. Each request in the live requests list was called to send data in turn, one per pass of the timing loop. The network simulation therefore gave a single throughput across concurrent requests. The playout of data used a variable window, to simulate some of the effects of TCP. The initial window size was 128 bytes and was increased by doubling at each send for a request to a limit of 512 bytes. Data was returned to the application in “packets” of up-to the packet size, or whatever data remained at the end of a request. The data sent was the next section of the stream selected so that data remained in-order. The inter-packet delay was modified according to any delay in the sending process to maintain an overall constant inter-packet delay (as Java’s thread timing only guarantees minimum times and interaction with the application meant that some delays were substantially greater). As a network simulation this technique is clearly simplistic, however it was sufficient for the purposes of testing our system’s ability to select data in situations with reduced throughput or long round trip times. At higher bandwidths from around (50kB/s) the inter-packet delay becomes small and the playout starts to become subject to variations in the real network between the systems.

A model of loss was also implemented, to test our system’s ability to perform in poor quality wireless networks. As the inter-packet delay was calculated at each pass the possibility of a random loss was calculated. If the loss was found to have occurred then the inter-packet delay was doubled and the window size halved (to a minimum of 128 bytes). Once a loss was being simulated subsequent loss was also considered. This could be set to a different probability (following the Gilbert model), to simulate the bursty loss which is a feature of wireless networks. The same addition of the delay and reduction of the window size was made in this case too. The window size doubling was applied to subsequent loss-less packets. While not an exact simulation of TCP the effect is representative of adaptive protocols running over a wireless network.

The network simulation was designed to accept a file of network condition descriptions, to be applied at given intervals. This enabled automatic testing of a series of network conditions and the response of the system to change in network conditions. Similarly the location service was designed to take a file of location descriptions as a series of locations to be notified to the application, at given intervals. Locations were defined in this file as 3km * 3km squares of urban and suburban locations (areas then expanded to $10.7km^2$ to match aspect ratio of window dimensions). This enables automatic testing of behaviour using a variety of feature densities and on a scale in which the effects of individual requests and the map data offered would not dominate the results. In general, each network condition tested used a series of twenty or thirty map tiles. Where the step between network conditions was not intended to be part of the test the first three loads with the new conditions were not included in the results presented. As we shall see this is sufficient to allow the network model to adjust.

9.3 Network Monitor – Scalability and Responsiveness

The first group of tests examine the ability of our network monitor to respond to change and the quality of the predictions of throughput and round-trip-time.

9.3.1 Throughput

In figure 9-1 we show a trace of expected throughput and measured throughput, as the simulated bandwidth changes. The simulated bandwidth ranged from 24kB/s to 3.0kB/s with a round trip time of 100ms and no loss. The steps correspond to factors of 1.25, 1.5, 2 and 4. The points on the graph correspond to individual loads. All loads were made with a deadline of 30 seconds.

The achieved throughput for meta and map data and the simulated bandwidth vary significantly from each other. This is an effect of the compression, as the data volume is measured after decompression but the network simulation applied with compression. This reveals that the meta data experiences a much greater degree of compression than the map data. This is not a great surprise being formed of rather less dense and more repetitive XML compared to the dense ASCII of NTF. The minor fluctuations between loads can be attributed to differing characteristics of the map areas requested, the data selected and fluctuation in load on the network connection between the test system and the map server, the two systems themselves.

The absolute value of the model is not the value which requires critical evaluation here, rather it is the relationship between the expected throughput (from the model) and the experienced throughput, in particular for the map data. What we find is a stable model in the presence of minor fluctuations, which nevertheless reacts quickly (as expected) to larger changes. A typical change will have three substantially incorrect (but improving) expected throughputs before the prediction and measurement converge again. This is constant across the range of jumps – the model is as rapid in adjusting to a change of 12kB/s to 3kB/s (4*) as it is in a change from 3kB/s to 4.5kB/s (1.5*). The only noticeable departure from this case is when moving from 4.5kB/s to 5.6kB/s where there is a substantial lag. This occurs as the jump is not triggered and the error seen is acceptable for our purposes. Note also the stability of the expected throughput under minor fluctuations, particularly noticeable in the meta data trace.

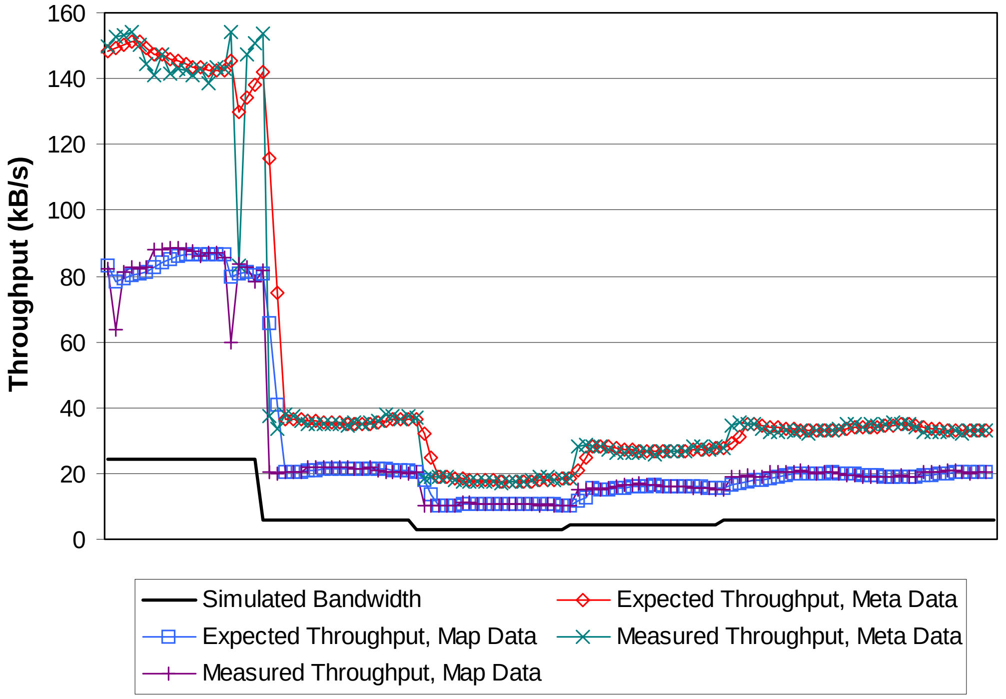


Figure 9-1: Response to Change –
Expected and Experienced Throughput Trace

The response to change in throughput is presented again in figure 9-2. Here we highlight the error between the expected throughput and that experienced. Again we see the three steps clear from the zero-error line and few minor fluctuations.

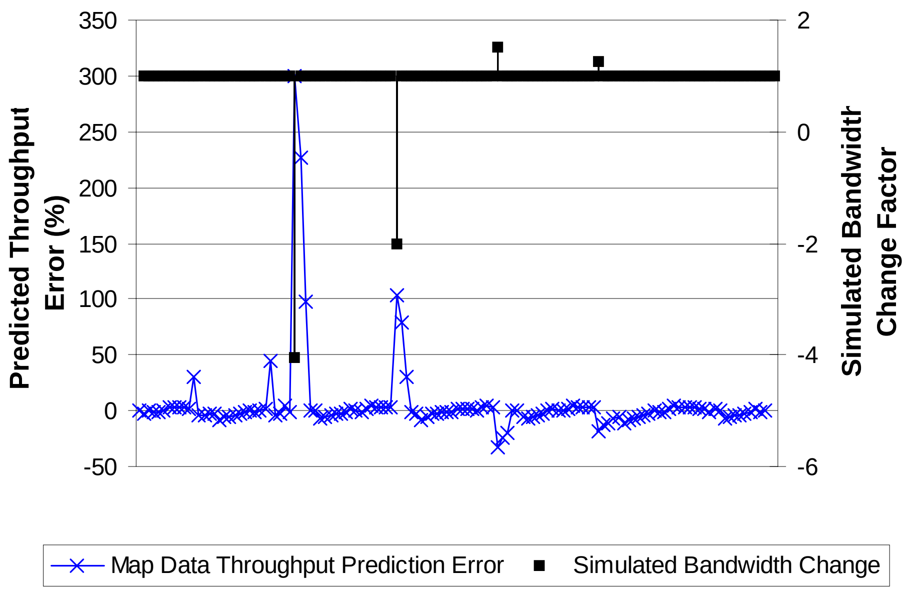


Figure 9-2: Response to Change – Throughput Prediction Error Trace

9.3.2 Throughput with Simulated Loss

We next tested our throughput model's performance with a lossy network simulation. We compare plots from test runs with and without loss. The loss rates are given as a probability and the probability of a subsequent loss, allowing a basic model of error bursts. The loss rate used was 5% (1st drop) and 20% (subsequent) – quite a severe network error rate. The test was still performed over a the local network on a server with some background load and so additional minor variations could also occur, as before.

In figure 9-3 we see traces of prediction error, with changes in the simulated conditions noted. In all cases there was the delay, as before, in reacting to a change. The trace for the lossy network also show prediction errors leaving the zero-error area which the loss-less trace occupies in the main. This suggests that the model may be slightly inaccurate where losses significantly affect an individual load. This is to be expected as losses occur randomly and cause deviation from the simulated bandwidth. However, in most cases there are no series of points with a substantial error away from the changes in simulated conditions. This suggests that the tend towards short-term stability in our network model correctly does not change its predictions on the basis of one load being different.

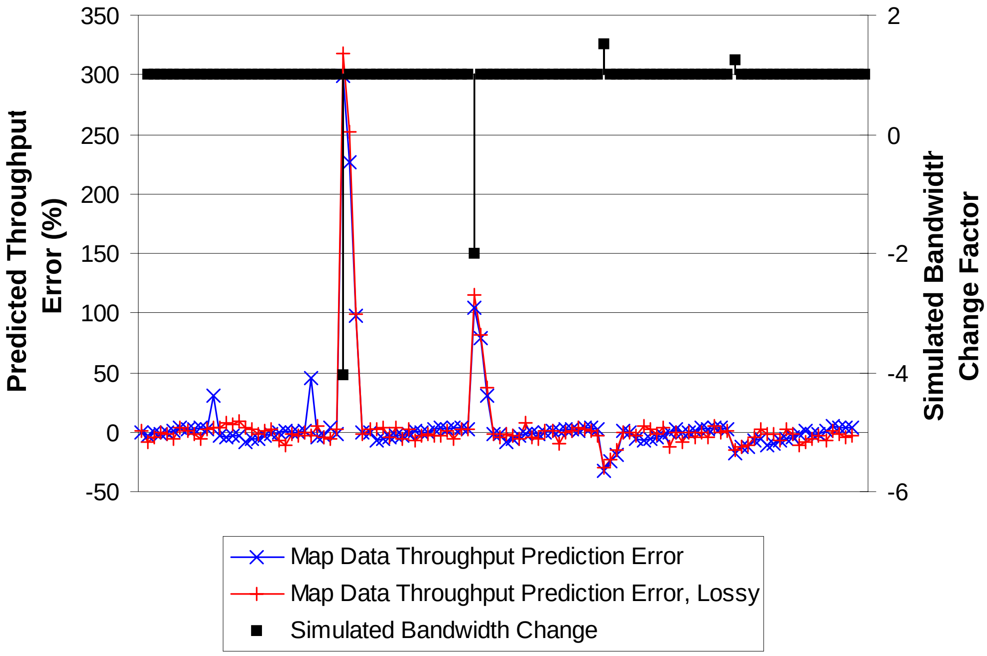


**Figure 9-3: Response to Change –
Throughput Prediction Error Trace with 5/20% Loss and with No Loss**

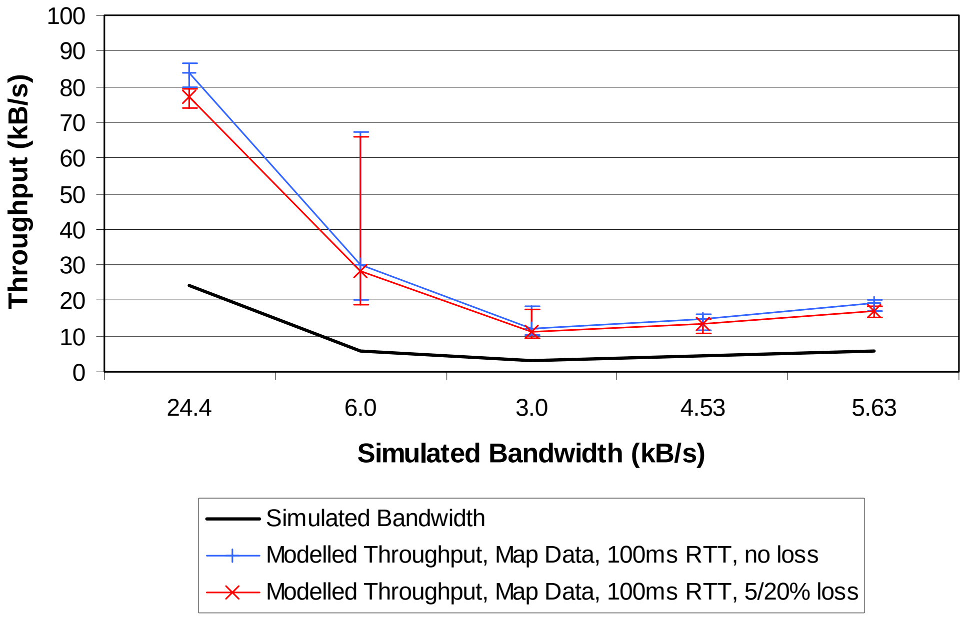


**Figure 9-4: Response to Change –
Average Throughput Under Varying Bandwidths with 5/20% Loss and No Loss**

Figure 9-4 plots mean, 10th and 90th percentile modelled throughputs. We see that the losses aggregated over the fairly large requests performed by the application caused an effective lowering of throughput. The error bars (10th and 90th percentiles) also become at least as wide, or slightly wider, as the error rate rises in most cases. However the lossy and loss-less cases perform similarly, indicating the stability of the model.

9.3.3 Throughput Compared with No-Jump Network Models

In evaluating our network model we also wished to confirm that we can offer a benefit over either a rapidly responding or stable simple moving average model. We ran the throughput reaction tests with throughput models which took simple moving averages rather than our model. We modelled systems which offered similar stability to ours, referred to as “slow reactions” and also similarly rapid response to ours, referred to as “fast reactions”. Our system is referred to as the “jump algorithm”.

In figure 9-5 we observe the errors in each model as the network changes simulated characteristics, with no loss. As might be expected the slow reactions model takes a long time to adjust to changes. The fast reactions algorithm on the other hand moves at a similar rate to our jump algorithm in responding to step changes.

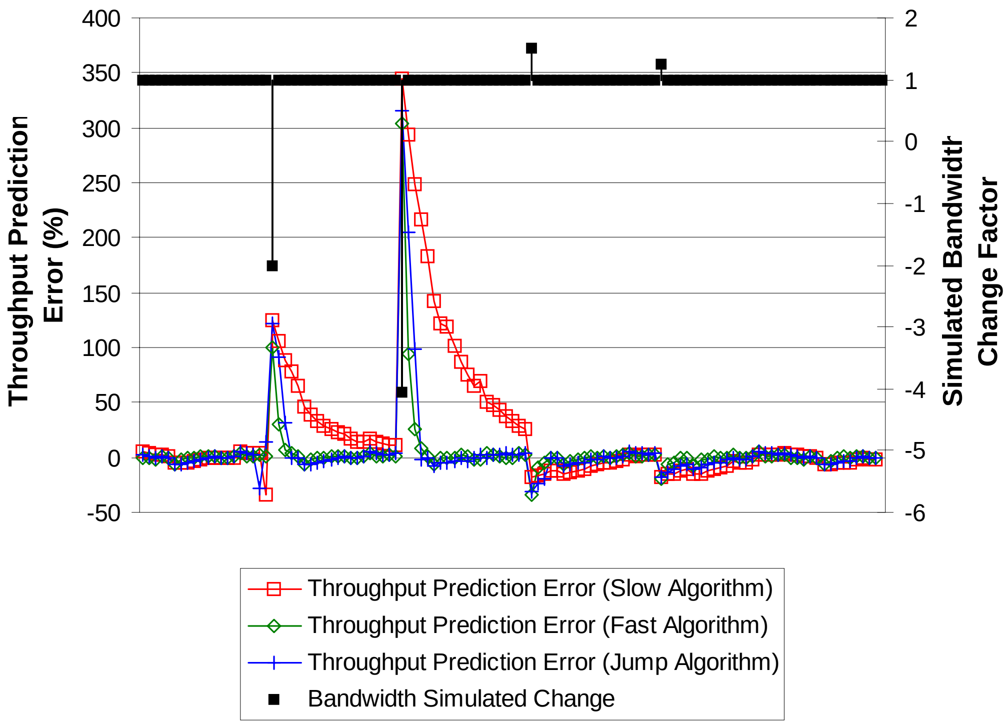


Figure 9-5: Response to Change – Throughput Prediction Error Trace with Different Models, No Loss

Figure 9-6 presents similar results, under the 5/20% loss model. The slow reactions model displays similar traces to the previous case, this is to be expected as the errors will not impact its measurements greatly. On the other hand the fast reactions algorithm substantially departs from the line of the jump algorithm and is slightly outside its trace for much of the test.

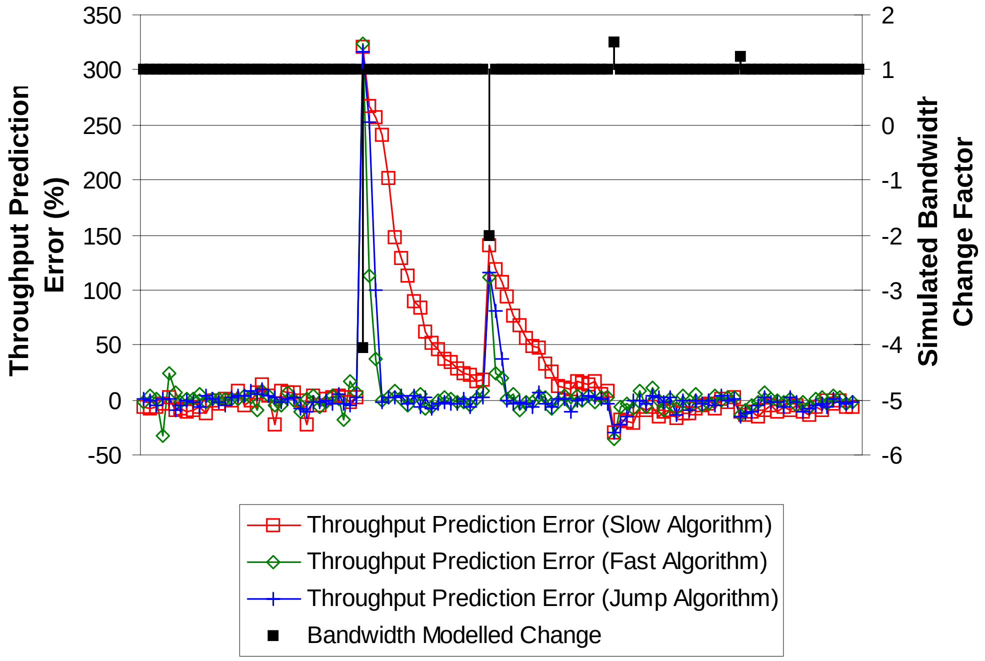


Figure 9-6: Response to Change –
Throughput Prediction Error Trace with Different Models, 5/20% Loss

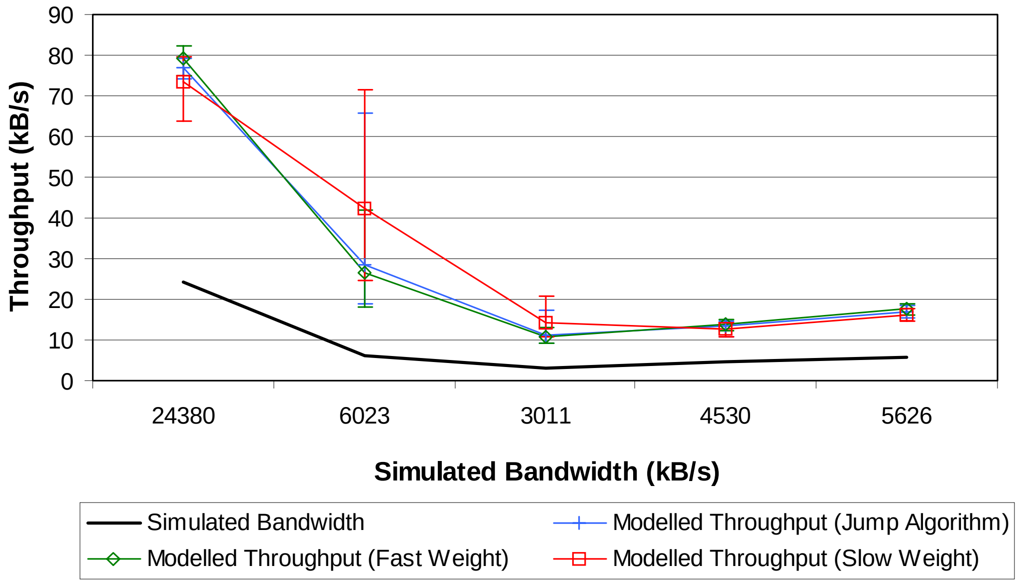


Figure 9-7: Response to Change –
Comparing Monitoring Algorithms Under Varying Bandwidths, 5/20% Loss

The results of the algorithm comparison with lossy networks are summarised in figure 9-7. We see that the slow reactions algorithm averages tend to be influenced by the previous / subsequent value (with the largest difference to the current simulated value where both exist). The 10th/90th percentiles are also substantially wider, reflecting the slow response to change. The fast response algorithm’s mean points are close to the jump algorithm’s mean points. However, the error bars for the jump algorithm are wider, reflecting the inaccuracies caused by inappropriate reaction to transient throughput changes caused by loss.

9.3.4 Round Trip Time

A similar test to that for throughput reactions was performed for round trip time. Here simulated round trip time was varied between 200ms and 1500ms. The bandwidth simulated was 20kB/s. The steps taken correspond to 1.25, 1.5 and 2* changes.

In figure 9-8 we see that the modelled and experienced time follow each other closely. There are transient discrepancies between the simulation and reality, which can be accounted for through addition delays experienced at the web server and the network between systems. The “bump” seen during some network conditions is due to a change in the density of features, being reflected in a change of overhead in transmitting the GET request and processing the selection at the server. The same trace is represented as a sequence of prediction errors in figure 9-9. The error degree around zero which constitutes noise once the model has settled is wider than for the throughput trace. This reflects the noise in response time which we noted above and which the simulation does not remove to the extent that it does for the throughput. We also see more clearly the error spikes around changes in the simulation, which are of the degree expected and again are resolved after three loads for the various degrees of change simulated.

Figure 9-10 illustrates the mean and 10th/90th percentile round trip times modelled for meta data and data. We see that the meta data takes longer to return than map data. This will be due in part to the look-up methods to access the data in each case (actual server response time is added to the simulated round trip time). The range of error remains under 250ms for all cases.

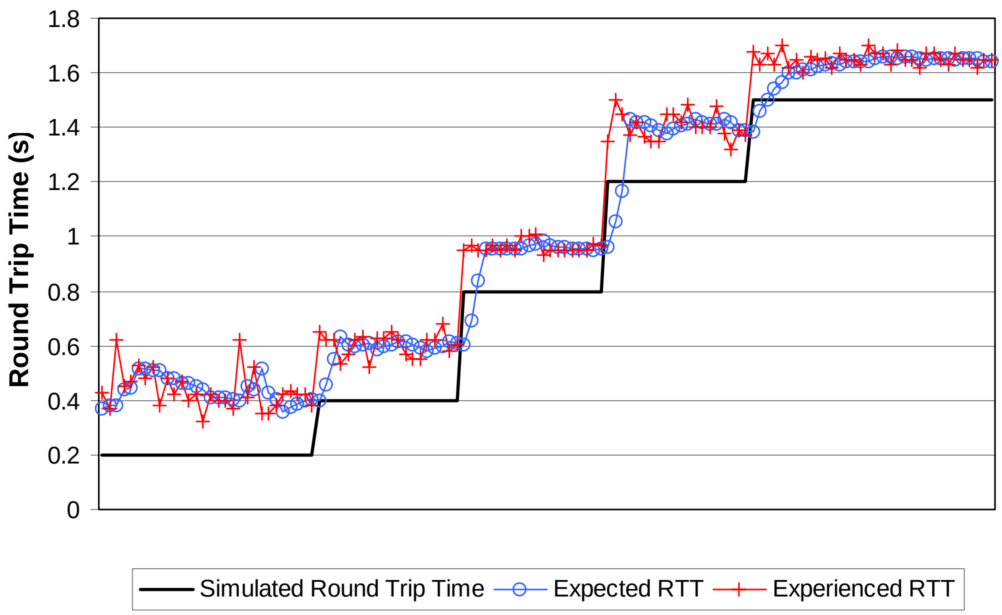


Figure 9-8: Response to Change –
Expected and Experienced Map Data Round Trip Time Trace

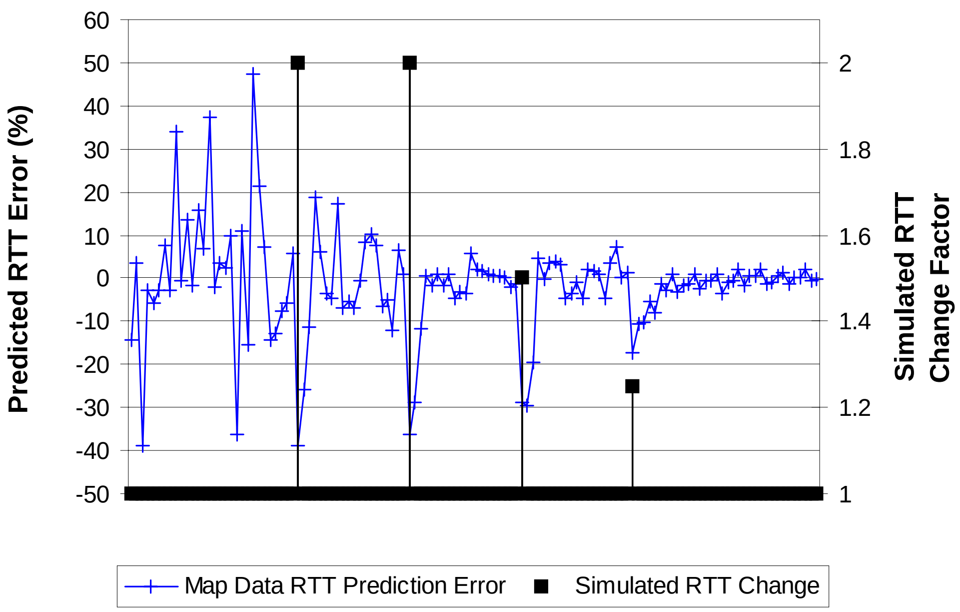


Figure 9-9: Response to Change –
Round Trip Time Prediction Error Trace

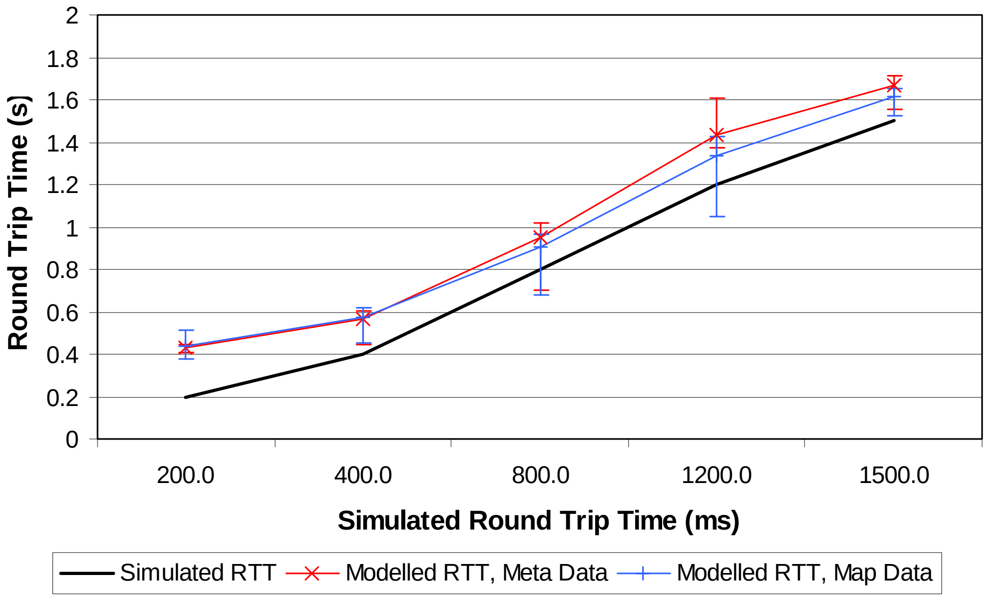


Figure 9-10: Response to Change – Average & 10/90th Percentile Round Trip Time Under Varying Degrees of Change

9.3.5 Range of Monitoring

Figure 9-11 illustrates how the throughput model and the simulation correspond over a wider range of bandwidths. This graph is extracted from deadline satisfaction and scalability tests, where bandwidth simulated was varied between 1 and 215kB/s stepping with even ratios from 1 to 2.15 to 4.64 to 10 etc. and round trip times simulated were 100, 500, 1000 and 2000ms. Each bandwidth / round trip time was loaded 20 times, with a deadline of 30s, the first three loads being omitted from these results to allow the model to adjust. The figure shows a graph of the throughput model at each simulation point and the simulated bandwidth. At the lower throughputs the points for different round trip times at each bandwidth are close together. This indicates that the calculation of throughput was not being altered substantially by changes in the delay or external network. The gradients of the two lines (test trend line and simulation values) are quite different. The monitored throughput is three to four times that of the simulation between 1k and 10kB/s, falling to equal to one tenth the simulation at higher speeds. As the simulated bandwidth increases the timing approaches the limits of the system. At 46kB/s the modelled throughput ceases to change significantly. The difference at high speed may also be due in part to inaccuracies in timing, although the spread of points for different round trip times indicates that the interaction between throughput and round trip time calculation may be a factor.

As for throughput we also plot a scatter graph of mean modelled round trip time against simulated round trip time, at a range of simulated bandwidths and delays, shown in figure 9-12. We see that the round trip time modelled varies with throughput (in a similar way to throughput's variation with round trip time). This also illustrates again the additional delay caused by the real system delays and the request length based delay over the simulated delay.

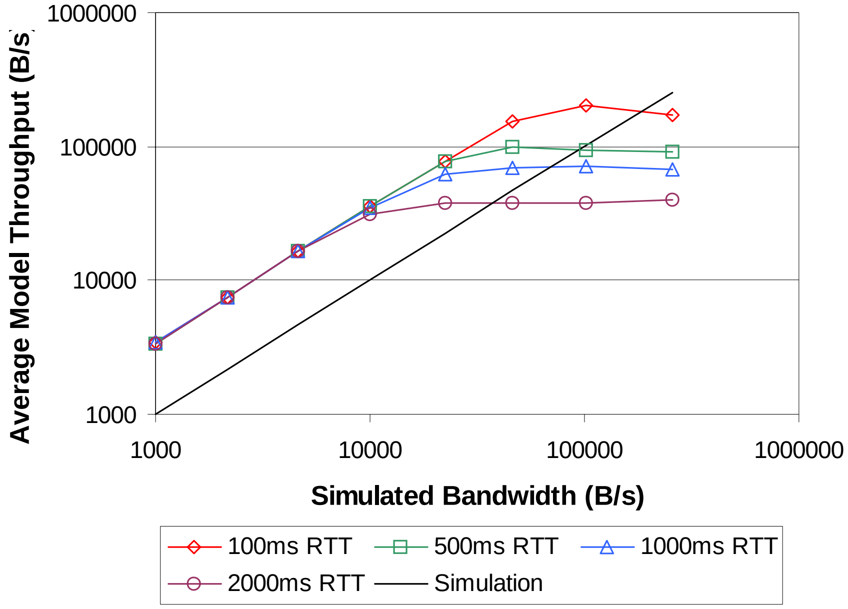


Figure 9-11: Range of Throughput Prediction

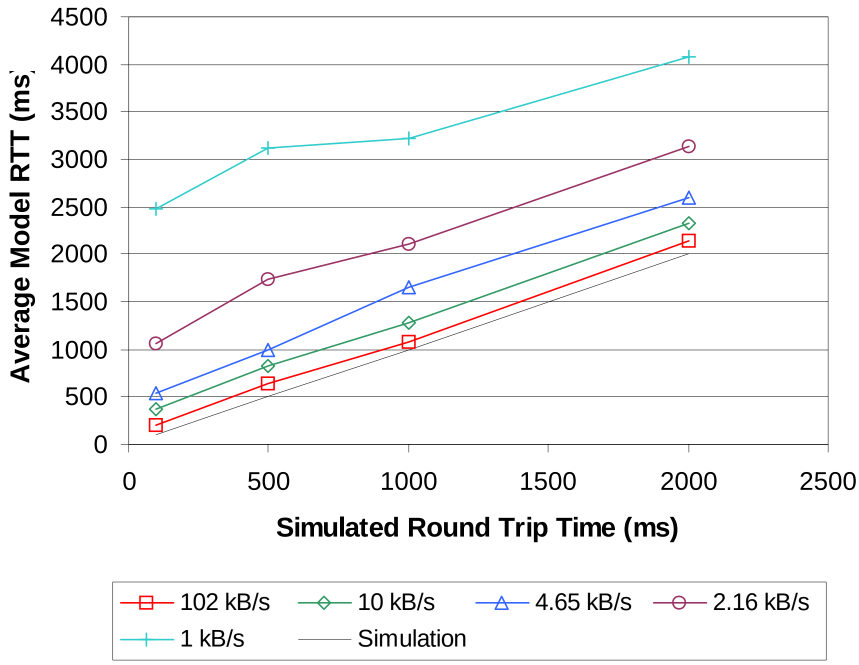


Figure 9-12: Range of Round Trip Time Prediction

9.3.6 Summary

In all these tests we have found the response to change to be acceptable – changes with a magnitude of between 1.5* and 4* were all reacted to within three loads while stability was maintained in the face of passing fluctuations. A smaller inter-dependence between throughput and round trip time would have been desirable. It should be noted that these results were obtained with deadlines which significantly limit the data loaded at higher round trip times and lower throughputs. Data size would affect both compression and the effect of timing errors on the monitor. It is encouraging that the model maintains stability as variable throughput due to loss is simulated, even for large loss rate simulations.

9.4 Deadline Satisfaction, Scalability and Overheads

9.4.1 Deadline Satisfaction

The quality of the network model is only a tool to enable deadline satisfaction in our system. It was necessary to show that a range of deadlines could be met over a range of throughputs and round trip times. We also demonstrated that the system offered some benefit in meeting deadlines and giving predictable download time over a control system, which performs no mediation. The limits of applicability of the mediation and the overheads incurred were also studied.

It should be noted that the range of network conditions and deadlines which show mediation is dependant in a large part on the range of data offered. The maps mediated in these tests described roads, administrative boundaries, tourist features and other amenities. Many possible features exist over these, including building use, contour details, hyperlinks, traffic information etc. For an increased overhead the range of data to be selected amongst could be substantially increased.

The tests were the same as those used to show the ability of network monitoring to handle a range of throughput and round trip time. Both bandwidth and round trip time were varied. Round trip times from 100ms to 2s and bandwidths of 256kB/s to 1kB/s were used. All round trip times were tested for all bandwidths. Three loads were made at the start of each setting, to allow the network model to adjust to the simulated network. At each network condition 20 loads were made: 10 each of 3*3km suburban and urban areas, as before. Loads were made with 20, 30 and 45s deadlines and a control case. The control performed the mediation but always took the “best selection” regardless of any resource constraints. The times reported for the control are without the meta data and selection overhead.

In figure 9-13 we show the scalability results, plotting the mean download times for each test condition, with 10th and 90th percentile error bars. We can see that for the higher throughputs the mediated application and control offer little difference in mean time. As the bandwidth moves below 10kB/s the managed applications' download times remain stable while the control's increases. The three deadline cases offer fairly stable download times, clearly separated and generally below the deadline time. The variance in the control is also much wider than for the managed cases, indicating that the mediation also offers a substantial benefit in predictability of download time. The variation in the control case arises in part from the variation in size and density of features in the map tiles requested.

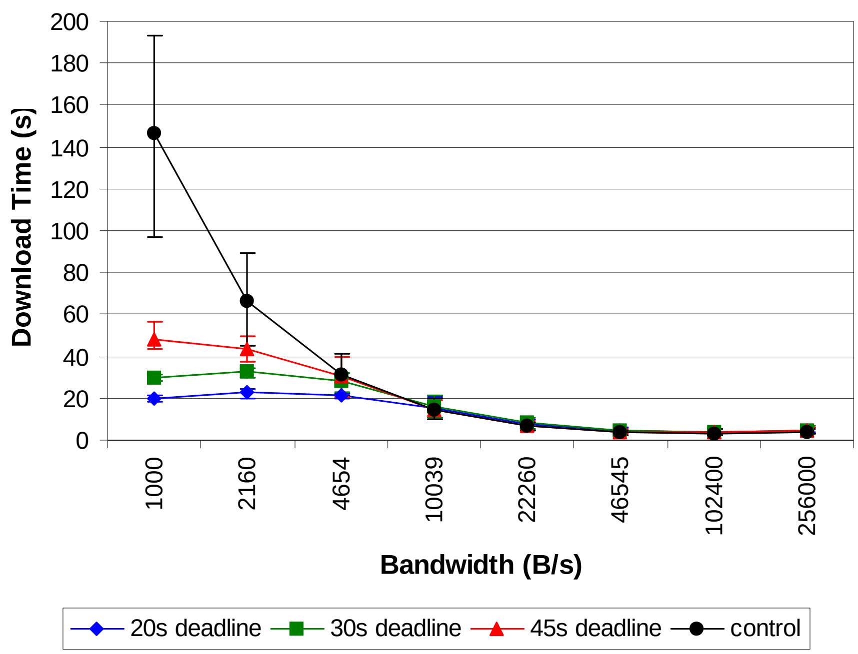


Figure 9-13: Scalability of Admission Control:
Time Taken Over Varying Network Conditions Compared to Unmediated Time Taken

In figure 9-14 we highlight the ability of the managed cases to meet the deadlines set. We plot the difference between the download time measured and deadline for each network condition simulated for all three deadlines. 10th and 90th percentiles are shown with the mean values in each case. Ideal results would show all points below zero. Where the selection made makes full use of available resources the points plotted would be close to zero, however we do not consider maximising resource use to be a goal: it relies on the granularity of selection possible and sufficient data being offered to use all resources, as well as an accurate resource model. As expected, the 45s deadline shows the greatest apparent inability to make full use of resources, however this is due in large part to there being insufficient data to make full use of the time available. At the slower throughputs we see the three lines running close together around the deadline. The inaccuracies arising from the dependency between throughput and round trip time we saw in the model earlier are visible here as failure to meet deadlines at lower throughputs. In the majority of conditions the deadlines are met and only exceeded by 5 seconds at most by the 90th percentile.

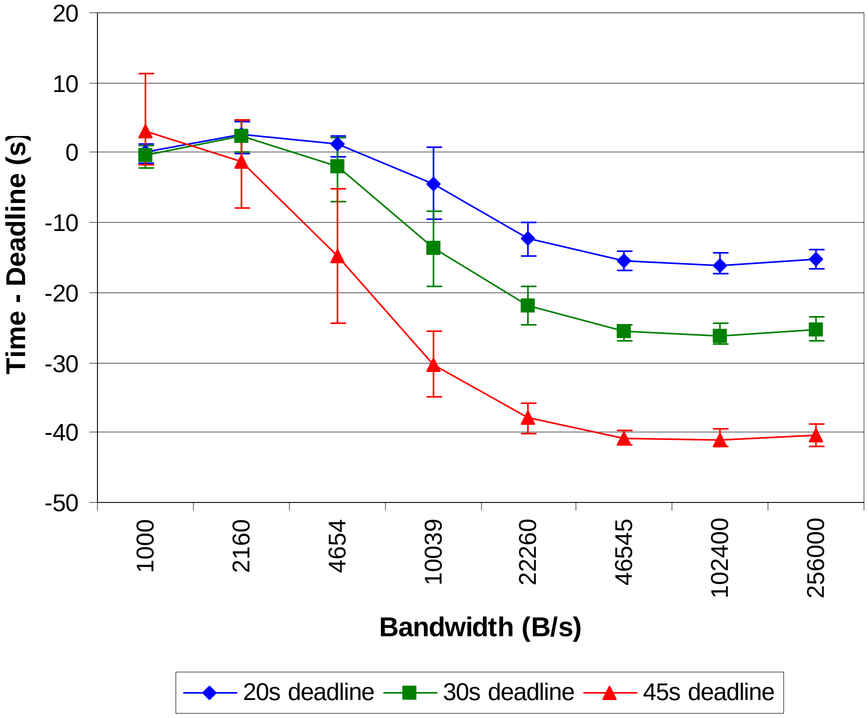


Figure 9-14: Scalability of Admission Control:
Difference to Deadline Over Varying Network Conditions

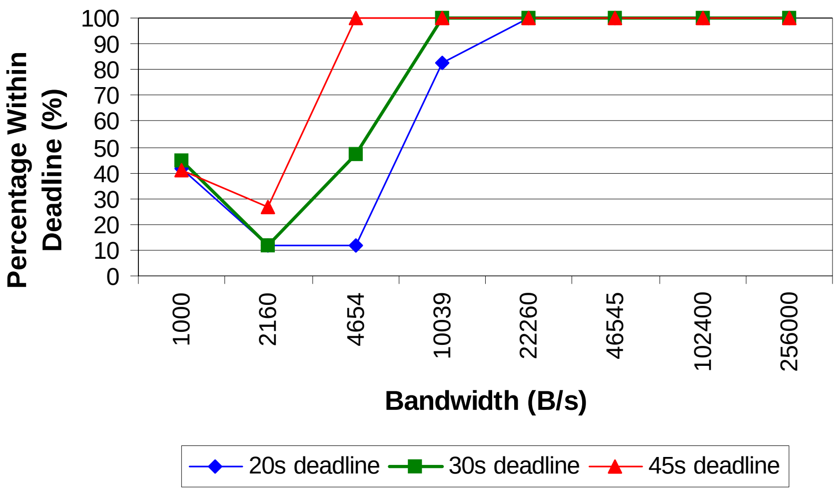


Figure 9-15: Quality of Admission Control:
Percentage Meeting Deadline Over Varying Network Conditions

In figure 9-15 we examine the ability to meet deadlines in a cruder form, simply considering whether the deadline was met for each simulated network condition. At worse network conditions not all requests are accepted as possible. The percentage represented for a failure out of data requests made therefore rises. For instance, the 42% success rate for the first point at 20s deadline represents five loads meeting the deadline out of the twelve attempted, five more which would otherwise count for the result were not attempted. These results should therefore be interpreted in light of figure 9-21, which indicates the proportion of requests for which some data was requested.

In figure 9-16 we examine the ability to meet deadlines as the round trip time is varied. In the portion of the graph where mediation is omitting some data (below 10kB/s) we see that for a wide range of round trip times the ability to meet deadlines is closely matched in all four cases. This indicates that the network model and mediation are acceptable across a wide range of round trip times. As the bandwidth increases beyond 10kB/s each case becomes able to make a best selection and the difference between cases reflects the difference in network performance.

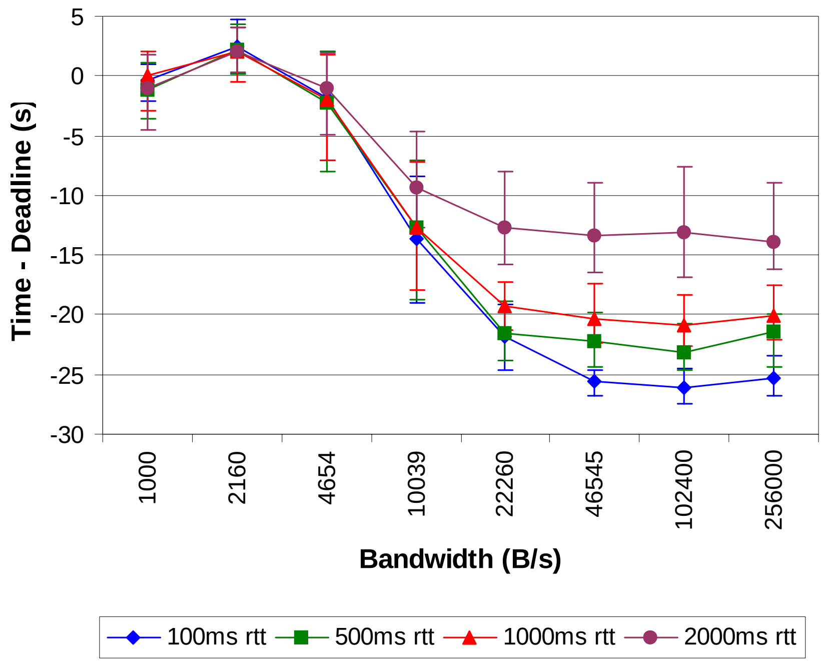


Figure 9-16: Quality of Admission Control:
Percentage Meeting Deadline Over Varying Network Conditions

9.4.2 Deadline Satisfaction with Simulated Loss

The deadline satisfaction tests were also run for a 30s deadline at 5/20% loss. The average download times, together with loss-less, control and 5/20% loss control are presented in figure 9-17. The managed tests have close mean times. The lossy loads do show a greater variation in some cases, but not a substantial degradation. On the other hand the unmanaged loads show a clear difference in mean download time between the no-loss and 5/20% lossy case and a still greater difference in the separation of the variation bars. The unmediated system therefore shows an increased variation in download time with loss compared to the mediated system.

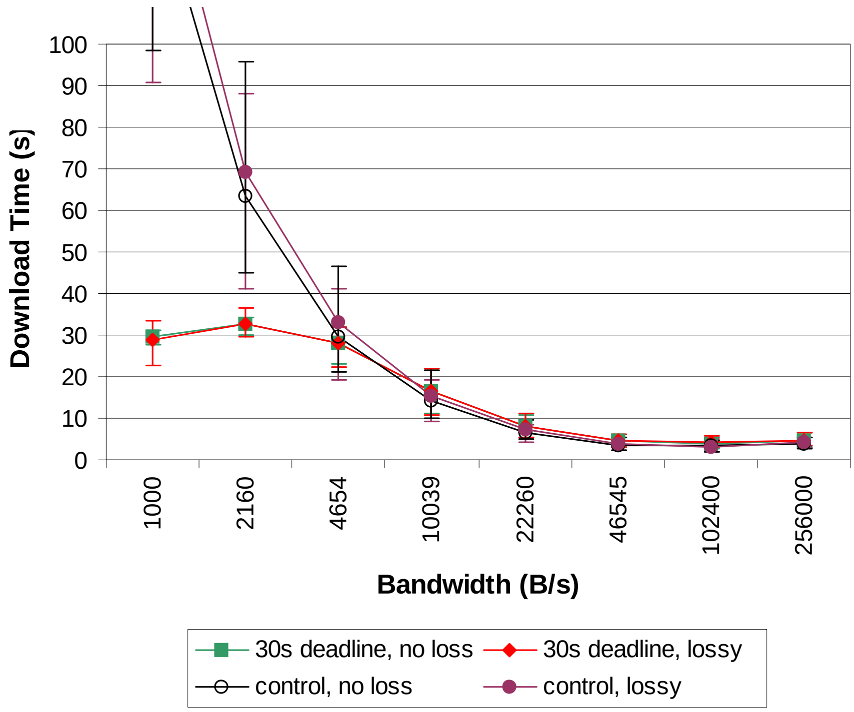


Figure 9-17: Scalability of Admission Control:
Average Times Over Varying Network Conditions with Simulated Loss

The plot in figure 9-18 illustrates more clearly the ability of the managed systems to meet deadlines under varying loss. The mean download times show little consistent separation. The 10th/90th percentiles show that some loads suffer in the lossy networks, as was expected. However, the difference in most cases is under 2 seconds. There are a small number of cases where the 90th percentile for the lossy cases exceeds that of the no loss case by a substantial amount, where the random loss had caused substantial problems in a few cases. Similarly there are a few cases where the 10th percentile for the lossy cases was substantially below that for the no loss case, reflecting an underestimate of the available network capacity after experiencing loss. The fast network / long round trip time problem exists as before.

Examining the percentage within the deadline for the different loss cases in figure 9-19, we see a similar ability to meet deadlines in most cases. In a few cases it seems that loss has caused underestimate of network resources and that deadlines have been met more easily by a less ambitious data selection.

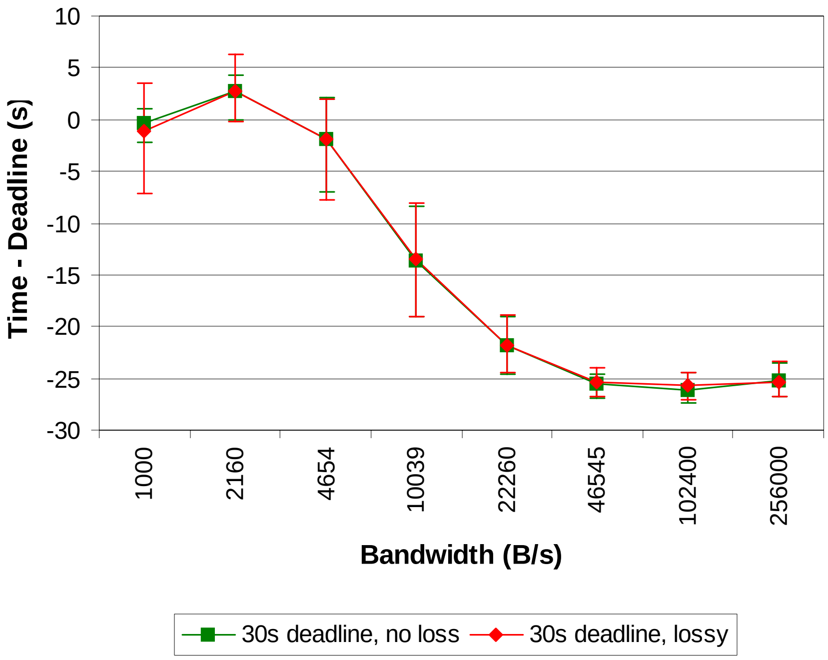


Figure 9-18: Scalability of Admission Control:
Difference to Deadline Over Varying Network Conditions with Simulated Loss

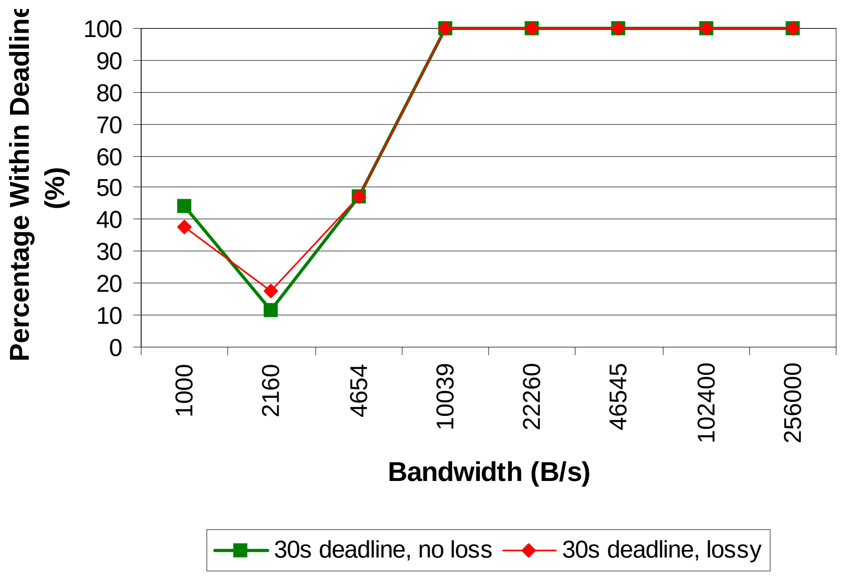


Figure 9-19: Quality of Admission Control:
Percentage Meeting Deadline Over Varying Network Conditions with Simulated Loss

9.4.3 Limits of Selection

We also stated that we wished to examine the range of applicability of the mediation technique across network simulations, given the available data. It is clear that with richer map data (e.g. including individual buildings, traffic information, contours) the range of possible selection becomes wider. The results presented here then illustrate the range of applicability with a basic data set.

Figure 9-20 shows where the “best” selection was taken – all data which has a weight of over zero is selected at the highest utility variant available. Best selection data need not be the same as no mediation. Zero weights were given to various categories of feature in our tests, as one would expect in real use. At low (100ms) round trip times best selections become possible at 4.7kB/s bandwidth for a 45s deadline to 22kB/s for a 20s deadline. At lower bandwidths there is a progression into best selection, representing lower density of map data (suburban vs. urban areas). In these cases the meta data overhead drops and the data size of the best selection drops, making best selection easier to achieve in two ways at once.

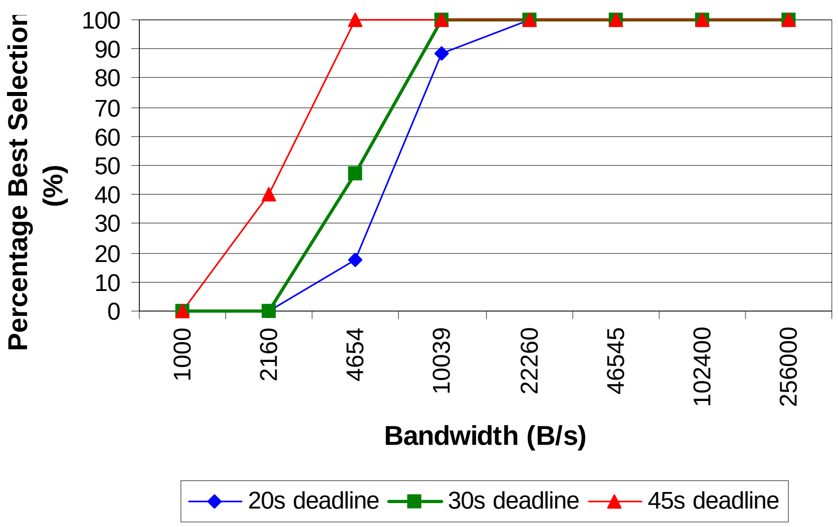


Figure 9-20: Limits of Selection – Taking Best Selection

With reference to the ability to meet deadlines at moderate bandwidths (10-46kB/s) it may be considered that mediation is providing less data for a similar time. However, mediation also offers the ability to remove uninteresting data from the user’s view (although this could also happen in the application itself, albeit as a less general solution) and to remove unnecessary data from the application’s consideration. If a memory resource manager were also applied it is likely that at some network capacity download time would cease to be the limiting factor and that the overheads negotiation place on the network resources would cease to be an issue. Contextual mediation also offers a general mechanism for data filtering to the user, rather than requiring many independent ones.

The converse case to best selection, of no/some selection, is presented in Figure 9-21. Here we show the proportion of cases where it was predicted that some data could be loaded within the deadline. We see that for the shorter deadlines of 20 and 30s this is a problem at 1kB/s and also occurs in a few cases at 2kB/s for the 20s deadline. This is to be expected and in real use could be used to engage the user in a revision of their expectations before making further requests. The main advantage in mediation is that expectations can be met with greater consistency than where mediation is not used. Where no data can be loaded within a deadline the user may choose to modify that deadline and be less disappointed when longer downloads occur or choose not to interact with the system rather than be frustrated by it.

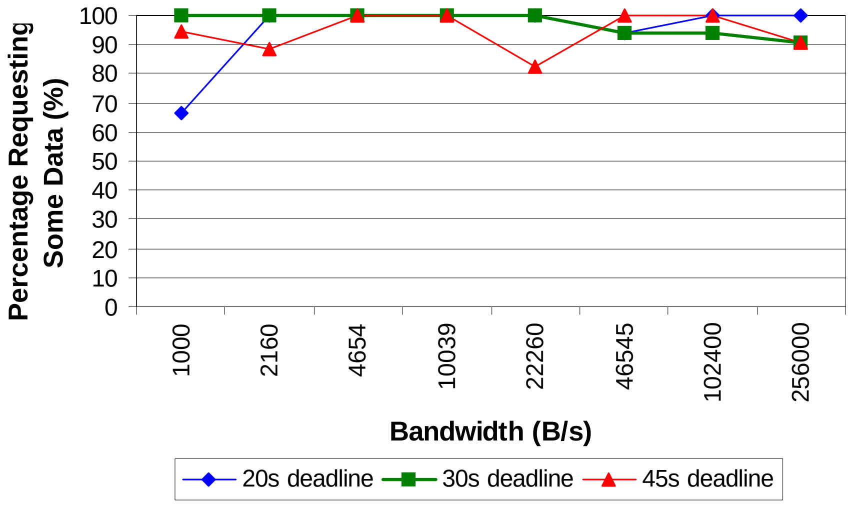


Figure 9-21: Limits of Selection – Making Some Selection

9.4.4 Overheads

The final data taken from the simulated loads was a measure of the overheads in time. The overhead is considered to be the time from the request leaving the application to the time the selection has been made (the bulk of which is the time to load the meta data). These are presented for the three deadlines in figure 9-22. As expected in most cases the overhead is similar in all three cases. The separation of the three deadlines at 1kB/s reflects the number of tests which failed to make any load and were not considered in the results. For 20s at 1kB/s those tests where any load was made would be those with the smallest areas and lowest feature density. The meta data overhead in these cases would be lowest and so the overhead time lower. Note that the variance also rises as the bandwidth becomes less. This is the same effect as the variation in the control load times – the different map tiles take different amounts of meta data to describe. The effect is less than for the control map loads as many features are grouped in the meta data and so the increase in meta data to transfer is not linear with change in potential map data. At lower bandwidths the overhead becomes substantial, however the benefits are arguably greater than this overhead. Our expectation that the selection process is not the most significant part of the overhead of mediation is borne out by the separation of meta data loading and selection processing times, in figure 9-23. The results in this bar chart are for the 30s deadline. Even at 256kB/s the overhead of meta loading is greater than that of selection on a 500MHz Pentium II system.

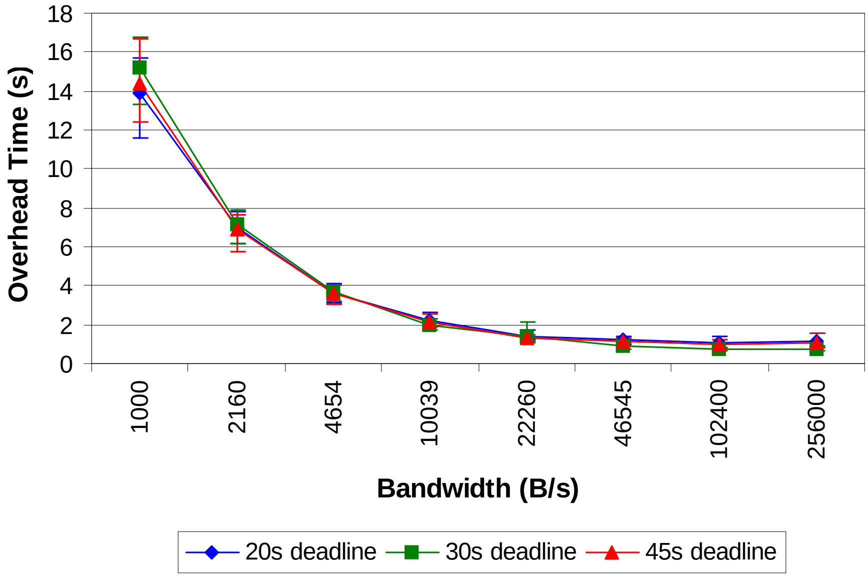


Figure 9-22: Overhead of Mediation at Varying Bandwidths and Deadlines

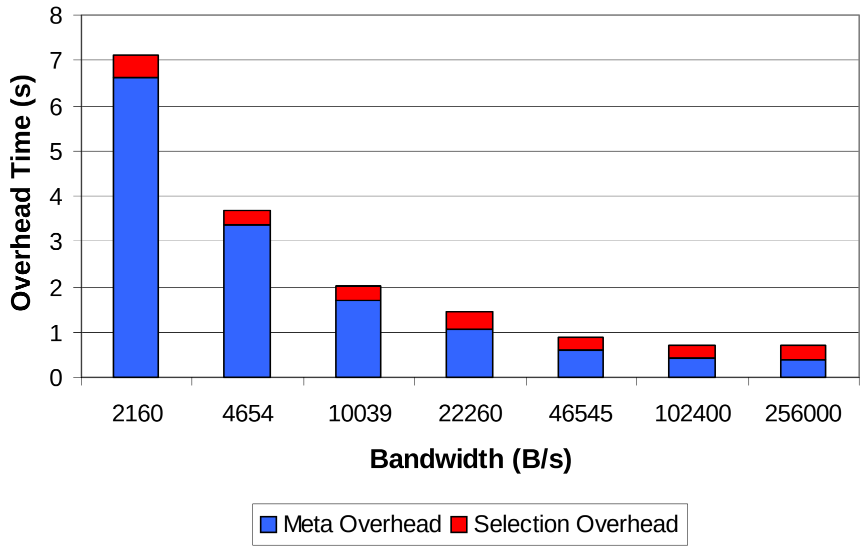


Figure 9-23: Overhead of Meta Data Load and Selection at Varying Bandwidths

Caching would also alleviate some of the overhead. Indeed caching may affect the meta data faster than the data. Meta data would be offered to all requesters, while not all data would be selected by each requester, so caches would not always have data cached. Meta data for dynamic data may not need to change as often as the data, particularly if the data creation time were not given. In addition to this the meta data format is a somewhat verbose XML encoding, useful for test and demonstration purposes, but not as efficient as possible even when compressed. The overhead incurred can be addressed in more than one way then in order to reduce its effect.

9.5 SCREEN DENSITY

Taking a 7.4*6.1km map area in London, we show in figure 9-24 six versions of the map with different density limitations. All maps have a screen size of 500*500 pixels and the context for each was driving a car, for work, at 25mph. The network capacity was set so that deadline was not a limiting factor for any of the maps. The top left map has a density limit of 5.0, which is greater than the highest density without mediation. This map simply shows the data eliminated by mediation due to preferences with no degradation from the best selection. The density limit drops (reading left to right, top to bottom) to 2.5, 1.5 and 1.0. The clarity of the maps remains useful and the mediation process ensures the useful data is shown in each case. However, at the lowest limit the specifications and density limit have resulted in a discontinuity in the road data. Applying any resource limit may result in data which is not as good as may be hoped for. There two obvious responses to this: increase the resource limit or zoom in so that there is less competition between features for screen space. The first option requires that the user becomes able to interpret the extra data, for instance by slowing down the car so the screen can be read more easily.

The actual density for each zone of the screen for the maps in figure 9-24 are shown in table 9-1. The 500*500 window size resulted in a 4*4 density model for the map data. Note the effect of the selection algorithm means that degradation beyond the zone limit occurs in some cases. For instance with a limit of 5.0 all but 5 zones have a density less than 2.5. However in the case where the limit is 2.5 all zones have a lower density. This is due to the consistency which the selection algorithm tries to maintain. It would be confusing if parts of the map were degraded more than adjacent parts without reason. For instance if the minor road detail in the highest density map were to be eliminated in the most dense areas to the right of the map but retained to the left of the map the adaptation would be confusing.

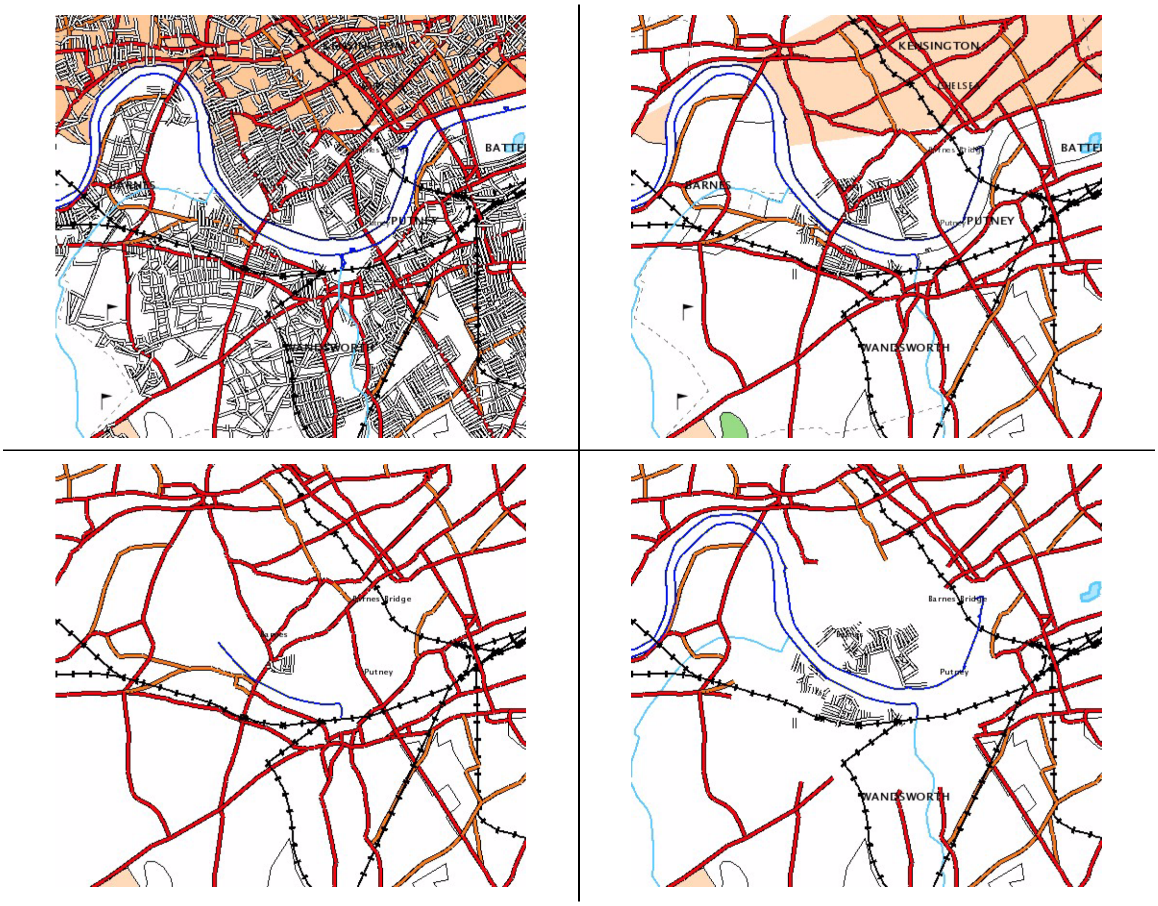


Figure 9-24: Meeting Screen Density Limits
Limits Were Top Left=5.0, Top Right=2.5, Bottom Left=1.5, Bottom Right=1.0

Density Limit	Zone Densities			
5.0	0.701	1.364	1.464	2.215
	1.000	2.160	2.352	3.042
	2.280	2.371	3.190	3.840
	2.180	2.932	2.268	4.489
2.5	0.405	0.765	0.867	1.343
	0.466	1.346	1.372	1.754
	1.163	1.441	1.952	2.265
	1.185	1.678	1.303	2.345
1.5	0.094	0.280	0.310	0.487
	0.111	1.047	1.103	0.644
	0.414	1.149	1.495	0.952
	0.517	0.697	0.558	0.759
1.0	0.126	0.248	0.345	0.533
	0.143	0.308	0.311	0.560
	0.446	0.344	0.428	0.890
	0.519	0.695	0.562	0.759

Table 9-1: Densities in Example Maps

9.6 DATA UTILITY

We now present a view of the selections made at various simulated network conditions by the two selection algorithms. Figure 9-25 presents selections made with the cost-based selection algorithm, figure 9-26 those made with the sequence-based algorithm. In all cases the map area loaded was 4km*4km in the location service, expanded to $19km^2$ by the application. A 30s deadline was used in all cases. The best selection and no selection points have moved slightly due to the increased map area in comparison to the scalability tests. The various network conditions are colour coded. At the far right is a *best selection*. Note that elements are not available for all weights at any given simulated condition. The vertical axis represents *utility ratio*. This is the utility of the variant selected over the highest utility variant available. A one indicates the best is selected, a zero indicates no selection. A grey floor to the chart at a weight / network condition indicates that there was no element available at that weight. The weights taken are in numerical order, higher being a stronger preference. A translation between the weights and element parameters is given in table 9-2. The weight depends on both the element type and the distance of the element's centre point from the centre of the map. A weight of zero is never selected.

It can be seen that the lower weight elements are dropped before higher weights in general, while smaller changes in selection are used to enable the deadlines to be met in other cases. Although the benefit in data size of the variants generated was small selections with a utility ratio between 0 and 1 are a common feature. The cost-based algorithm seems to use degraded variants more than the sequential algorithm, as small degradations across the weight range are taken earlier whereas the sequential algorithm applies weight order more heavily. The selections at 1.2kB/s seem marginal, however once 2.4kB/s is reached a substantial representation of the roads is being given, particularly towards the centre of the map. The selection gradually improves the range of features and their quality towards 12kB/s. So the mediation is performing a selection which is more complex than no data / marginal data vs best data for a range of network conditions representing almost an order of magnitude in bandwidth and a common range of download deadlines.

The specifications used are similar to those shown in appendix 2.3. The context matches give rise to the structure of specifications shown in figure 9-27. The correlation between types and weights are shown in table 9-2, to aid in interpreting the selection graphs.

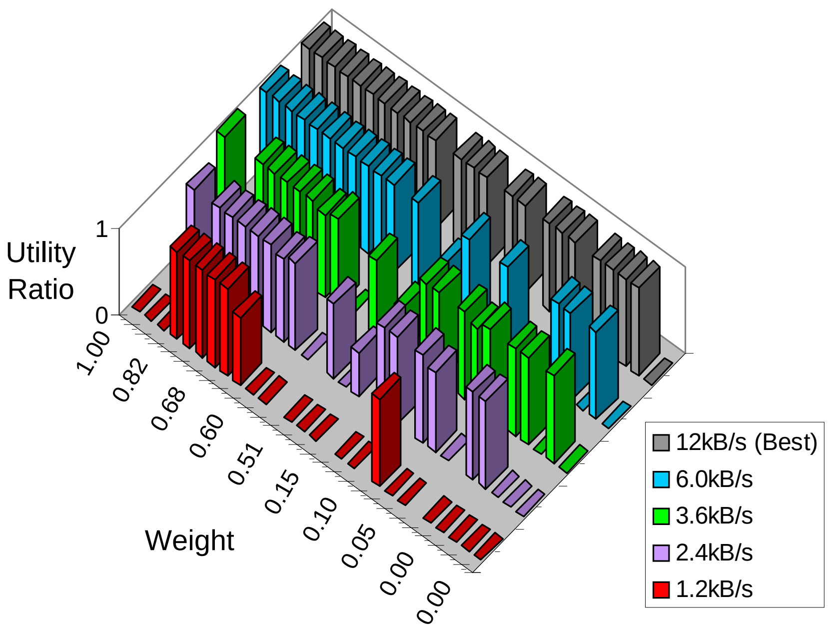


Figure 9-25: Effect on Quality Due to Network Conditions (Example Selections with 30s Deadline, Cost-Based Selection)

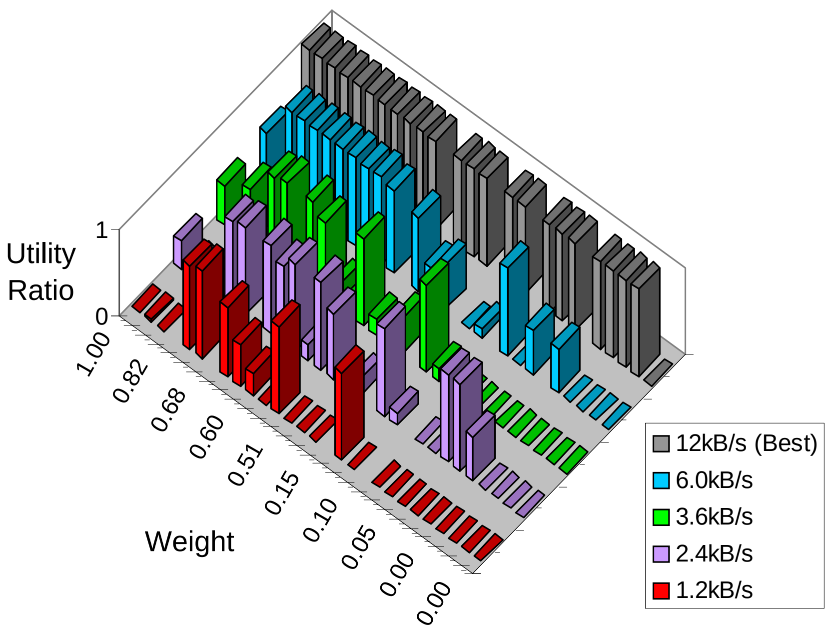


Figure 9-26: Effect on Quality Due to Network Conditions (Example Selections with 30s Deadline, Sequence-Based Selection)

Weight	Type(s) & Distances
0	other features (including path)
0.00003	* at 5-10km
0.00007	* at 2-5km
0.00009	* at 1-2km
0.0001	* at 0-1km
0.03	(topology, water) at 5-10km
0.045	railway at 5-10km
0.09	(topology, water) at 1-2km
0.1	(topology, water) at 0-1km
0.105	railway at 2-5km
0.135	railway at 1-2km
0.15	railway at 0-1km
0.219	building at 10-20km
0.27	antiquity at 10-20km, road:motorway at 10-20km
0.42	road:minor-road at 1km-5km, (road:a-road, road:dual-carriageway) at 10-20km
0.511	building at 2-5km
0.525	(building:church, building:lighthouse, building:windmill) at 2-5km
0.54	road:minor-road at 400m-1km
0.6	road:minor-road at 0-400m
0.63	antiquity at 2-5km, road:motorway at 5-10km
0.657	building at 1-2km
0.675	(building:church, building:lighthouse, building:windmill) at 1-2km
0.73	building at 0-1km
0.75	(building:church, building:lighthouse, building:windmill) at 0-1km
0.81	antiquity at 1-2km, road:motorway at 2-5km
0.82	(road:a-road, road:dual-carriageway) at 5-10km
0.9	antiquity at 0-1km, road:motorway at 0-2km
0.92	(road:a-road, road:b-road, road:dual-carriageway) at 2-5km
1	tourist feature at 0-1km, (road:a-road, road:b-road, road:dual-carriageway) at 0-2km

Table 9-2: Mapping of Types and Element Parameters from Weights in Example

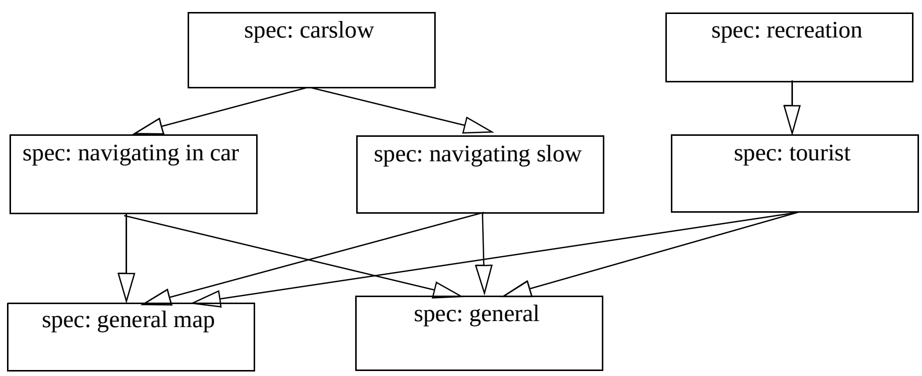


Figure 9-27: Context Matched Specifications In Tests

9.7 DATA UTILITY – USER TESTS

In addition to examining the performance of our application against simulated networks we performed an initial study into user's reactions to the maps which our system generated. The results of this are reported next.

9.7.1 Example Maps Used In Survey

A context of in-car travelling at 30mph for work was chosen. The context resulted in a deadline of 25s and a maximum density of 1.1. The simulated network was 3.1kB/s (approximately a good-quality GPRS connection) with a 100ms round trip time (shorter than most GPRS connections provide at this time). Screens of 240*320 pixels and 600*400 pixels were used, to simulate devices in the range of PDA to handheld-PC or dashboard device.

The map application developed had some technical, HCI and data distribution limitations which would have made an extensive user test with the application difficult. We chose instead to take example maps which demonstrated contextual mediation of maps being applied and provide them printed on card to a small group of users. This enabled us to make a preliminary study of the reaction to maps at these sizes and compare user reactions to mediated and unmediated maps. This approach removed any effects due to long and variable download times which mediation solves and the impact of HCI and device behaviour.

In many cases with the map data available to us there was no action as a result of the screen density limit and indeed there was little appropriate action. The map data was taken from an Ordnance Survey data set designed for printed maps, so unless an extreme zoom-out for the area was being taken the data was designed to be drawn together (and features are commonly adjusted to ensure clarity in printed maps). The application of the screen density resource was therefore not used in all test cases. If further data had been available, which was not designed to be used with this data set, then it would have been possible to test the use of this resource more thoroughly, as competition for space would increase.

Control maps were produced, using the *best selection algorithm* and the same screen size. The resulting map was saved as a JPEG with compression set such that the image file took approximately the same size as the NTF data was estimated to take when GZIPPed (closest size with integer JPEG compression factors). The intention was to simulate a proxy responding only to system limitations: the image was delivered at the correct pixel size and the download time would be within the deadline (assuming that the image management has similar overheads to our system). In all cases the unmanaged map application took more time than the deadline to load the vector map data alone (not including the meta data load and selection process).

The maps were of “inner-city” London and so contained quite dense road data. The mediation would be most necessary here, so the tests will show our system at its most beneficial. The maps used in the survey were printed at 100dpi, which is approximately the pixel pitch of LCD screens used in the classes of devices represented. The card maps therefore had dimensions similar to those of a PDA screen. These maps are shown here in table 9-3, however at 150dpi (in order to fit the page layout). Note that the areas given are the areas passed to the application from the location service, the actual areas loaded have been expanded to maintain uniform scales in each direction despite the aspect ratio of the screen.

A brief examination of the maps reveals a factor which we expect to be present in any real system, even if it makes mediation seem less appealing in our tests: In several cases the control and mediated maps are very similar, differing only in very minor or outlying features. When maps are almost identical then little difference in usability is expected to be found between them. In this case no preference is to be expected. The reverse condition is also demonstrated in the test maps, where road features required for navigation are obscured by tourist information symbols in the unmediated map and visible in the mediated map, due to the screen density management. In this case mediation should show an improvement in usability. Where goals are too restrictive for the data being requested, or specifications are poorly written, then the mediation may remove necessary data and result in a less usable map.

9.7.2 Survey Results

Our user test consisted of two parts. First a simple survey to help gauge the experience of our subjects with mobile systems and maps, plus some basic demographic data. The majority (although not all) of the 14 subjects were staff and research students within the Department of Computing. Most subjects were between 25 and 40, although users under 25 and over 60 were included. The male / female ratio was 9:5 All used a desktop computer somewhat, most for at least 20 hours a week and all used the web on their computer. Over half had a laptop and used the web on this too. Slightly less than half had a PDA and of those none used the web through it. All participants used road atlases and map web pages at least once in the last year, a substantial majority using both at least once a month. A few participants used more detailed maps for walking, sport, or work. No subjects used in-car navigation systems.

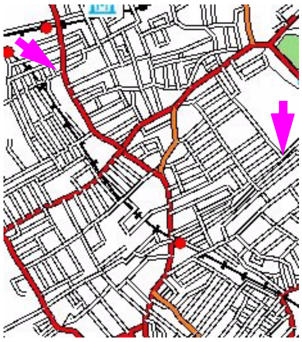
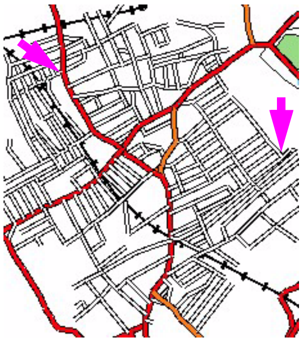
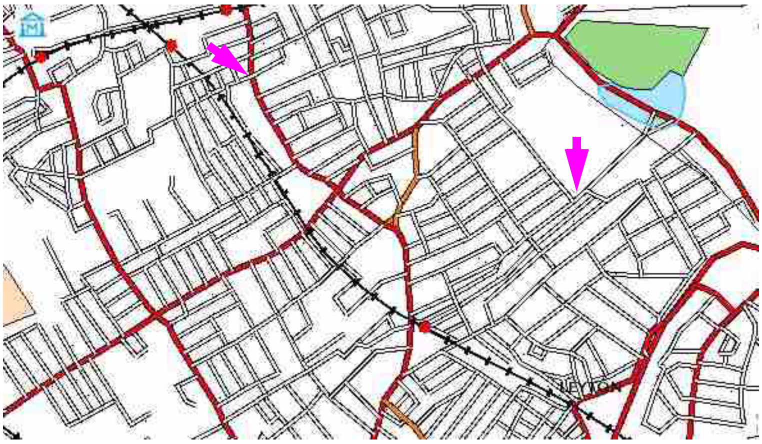
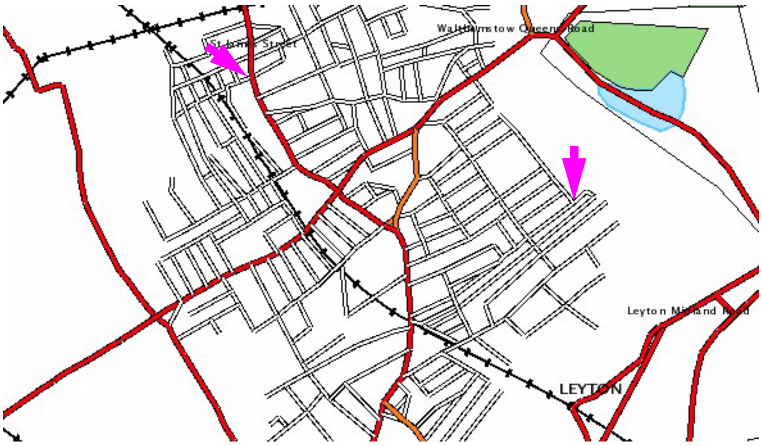
Map Area and Type	Map Image
2*2km small screen control, mediated	 
2*2km large screen control	
2*2km large screen mediated	

Table 9-3: Map Images Used in User Tests

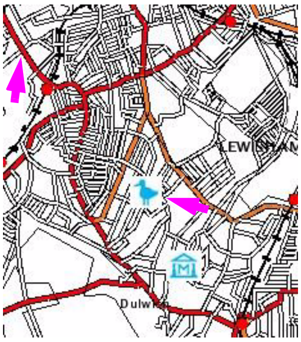
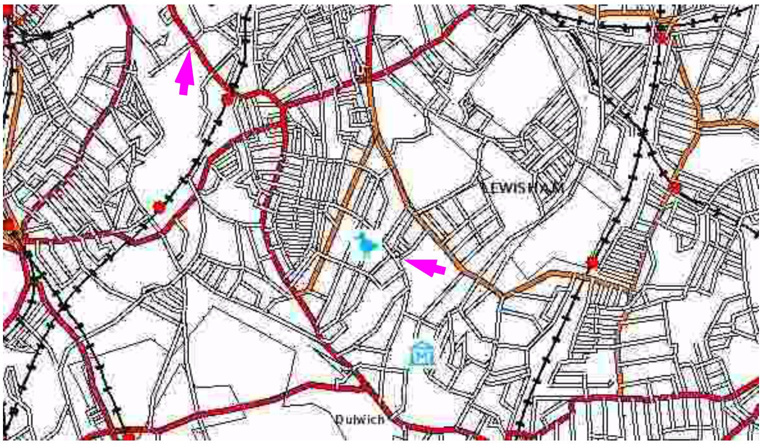
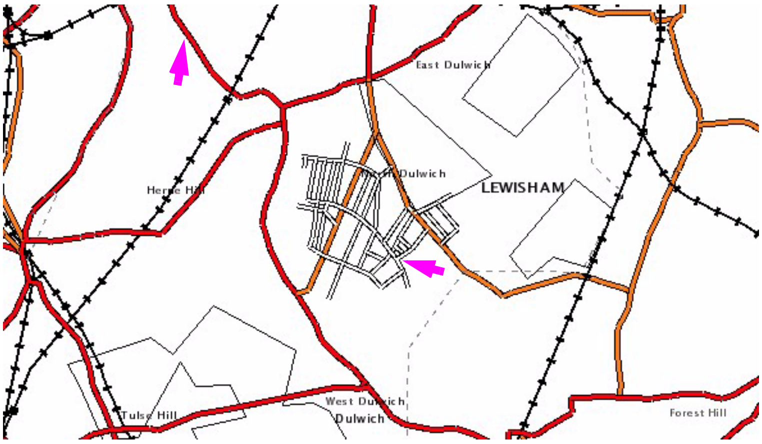
Map Area and Type	Map Image
3*3km small screen control, mediated	 
3*3km large screen control	
3*3km large screen mediated	

Table 9-3: Map Images Used in User Tests

In the next part of the test we gave the subjects a series of three of the test maps. Large and small mediated maps and a control map. The specifications used were as described in chapter 8 and shown in some detail in appendix 2.3. The context used in generating the maps was in-car, as a tourist at 30mph. For each they were asked to put themselves in the context of navigating in a car (either driving or as a passenger – the scale is similar to or larger than a road atlas, so the maps would be most appropriate for a passenger as the navigator) and to describe the route taken between the marked points. Features used to identify places were to be described. The actual route taken was not noted, but the time taken to describe each was. At this time a qualitative judgement of how easy the map was to use was also taken. The order varied with the control being first in about half the cases, the small mediated map first in the other half. As the maps essentially took the form of road maps lack of familiarity was not considered to be a big problem although the trend was to speed up slightly by the third map. The change of order was designed to counter this effect in the results.

Once all three descriptions had been given the subjects were also asked if they felt that the mediated maps were missing useful information and to place the three in order of preference as a navigational aid.

The first and most striking result of the timings was the variation in times taken to describe routes of similar complexity. The range of times to describe the route was 5 to 64 seconds. All subjects were consistently fast or slow in broad terms. To present the times to describe the routes the timings needed normalisation. We took the difference between the mediated map and the unmediated map and divided this by the unmediated map time. This then gives the proportion of the time taken by which the mediated map is faster or slower to use. We present the mean, 10th and 90th percentile times for the two mediated maps in figure 9-28 on the horizontal axis. A negative value indicates that the mediated map was faster.

We also wished to present the qualitative response to the different maps. The vertical axis of figure 9-28 gives a score. A higher score indicates a preference for the mediated map. The score was calculated as the sum of the scores for each question shown, as shown in table 9-4. A comparison between the small and large mediated was also made. A higher score indicates a subjective preference for the mediated map over the unmediated map (or small over large mediated).

Figure 9-28 shows us that the large mediated map was faster to use on average than the unmediated map, while the small mediated map was slower. The range of times for the large map was also substantially less than for the small map, reflecting a general preference for the larger map. Even in cases where the mediated map was found to be slower it was slower by a smaller degree for the large map. The large mediated map was faster than the unmediated map by slightly more than the improvement the large map showed over the small map, suggesting mediation can offer similar benefits to a change of form factor. The speed difference was not substantial for either small or large mediated map over unmediated, the real benefits in speed of use would come as the map became more cluttered. The result assures us however that the loss of information mediation causes does not have a large adverse impact on map usability for basic map data. The subjective scores in fact indicate that in general users prefer the mediated map, which at worst is similar to the unmediated one and at best is less cluttered. Again these scores reflect the preference for large maps over small.

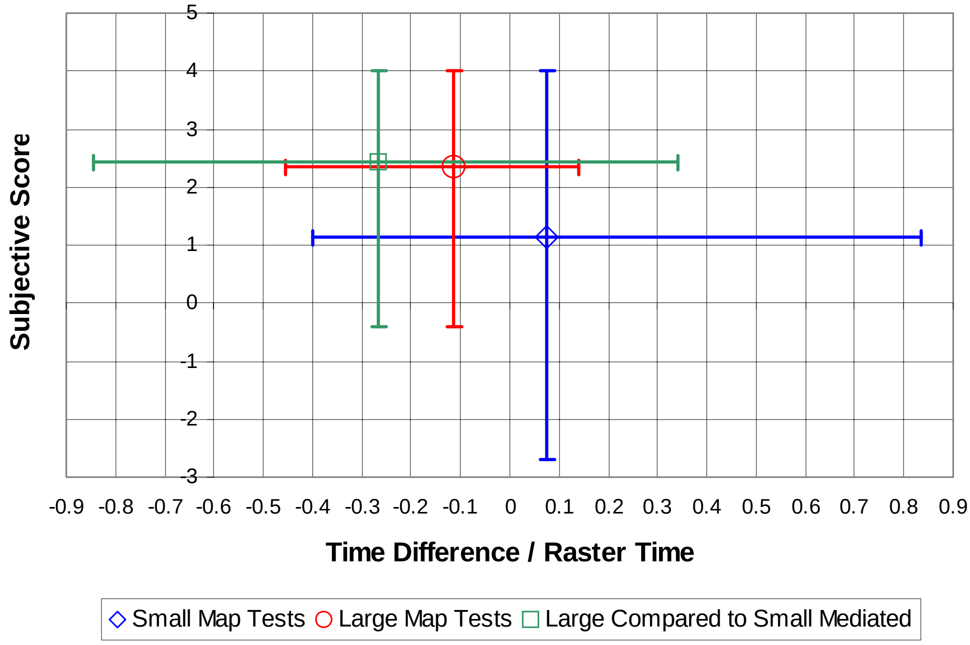


Figure 9-28: Ratings from User Tests

Question	Answer	Score
Did you find this map easy to use?	No, hard to use	-1
¨	Mostly OK, but difficult in places	0
¨	Yes, easy to use	1
Did you find the map easy to use? (This is then compared to control case ease of use.)	Easier than control (“easy”, control “mostly ok”)	-1
¨	The same as control	0
¨	Harder than control (“hard”, control “mostly ok” or “mostly ok”, control “easy”)	1
Did you feel that the map lacked useful information compared to the final map?	Yes a substantial amount. (This answer was not given by any subject.)	-2
¨	Yes a little, which would make navigation a little harder	-1
¨	Not enough to make a significant difference to navigation	1
Place in order of preference the three maps	Below control	-1
¨	Better than control	1

Table 9-4: Calculation of Subjective Score

Comments by test subjects and observations of their response to the maps were recorded during tests and also capture responses to the tests:

> *Two subjects noted that they took a different route for the mediated and control maps where minor road features were obscured. One assumed that roads would continue where obscured and used these in their route.*

These two subjects took longer and/or more complex routes in the unmediated case, as roads on the best route were obscured by other, irrelevant, features. This is a clear demonstration of the potential benefit of mediation – the most important data can be revealed and extra information only included where the user has sufficient interest and the context of use make its display appropriate.

> *While the mediated map would be easier to use in a car, it may be harder to use for planning routes as minor road data (and so route options) are missing.*

> *More detail may be required around destinations, not just in the centre of the map. A path of interest would be a more accurate representation of need than a point of interest.*

The intuitive response of one user quickly separated the tasks of planning a route and navigating a route. Several users found that the detail around the centre was appropriate for routes in this area but noted that where their route took them away from the central area the lack of detail might be a problem. A "location of interest" model which provided a path of interest (possibly with the aid of a route planning application" would therefore be a desirable future development.

> *In the larger area maps some subjects still found it cluttered and would have zoomed if using the actual application.*

> *Lack of road numbers / names was cited as a problem by several subjects.*

The drawing of the map was not ideal and the presentation was not that chosen by all users. While these subjects tended to mark all the maps down for the deficiencies in presentation these issues can be rectified with further development of the application and map rendering. The tests as performed may not have shown the maps in their best light, but still provided a positive response overall.

> *One subject commented on the poorer quality of drawing in the control map.*

The image degradation was not substantial for most of the maps, so it is not surprising that few subjects responded consciously to the lower quality of the graphics.

9.7.3 Summary of Usability Study

Our initial user study is encouraging. It shows that in general users responded positively to the mediated map and the loss of information did not have an adverse effect in time to plan routes using the map. The application lacks the refinement which would have made a more substantial user test a useful proposition (in particular the map rendering and location represented as a point rather than a path). The tests carried out were made using basic map data and with no interaction possible. Where the range of data which might be displayed is wider then the benefits are likely to become greater. Where the level of user interaction is greater then user control over the map displayed is possible, such as zooming and modifying goals, which we expect to enhance usability.

9.8 SUMMARY

In this chapter we have presented tests on an implementation which examines various factors which were discussed earlier in this thesis.

We have found that a simple model of network resources at an application level is capable of reacting to changes in network conditions within four data requests, while remaining stable and accurate in the face of loss (and the variation in characteristics that this manifests itself as). Our tests of scalability of mediation and ability to meet deadlines show selections which load in close to but just under the deadline in the majority of cases, even when subject to loss. This demonstrates that the network models are sufficiently accurate for the application domain and the 20-60 second deadlines for data loading which are our range of interest. The range of variation found shows that download time becomes quite predictable. The comparison between unmediated and mediated load times is favourable, particularly at lower throughputs where lack of mediation results in very long and variable data loading times. At higher throughputs mediation does not present an unreasonable overhead or loss of performance.

The selections engaged in as the networks change illustrate that the range of simulated network conditions which result in a selection which is not a simple no data vs best data selection cover an order of magnitude in bandwidth. The network conditions which are most interesting are those in which the application is most applicable with the technology which is becoming available at this time. Bandwidths of 1 to 10kB/s and round trip times of up to 2s represent current and future mobile and nomadic network access technologies. This space is also where our network models perform best. With an increase in the richness of data available the useful selection space expands rather than moving. As network speed increases, or data size decreases, deadlines of under 20 seconds become practical.

The preliminary user tests performed are encouraging. They illustrate that the loss of data which mediation entails does not create a problem in interpreting maps in most cases and in many in creates an advantage. The mediation here was performed in the context of a realistic network simulation and a PDA level screen. The screen density management aided in providing usable maps.

The usability study should be revisited once improvements have been made to the map application and it can be used without interfering with the tests of the mediation principles.

Chapter 10 Case Study - Web Pages and Hyperlinks

10.1 Introduction

We have concentrated on mediation of map data in this thesis. However, it is our intention that the techniques may be applied to other forms of application and data. Web pages present themselves as a good example. We addressed some of the prior work in this area in chapter 2. Web browser / server interactions have a similar form to the simple request / data delivery we have seen in maps. Pages are constructed of many elements many of which can be presented in alternative forms or omitted without the document as a whole being rendered valueless. The data carries information with a wide range of semantic content and is used to encode data to support many different tasks. Contextual mediation is therefore an applicable technique and web data a suitable medium for applying our techniques.

10.2 Issues in Mediating Web Pages

Although we are confident that contextual mediation is an applicable technique, the mediation of web pages is somewhat different to vector map data:

- The data are in various formats, text, images as GIF, JPEG, PNG, programs etc.
- The variants of images may well occupy different spaces on a screen, as well as offering different qualities.
- The selection of variants is more likely to affect the structure of the web page. Different variants of the HTML will result in different images being included, etc.

The total size of the page's data may not be as large as for map data. In all but the most limited cases (of throughput / deadline combination) meeting deadlines may not be the limiting factor. Where throughput is limited the mediation will still tend to result in the degradation of the most significant elements, as these will also tend to be the largest (in bytes), particularly in the case of images. The speed of the server is also a significant issue and may be more limiting than the last-hop network for better-enabled devices where pages are not cached. However, slow loading of web pages is still an issue and if consistency in presentation can be achieved in this respect, without constraining page designers to the lowest common denominator, then the application of mediation will immediately be a benefit.

The size of screen may well be the greatest limitation in the ubiquitous computing scenario. As WAP on 2nd generation mobile phones has shown there is no substitute for screen space in usability terms. However with appropriate data and design it is not necessary to abandon expectations of usability on a large range of displays. The layout of pages, both in the major components of the page and the decoration, can be adapted for more limited displays. Where the screen is small, poorly lit or vibrating various techniques have been shown to be effective in improving usability, including: reducing clutter, increasing font size, limiting the need for scrolling through structure and simplification of content. These needs can be reflected in our approach to contextual mediation.

The organisation of a web page does however afford a range of choices for adaptation. In addition to presenting reduced variants of data or eliminating some elements the hypertext structure may be used to break the web page into fragments, each of which meets needs for download time, screen space etc. An example of different approaches to device-independent access to web pages is described by Bickmore and Schilit in [13]. Jones et. al. consider usability of web pages with small screens in [67].

Weights may encode preferences for data identified as carrying the main information in a document. Weights can also differentiate navigational and decorative content. Having loaded data its presentation may be modified according to its importance to the user. For instance, HTML need not require a specific font size but a browser which has identified one body of text as more important than another may well be in a position to select the most appropriate rendering of each. In addition to modifying size the order of elements may be modified. Work on the semantic web [10] is moving towards providing a standard description of the semantic content of media elements in the WWW. This has the same basis as the typing in our meta data and we could make use of this in providing for contextual mediation. The weights reflecting type are therefore a realistic proposition for the future and supported by standards under development at this time. Types in web pages will to some degree represent common element types, e.g. contents list, navigation button, logo, main text, while other types will require definition within the subject area of the document in order to provide the best support for differentiation and clearest definition, e.g. home news, foreign news, family photo etc.

Utilities can encode preferences for data formats, languages for text and size for images. These descriptions can help eliminate data which cannot be understood by the device or user. In the case of images, loading the most appropriate can save time and client-side processing, e.g. in reducing an image to fit within the screen. This part of the mediation has similarities to that provided by HTTP [59], CONNEG [73] and CC/PP [99] and others in the literature – it is a clear and central need for adapting to mobile devices. Where the screen is not an appropriate output format, in very poor lighting or for visually impaired users, variants of data may be audio files. Text is amenable to client, proxy or server side text to speech translation. Audio versions may be given by the author (common in the case of news stories), read by a professional or computer generated. Descriptions of images and audio cues for navigational buttons may also be provided.

Consider a news page, formed from a series of text elements listing stories in different categories. These elements may be typed as “headlines”, “world”, “europe”, “home”, “sport”, “financial” etc. A user may describe preferences for these with weights, for instance ranking “sport” lower than “financial”. The variants of each element may contain different numbers of stories and/or use headlines or headlines plus a brief paragraph. The order and presentation of the elements could be modified by context aware rendering as well as the mediation selecting limited versions of less interesting categories.

The grouping within the selection can help ensure consistency in presentation, for instance by ensuring a series of different icons with related functions are selected with similar sizes. The selection of common languages in text is not guaranteed by our approach, although the utility arising from different languages would be applied across all applicable elements. Further work may investigate the need for and techniques to achieve consistency across differently weighted elements or in certain parameters of variants.

10.2.1 Web Page Mediation

We present our mediation process again, in this case walking through the load of a web page on two different devices. The impact of context on the selections made will be highlighted here, rather than the process of selection. The contexts may be thought of as a laptop and a PDA. The laptop will have a larger, higher resolution screen than the PDA. The PDA is also likely to be moving more than the laptop, e.g. if both were on a train but the laptop on a table and the PDA hand-held. Otherwise we shall assume similar network characteristics, deadlines, tasks, interests, eyesight etc. in the two cases. The network is somewhat limited with respect to the throughput, requiring some reduction in the data loaded compared to the unmediated version.

We shall now introduce the example web page we shall consider. This is a simple news site mock-up, which contains a series of blocks of HTML linking to stories in different categories: headlines, home news, foreign news, city news, links to other services, navigation buttons etc. The headline story in each category is illustrated with an image. In figure 10-1 we show the unmediated web page, which uses default renderings of the largest icons, images and the fullest text available. The styling of the web page is limited to a simple background texture and coloured separation bars.

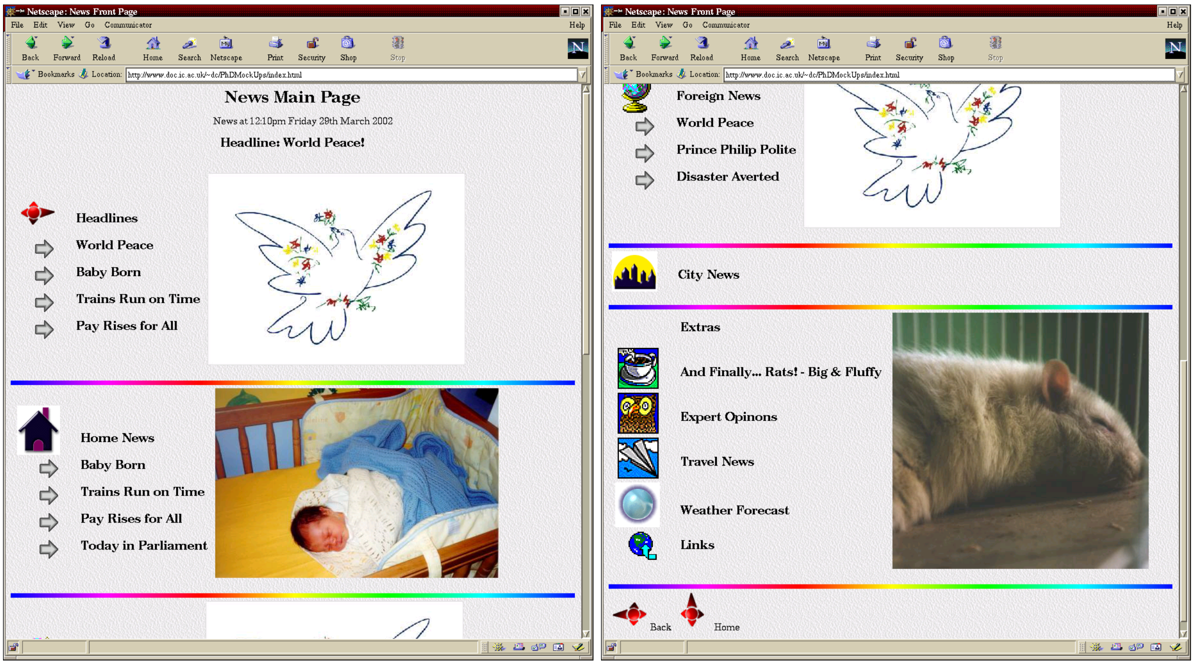


Figure 10-1: Basic Web Page

On a laptop the page is only adapted somewhat, as shown in figure 10-2. The screen space use has been limited so that the data should fit within two browser screens. This allows the user to avoid a long, detailed list of stories on this menu page and to reduce the need for scrolling. The "headlines" section of the page remains largely unaltered, being the most important to this user. The other sections: "foreign news", "home news" etc. are reduced to their category title. The stories are all available from the page linked to by the category should the user wish to look further than the headlines. The story pages may allow more information on the screen to present a detailed story. The images become smaller and are compressed– in part to meet a download deadline and in part to fit within the screen. Utility functions have described a preference for images no more than 75% of the width or height of the screen. This causes the largest images to be rejected. The screen space reservation took another step in the trade-off between the extra text in the headlines and the image size. JPEG compression to a quality index of 50 accounts for the rest of the data size loss. Icons are still in their large form – neither the screen space use nor the download time are overly impacted by these. The background texture and separators are also retained – the screen density allows for the extra detail in the background and their size is again not significant.

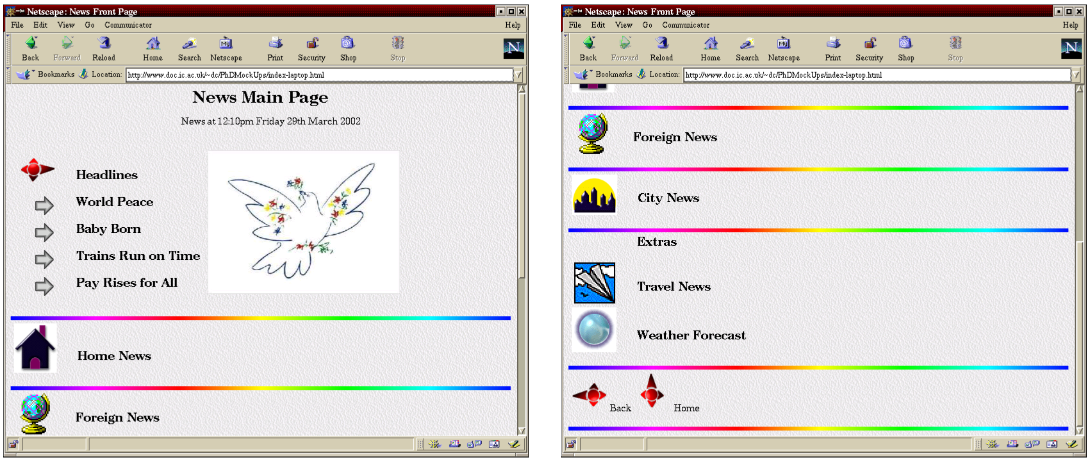


Figure 10-2: Web Page for Laptop

On the PDA the page undergoes a rather greater transformation, as shown in figure 10-3. The screen space use is again limited to two browser screens, although here that is a rather greater limitation. The “headlines” section of the page retains the list of stories, being the most important to this user. The other sections are again reduced to their category title. This reduction of a page to a series of links to sections of the original page, to reduce data size and the need for scrolling, has been presented in the literature for a variety of devices and types of page.

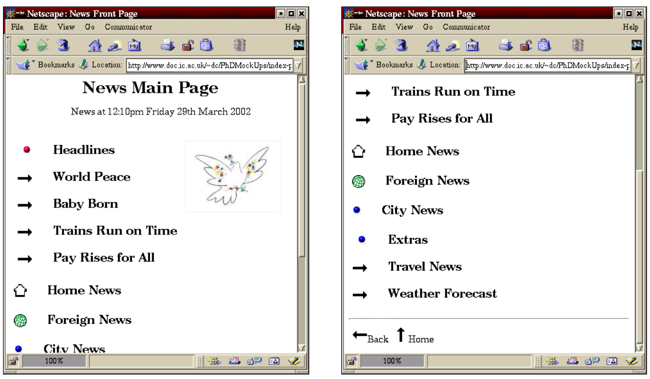


Figure 10-3: Web Page with PDA

The images become even smaller – in order to fit within the screen. The utility functions have limited the images to 75% of the screen size as before, however the screen space limit has taken the degradation of the headline image still further to fit the other more important information on the screen. Given the size reduction the JPEG compression index need only be set to 60 in order to meet the deadline with the other data. The icons are also reduced to smaller simpler figures, according to a utility function for icon type images, which limits their size to a small proportion of the screen space. The background texture and separators are removed – the screen density does not allow for the extra clutter in the poorer reading conditions. Overall the decorative graphics are not entirely discarded, but are reduced in size in line with their less important role within the page. Conversely the rendering has enlarged the font size of the more important parts of the text to aid readability, making best use of the screen space available.

10.2.2 Image Adaptation

Our approach to adaptation and utility functions is similar to that in Lehikinen and Walpole et. al. [78,115]. Linking of the technical impact and user studies on image degradation with a view to adapting media are presented by McIlhagga et. al. [64] and Chandra et. al. [33]. However, we shall give a brief example here to aid understanding. A JPEG image may be given at various sizes, compression ratios and colour depths. These factors all impact the perceived quality of the image and the data size. Image size also impacts the space taken on the screen. These trade-offs are similar to those for map data illustrated in chapter 4 and chapter 6. In figure 10-4 we show at the top-left a sketch graph of variants of an image which have been scaled to different sizes and had lossy compression applied to different degrees. The different variants are colour-coded by their point. In the top-right these same variants are compared by data size vs. utility. The colour-coding of the variants corresponds to that in the previous graph. The utility is taken from utility functions which consider larger images to be more important and lower compression to be preferable. While there are no scales shown the data size scale is taken from an actual image and is plotted approximately linearly. Three of the images variants are shown to illustrate the different renderings.

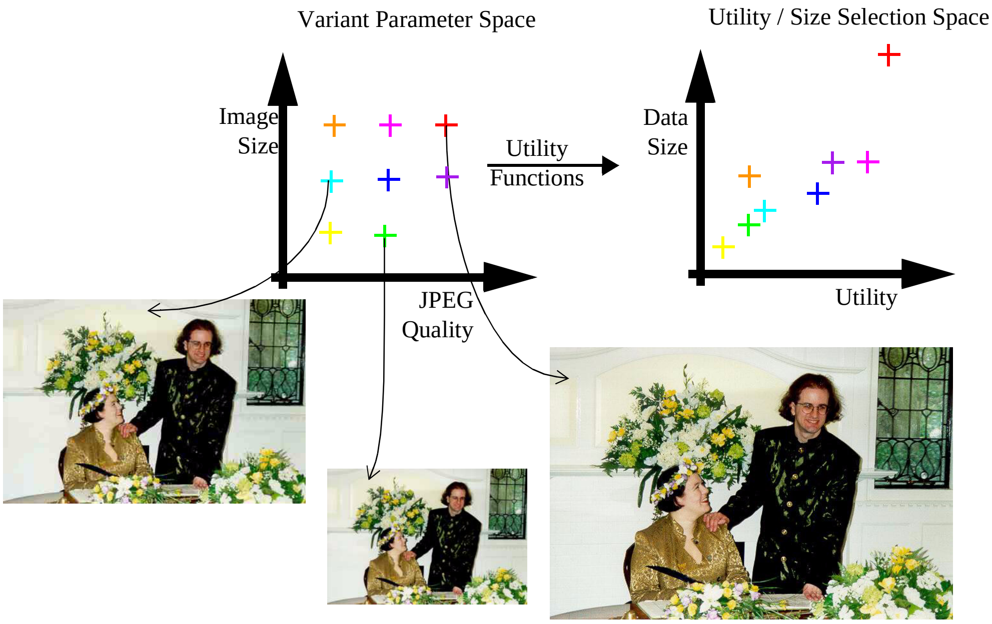


Figure 10-4: Image Selection Space, Photo Type Example

When defining the utility functions for images it helps to know something about the data which they convey. Photos can be scaled to a smaller size quite aggressively before utility drops to an unacceptable level. Where the detail in the image is important, such as for images containing text, e.g. buttons, or raster maps, then the loss of utility will be much faster [13]. Our system of specifications encodes this need by defining utility functions on parameters to be applied to variants of different type elements. The element type can encode whether the data is a photo, a line drawing, a navigation button etc.

When image size (pixels) is changed to reflect screen size then the size ratios between the various images in a document should, at least to some degree, be maintained. While there may be little data reduction achieved by reducing buttons and icons the proportion of the screen which they should occupy should be reduced to maintain the appearance of the document. This is not best achieved through the screen space resource as this considers the whole of the document. A page with a single main photo and several icons may meet the screen density limit by reducing the size of the main photo, depending on the weights assigned and the selection algorithm used. The utility functions should again be used to limit the largest acceptable size according to the type of the element they represent. Main images may have a larger maximum size than icons, reflecting the utility the user derives from each at the various sizes.

10.2.3 Text Adaptation

There has been substantial work on adaptation of text, particularly in the information retrieval community. Processes for summarising text, extracting keywords, splitting text into sections, translating between languages, translating between encodings (such as HTML, word and PDF) or converting text to speech are not our focus of research. However, the existence of techniques to provide these facilities underpins mediation of web pages. Differing representations of pages which can be generated automatically are vital to the approach. Page authors should not be burdened with producing multiple versions of content. Automatic processing of images allows a high quality original to be processed in various ways to meet limits in pixel dimensions, data size, colour depth etc. Text is also amenable to translation which will affect parameters at the element level (splitting into sections) and at the variant level (translation, representation in different encodings, summarisation etc.).

10.3 Mediating Hyperlinks

In the map application we considered the problem that there may be too much data which is somewhat appropriate to a location to display. In the case of web pages we develop this problem to consider the case of there being too many hyperlinks from the data displayed to allow effective use. This may be a significant problem on devices with small screens or situations where selecting specific points or regions can only be achieved in a rather clumsy way, e.g. with a touch screen.

There are many examples of other work in spatial hypertext, including map applications with hyperlinking, e.g. [29,36]. We assume a model where the data linked from and to and the data describing the links are separate, as in work at Southampton [100]. Links related to an area and possibly other factors can be provided on request. This may be by performing a query on a link-base, or by having the link-base return links given a view of the data being linked from. When an area of the screen is selected one could either follow the “best” link, or offer a limited choice.

It seems that there is potential for selection here. There are various factors which may be used in deciding which links to offer for a given selection: the element linked from, the area of screen the link covers, the element or document linked to, the source of the link, the number of search criteria the link satisfies and the ranking of the link by the link-base or search engine are all factors which seem reasonable. The selection for each area of the screen (word, area of map, area of image, etc.) could be bounded by the number of links that can be listed at one time on the screen and only those with the highest weights and/or utilities shown.

10.4 Related Work

In earlier work we implemented some adaptation of web pages for mobile systems [30]. This work represents an early version of our mediation techniques, without much of the context dependency and refinements presented here. However, it does demonstrate using similar techniques to perform mediation of web pages towards a download deadline.

The web adaptation system presented by Nagao et. al. in [90] offers many of the features we describe for web page adaptation. The transcoding is performed according to the semantics and preferences in their system. In ours the selection is made from offered variants, which may be the result of transcoding. The most appropriate variant is selected according to specifications which take into account the semantics. Our approach does not preclude the transcoder being aware of the semantics of the data, indeed for text transformation this is almost essential. It is also possible to advise the transcoder of preferences when requesting variants it can offer, although this is not required in our approach.

Examples of web pages adapted to meet download deadlines in the face of limited throughput and with images limited to fit within a screen size are given by Mohan et. al. in [88]. Their approach to adaptation is somewhat different to ours, as described earlier, although similar in basis.

Fox et al with their “Top Gun Wingman” system [52,54] demonstrated the benefits of adapting images and text formats for wired and wireless modem connected systems with small display systems, particularly Palm PDAs. However while demonstrating that transcoding or distilling data on-the-fly has great benefits they do not extent their implementation to one which reflects the diversity of devices and needs. Neither do they claim to do more than speed things up or make data formats more appropriate to the device, such as consider how page elements combine to form a web page with a total screen space or download time and use this to meet goals. Another proxy based approach to restructuring web pages, in this case to address screen size limits is described in [61] (Hori et. al.).

10.5 Summary

In summary, there is potential for mediation of web pages, indeed work such as [53,65,67,88] points to the benefits of adapting web pages to the network and device. Our approach to mediation supports sensitivity to other aspects of context and provides a framework in which web pages may be adapted in a general manner rather than for a small range of specific situations.

The work on the semantic web and the inherent range of elements between and within web documents supports the approach of weights to reflect preferences for semantic types of elements. Work on adaptation of text, images and other media offers great potential for alternative representations which differ in many ways. The need for utility functions to express preference among a complex set of choices becomes greater as the available representations of elements becomes wider. The issues of screen space, download time, computational load etc. remain significant in web pages. While some web pages may be designed for low-impact on simple devices the general trend is for ever more complex pages and richer media. An automated way of scaling this richness just enough to meet the needs of the user and operate within the limits of the context is of widespread benefit – not just to users of the most limited devices.

CHAPTER 11 CRITICAL EVALUATION

In this chapter we shall briefly examine our success in meeting our aims. Shortcomings which have emerged during development and testing shall also be discussed. We shall discuss the map mediation application as a whole initially, before drawing out some specific issues.

11.1 MAP MEDIATION APPLICATION

We stated that our proof of concept application should implement sufficient power in context awareness, resource management, specification and execution of mediation. It should also be portable in code and user interface between platforms. We have achieved these aims, to varying degrees and arrived at a useful proof of concept from which substantial testing and illustrations throughout this thesis have been taken.

The context awareness implemented is somewhat limited. Context states, as defined by aspects and the values they may take, are defined in a simple configuration file and presented on a user interface. There is no automatic sensing of context state. The only exception to this is the location service (discussed further below), which is not used as input to specification selection. Greater automation in this area would be desirable before embarking on substantial user tests but is outside our current area of research, often being hardware and signal processing based. The management of dynamic resources has received rather more attention and is discussed below.

The user interface to the map application was designed with portability in mind, a screen shot is shown in figure 11-1. The map display is on one window, with only the map display and a simple set of mode controls shown. Controls are designed to be operated by selection with no left/right or double clicking, which are often impractical with mobile operation of devices or on touch screens. A progress bar provides the user with feedback on the proportion of any deadline taken and the phase of data loading the application is in (loading meta data, selecting, loading data, drawing or done). The remaining controls and feedback from resource managers etc. were collected into another window, with a tabbed pane for each control function. Functions included: the location service, feedback on network parameters modelled, context state selection and test harness functions such as network simulation controls.

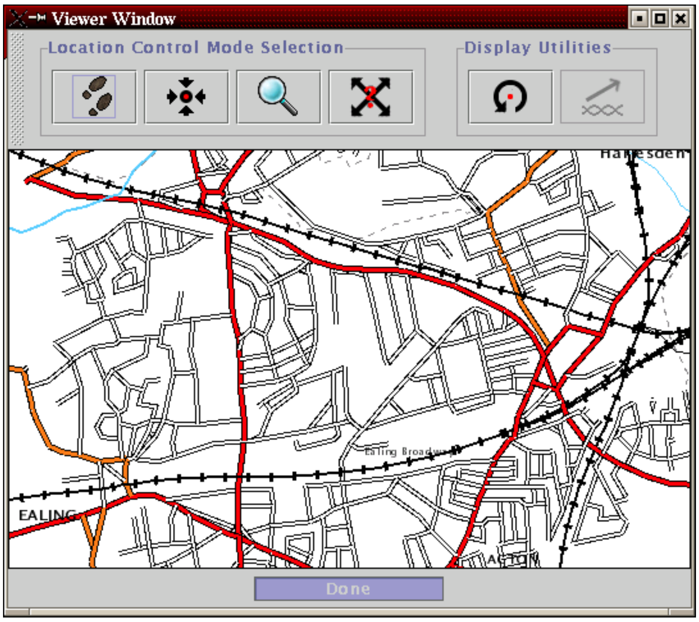


Figure 11-1: Map Application User Interface

The other major user interface component was the map rendering. The drawing engine was developed by us as there were no freely available drawing tools with the API features we required supporting data formats which were available to us. As the drawing was not the focus of our work this was not pursued as a priority. In general the rendering is sufficient (see screen resource management below for one extension which we would make in future work), although the rendering is not perfect. The map data is designed for tailored drawing and the placement of data is set up for printing on paper. There is no explicit shape model within the data to simplify tasks such as filling areas. Assumptions based on the presence of area seeds are required. Where the data is presented in part seeds, boundaries or spaces within an area may not be available. As there is no connection between these features in data we did not manage to ensure that both seeds and boundaries were collected into an anonymous element group. In consequence some conservative estimation and reliance on API delivered drawing functions was made, which means some area fills are poorly rendered. Another problem area we found was that text layout was defined with specific font sizes in some NTF variants. This made modification of rendering difficult. In further work a model based on ESRI Shape or GML map encoding formats may be preferable. These were both briefly tested during implementation. Unfortunately breaking up shape data in the server and providing the required application control in the client proved impossible to find for ESRI. GML only emerged late in our work and suffered from two prohibitive problems at that stage: First, the DOM based parser could not run any practical volume of map data on the desktop systems available to us due to a lack of memory. This would make development hard and any claims of portability to mobile devices impossible. Secondly, the coverage of the UK available to us in GML was limited. With more time and better data availability a SAX based GML implementation would have improved various aspects of the data processing and rendering.

In terms of portability the application is somewhat limited. It is written in Java which offers ease and range of portability in terms of platforms. The user interface is designed for use on limited capability devices. Satisfactory tests have been performed on 350MHz Pentium-II class machines with 128MB RAM running Linux and X. The application, in its test configuration, takes approximately 80-100MB RAM with a 120MB maximum heap set, having loaded several hundred maps. This memory load is inappropriate for many of the intended devices. There are three approaches which could be taken to rectify this. Firstly, there is code in its test configuration which is unnecessary for routine use. Components for writing logs, drawing graphs of meta data, simulating slow networks etc. come under this category. Secondly, the data representations chosen were often for ease or clarity of programming rather than efficiency. Refactoring code, in particular meta data representations and selection algorithms may yield run-time memory use improvements. Lastly, in the case of the most limited devices, there has always been an intention that components could be deployed on client-side proxies. The core mediation process and resource management could be moved, with context models, specifications and resource monitoring retained at the client for privacy and effectiveness. This approach would be reasonable, as long as the volume of context change data which needs to be transmitted to the proxy is not too high and may deliver a reduction in overheads due to meta data loading. Another candidate for migration to a proxy would be the map rendering engine. If the data were delivered to a proxy, rendered and a raster delivered to the client the overheads of map drawing and vector data management could be migrated. We did not pursue this approach for reasons of clarity in our management (there would be many more throughput and processor load variables to consider and control) and simplicity in design for an initial implementation. Again, this approach may deliver an improvement in data which can be loaded where there is a slow last hop. An improvement in selection and/or rendering speed, where the client has limited CPU power, may also be achieved with these techniques.

11.1.1 Specifications for Contextual Mediation

Our system of specifications to contextual mediation was possibly the clearest success of our work. Writing specifications requires insight into the way the data is used. Testing specifications requires careful and critical analysis of the data presented and is an iterative process. Preferences which may come from initial assumptions can be proved wrong when navigation is attempted using the resulting data, for instance in initial testing the use of information about railways for navigating in a car was substantially underestimated. As discussed earlier in the thesis, the specifications are unlikely to be written by the casual user. Specifications in real use will be the product of specialist knowledge and substantial testing efforts. Our encoding of specifications, in XML (see appendix 2.3), provides for the interested user to modify settings to their own needs if they wish.

Some degrees of freedom in treating data were found to be a problem in testing. Initial specifications included a preference for element groups with higher numbers of contained elements and elements with larger areas. These resulted in distorted selections, even where more information was apparently delivered. Similarly, when weight functions based on distance from the point of interest were first introduced these were defined as linear functions. This resulted in most elements of the same type having a different weight and being treated individually. The result was that gaps in the map were scattered throughout the display. We found that by using step functions the number of element weights assigned was reduced and more elements were considered together as they were collected into concentric circles around the point of interest. Consequently it was clearer what data was present and what was not, particularly in the case of dense data such as roads.

This highlights an area which needs greater work both in selection and rendering of data. In selection there are still occasional cases where features are noticeable by their omission. It would be interesting to experiment with indications of areas where data mediation had resulted in omission. This could serve as a warning to users that actions other than mediation may be necessary – such as zooming or modification of goals.

11.1.2 Resource Models

A substantial part of this thesis and the implementation has been concerned with modelling dynamic resources. We have developed resource models for: network throughput, network round trip time, screen space and cost. Our model of network throughput is not entirely accurate, but is generally sufficient. Causes of inaccuracy include: variations in compression due to different volumes of data and inaccuracies in estimating time taken to transmit data received in first data delivered. This is particularly noticed under changes of throughput and round trip time and where throughput is high or initial data delivery is large. Loss of ability to meet deadlines is limited to very high round trip time connections and even then the application outperforms the unmediated control. The generality offered by application level monitoring is an advantage and our model has no impact on the network performance, requires no operating system co-operation, has a low memory usage and little CPU overhead. Other resources which present themselves as having some merit for management include battery life, memory and drawing time. We have not had time to implement these. The literature has considered the impact of applications and wireless communication on power drain and a model which mediates data requested against available power would behave in many ways like the cost model implemented. A model of how data causes battery drain would require longer term monitoring and metrics than we have used in other resources. Memory use and drawing time (which is CPU / system bus intensive) would be similar to throughput reservation, assuming some largely linear model exists for the data's impact.

The tuning of the network model took some time to arrive at and required some assumptions about likely degrees of change. The tuning described will react to quite small changes and in practical use the sensitivity to change (causing jumps rather than moving average updates) and stability of the moving average may benefit from being less sensitive. Of all the areas of development the network monitoring required most changes in response to early test runs. Problems included the substantial difference in performance of meta and map data under compression, the difference in round trip time for meta and map data from the test server, inaccurate assumptions about the behaviour of Java's HTTP streams and errors and poor assumptions in the measurement and modelling code. Additional features such as interface to lower level QoS mechanism like RSVP was not implemented as this was not expected to give great benefit over the wireless networks which were our main consideration and deployment in the internet is relatively poor.

The screen space resource is still in early stages of development, but has proved successful so far. The impact of different feature types requires more work to improve the accuracy of metrics. Context sensitive and mediation controlled map drawing would also be a great benefit here, e.g. to modify the rendering of less important elements where these could still be shown. These improvements would also require work to the drawing engine.

11.1.3 Meta Data

The meta data was designed from a pragmatic point of view, using XML to facilitate parsing and incorporating ideas from the literature. The data model underwent one major revision early in the development, providing a greater separation between elements and variants and more clearly separating types from element identities. At this point XML was adopted rather than the previous ad-hoc language. Standards, such as RDF and Dublin Core, either did not provide facilities we deemed important (such as the element / variant structure and typing in addition to syntactic properties) or were not fully defined at the point we needed to make design decisions. We designed the language with the intention of simplicity and extensibility, with application specific needs designed as extensions rather than requirements. Indeed, we modified the language to include a cost for variants when working on our cost resource model. This required little effort and changes to the parser and meta data classes were could be performed independently. Further work on the implementation may include migration to standard encoding.

11.1.4 Location Service

We developed a simple location service, as part of the application. This followed the prior art [80,81] quite closely and is described in appendix 1. This was generally sufficient, although the used location services were mostly the application, location sequence files and a GUI chooser for named locations. A GPS input was part of early work but was not maintained as the serial API in Java requires platform specific compilation. As it stands the service works well, although the mechanism for cancelling inaccurate locations from sources is not as transparent as one might like.

Development of the location service in two areas would greatly facilitate the further development of the map mediation application. The first of these is the support for sub locations, to represent needs such as planned routes. A bounding location of interest which could contain a series of expected locations would provide a more realistic expression of interest in a location than the model we have used, which assumes the centre is the most interesting point in the map. Mediation to reflect this would offer improved usability. The second improvement would be the inclusion of location prediction. Again, this would result in a more interesting demonstration / trial application as data could be loaded in advance. A prediction of *when* the user would be in a location could also provide interesting input to the goals of mediation.

11.1.5 Map and Meta Server

The map data server we developed was intended only to aid testing. It suffers from a long start-up delay and high memory requirement as the data model is held in memory rather than a persistent database. The grouping of elements is also defined in the server configuration. We used groups of 700m squares with 100m overlap on each side and 4km squares with 1km overlap on each side. In future it might be useful to vary these groupings dynamically, which would require a spatial database of individual features. Variations might be to reflect granularity in specifications and also to reduce meta data volume for large requests or areas of great detail. The processing of data in the server to produce reduced detail variants follows a naive algorithm. We have since identified improvements which could be made in this respect. In general, however, the server performed its test system role well. It introduced little overhead and runs for months at a time. The grouping and variants of data offered have proved sufficient for a proof of concept. Any improvements in future work will not fundamentally change our results.

11.2 SUMMARY

The application developed remains a proof of concept. Substantial user testing with the application as it stands would not be practical as there are issues which would stand in the way of testing the principles of the mediation process, resource management or other functions under test. There are areas where the code needs attention before it is suitable for general consumption. Experimentation in various areas would lead to improvements in system performance and usability. Having said that, the application developed has provided a good test bed for ideas. None of its limitations prevents use for demonstration and testing. Our context and data representation and mediation specification and processes were all refined through application to map data.

CHAPTER 12 CONCLUSIONS

12.1 RECAPITULATION

12.1.1 Contextual Mediation

We have addressed a part of the problem of context awareness for ubiquitous computing. In the process of defining our area of interest we have arrived at some definitions. Firstly, for context:

> *Context is the circumstances relevant to the interaction between a user and their computing environment.*

We then went on to define context awareness:

> *A system is context-aware if it uses contextual information to provide relevant information and/or services to the user, or to enhance the provision of services.*

Context is clearly a factor with substantial impact when considering ubiquitous use of computing devices. This is a wide ranging research problem which has received much attention. Context awareness can manifest itself in a variety of ways. We are interested in a sub-set of the possible ways in which a system may be context aware – *contextual mediation*, which we define as:

> *Using context to modify a service. For instance to describe limits and preferences over a large range of offered data, in order to display the most appropriate parts. The request for the data being mediated need not arise from the context.*

The process of contextual mediation should provide the best rendering of the most appropriate information within the constraints of the context. This is a different prospect from providing as much information as possible at any time, or of making maximum use of resources at any time. The most appropriate presentation of information may be to do nothing. Providing the essential information clearly and without distraction will be more powerful than providing all the data offered, which can be expected to include much unnecessary information.

12.1.2 Aims

Chapter 2 included a review of the literature on problems and support techniques related to mobile and ubiquitous computing. We arrived at a list of techniques which required attention in order to better provide for systems which would adapt to a wide range of context through mediation in order to remain usable. In this thesis we have addressed the following parts of this work:

- A model of context, as a set of discrete aspects, which includes a model of dynamic resources, which may be reserved. The resources modelled were:
  - Network throughput and round-trip time.
  - Screen size and drawing density.
- Meta data to describe potential data to be used. This allows for data structure in both semantic and syntactic relationships and a description of parameters in each of these.
- A mechanism to specify selection over structured data, where elements of data may be available in many variants. This mechanism reflects preferences due to context, in terms of the different semantic types of element and parameters of variants as described in the meta data. This process we call contextual mediation.

A proof of concept application was developed to investigate these ideas. This provided contextual mediation for map data, which proved successful in tests against a wide range of simulated networks, a basic usability study and in subjective evaluation. This proof of concept has been the basis for much of the discussion through the thesis. Application of the techniques to web pages has also been described.

We shall now review our contribution and the success of our approach in the areas highlighted above. We shall then discuss the wider application of our work and future developments which may build on it.

12.1.3 Specifications and Selection

A new and highly flexible mechanism to specify selection over structured data, where elements of data may be available in many variants was presented in chapter 6. This mechanism reflects preferences due to context through a combination of:

- Weights, which treat preference due to different semantics of document elements.
- Utility functions, which treat preference due to parameters of data encoding and limits on data imposed by the context of use.
- Goals, which reflect limits due to usability in the context of use.

According to the context, different semantic types of elements are treated differently. This is achieved through weight functions due to type and other parameters of elements, e.g. one may prefer to see road data to river data. The treatment of the semantics of the data offers a granularity of mediation and response to the users needs which earlier approaches such as uniform scaling or limiting of data could not achieve.

A description of preference and usability limits for various parameters of variants is provided through the use of utility functions. Utility functions are selected according to the context and so are able to reflect the needs of the user, e.g. appropriate map scale varies according to the task at hand. The use of type in selecting utility functions may reflect a preference for high detail in road data but less need for detail in building outlines. Utility functions may also reflect the context of use, e.g. describing appropriate sizes for an image in the context of the screen in use.

A mechanism for defining limits over managed resources according to context is provided using goals, for instance download deadline may vary with speed of travel. The implementation of resource management and selection means that these limits are only “best-effort”, but have been shown experimentally to offer substantially more predictable behaviour than the unmanaged response of requesting all data. Limits due to screen space can operate over combinations of elements, causing a document presentation to fit within a screen. This contrasts to utility functions which can only treat single elements at a time. Our combination of these techniques allows mediation to treat data presented as a whole (formed from a combination of elements), rather than the series of disjointed decisions which typifies approaches in much of the literature.

Combinations of these descriptions of preference and need are selected according to context. Matching functions over context aspects allows preferences which reflect a narrow set of needs to be defined and selected separately to those reflecting other needs. As the detected aspects of context and needs due to context evolve, the specifications can be managed in a modular fashion.

So, specifications in our system consist of four parts:

```
Spec: ([c-match], [select-wf], [select-uf], [goal])
```

a. A context match, which triggers the specification as active, e.g.:

   ```
   c-match(speed, c-speed) = (30 <= c-speed <= 100)
   ```

   The weights, utility functions and goals in specifications reflect the needs of the context aspects which trigger them.

b. Weight functions are selected according to element type, which are either constant values or functions of an element parameter:

   ```
   select-wf(<null>, road) = 0.6
   select-wf(distance, road) = fn_nearer-is-better(distance.value)
   ```

where

```
select-wf: (Attribute, Type) →
  (named-weight-function: Attribute.value-type → 0..1)
```

When multiple weight functions on a parameter are selected for an element, a zero (positive disinterest) is honoured otherwise the highest weight is used. Weight functions of different parameters are combined by multiplication.

c. Utility functions are selected according to element type, which are functions of a variant parameter:

```
select-uf(scale, map, road) = medium-detail(scale.value)
```

where

```
select-uf: (Attribute, Format, Type) →
  (named-utility-function: Attribute.value-type → 0..1)
```

When multiple utility functions on a parameter are selected for a variant the values are combined by multiplication.

d. Goal functions are either constant values or functions of some metric over the current selection:

```
goal(deadline) = 15
```

where

```
goal: Resource → Value
```

or, taking a parameter from the selection made:

```
goal(deadline) = (10 + 10*max-weight)
```

where

```
goal: Resource → function(selection-parameter)
```

When multiple goals on a resource are selected for a document the most restrictive is applied.

These context selected specifications enable a wide range of differentiation in semantically rich applications with well structured data. Several specifications may satisfy their context matches at one time, reflecting different independent aspects of context. A mechanism was described to combine these specifications. We allow specifications to form hierarchies. Specialised specifications combine in a similar way to class inheritance. A specification specialises another specification if:

- It performs matches on all the context aspects of the more general one; and
- All those matches are equal or within those in the general specification; and
- The specialised specification has at least one match which is within one from the general specification, or over a context aspect not used in the general specification.

A weight-, utility- or goal function selector for a type / element parameter, type / variant parameter or resource / selection parameter (respectively) which exists in the specialised specification overrides the general one. For example, preferences due to being in a vehicle can describe the general case. Preferences due to driving or being a passenger or driving a lorry can develop from these simply describing the additional or different needs. Where no such overriding takes place the general specification's weight-, utility- or goal function selector is used. Combination of remaining multiple weights for a type etc., are as described above.

Algorithms for taking these specifications and performing the mediation of data were discussed in chapter 7. The selection is an optimisation problem, seeking to optimise the user value in the data. A degradation path is formed which takes a view on the impact of variant selection on the user's experience of the document. Element weight and variant utility are used to create an order in which variants of elements are selected. Once all resource constraints are satisfied an improvement phase occurs, which treats the taken steps on the degradation path in reverse, improving the selected data where this will not break resource limits.

12.1.4 Context and Resource Models

A description of context which includes a model of dynamic resources, which may be reserved was given in chapter 5. Resource management has been implemented and tested for the network characteristics of round-trip latency and throughput. Results of tests of the network monitoring were presented in chapter 9. Management of screen space and data density on screen and of money as a resource against data cost were also developed. Initial results for the screen density management are given in chapter 9 and illustrations of the technique provided in chapter 8 for maps and chapter 10 for web pages.

The network model provides an exponential moving average model of throughput and round trip time at the application level. Results are aggregated over multiple flows. A recent history is also maintained and can be used to reset the model when substantial changes are detected. The combination of techniques ensures a rapid response to significant (factor of two or greater) changes while maintaining stability in the face of minor fluctuations. The modelled value is adjusted (to within measurement noise) within four loads and has made over 50% of its adjustment within three loads. The combination of agility of response and noise immunity is rarely found in the literature – [72] is a recent paper by Kim and Noble describing a similar scheme to ours. Our technique also has the benefit of being independent of the network protocol stack, aiding portability between systems and between network protocols and having no network overhead. The model allows the mediated map application to meet deadlines through mediation in over 50% of cases for a wide and practical range of simulated bandwidths, round trip times and deadlines, even with a high and bursty simulated loss pattern. In most cases the ability to meet deadlines is substantially better than this and 90th percentile download times are within 5 seconds of the deadline (tested for deadlines of 20, 30 and 45 seconds) in all but one test case with a sub two second round trip time, between 1 and 256kB/s bandwidth and loss up to 5% with 20% subsequent loss. In summary the mediation system coupled with our network model offers the user substantially more predictable behaviour than an unmediated system, for a modest overhead. The range of network characteristics which have been demonstrated to be practical includes and goes beyond those of commonly available wireless systems. The tolerance of variation in characteristics, as found in lossy networks, is also good.

The screen density model is harder to test formally, but initial results and examples indicate that it provides a benefit in mediating data for small screen devices and/or situations where the density of offered data for an area is high. Managing data density on the screen provides a useful alternative to zooming in that it supports elimination of confusing detail while maintaining a wider view. Zooming remains an appropriate technique to apply when the detail is required.

12.1.5 Meta Data

Meta data to describe potential data to be used was described in chapter 4. Meta data was used to describe map data for requested areas in the application developed. This description included:

- A description of data element structure, so that a highly structured data may be presented for negotiation as one request. In the map application elements correspond to map features. The element structure indicates semantic dependency, e.g. road label only useful in context of road described.
- A description of multiple data variants which realise each element and also describe the structure of syntactic dependency between data, e.g. variants containing map data which realises elements depends on header data (header data does not itself realise any element).
- A description of data properties, to enable resource management and allow selection to reflect user preferences. Properties may be on elements, such as location, or more commonly on variants, such as data size and survey scale.
- A description of data semantics, to enable differential treatment of data elements and further allow selection to reflect user preferences. This is provided by typing elements according to a published type definition.
- Common language(s) for structural, properties and semantic type descriptions, which may be extensible to manage developing application and data classes. This was achieved through

XML encoding of meta data with an extensible set of tags to identify properties and a domain description for XML encoded type definitions. Mechanical mapping between NTF feature types and our type system was also developed.

Our results show that while there is an delay associated with the meta data, particularly where round trip times are high, it is not excessive (around 10 seconds at 2kB/s for 12 to 33km^2 suburban and urban areas). Much of this delay could be mitigated if the mediation were delegated to a proxy in the cases where the overhead was considered too high.

The meta data developed is unusual in its combined support for:

- A description of semantically structured data.
- Data described as a multi-element document rather than describing data per element.
- Description of variants realising these elements and also having their own syntactic structure.
- Support for different variants of elements.
- Description of variant parameters, rather than qualitative judgements.
- The use of a separate semantic type system to aid portability between application domains.

12.1.6 Case Studies and Implementation

The application of our specifications, selection techniques and resource management tools were illustrated in a case study for map adaptation (the main theme of the examples throughout) in chapter 8, drawing from the implemented system. A basic user study based on this system was performed. The tests of the range of applicability of mediation over different networks for the data set available and a user study were presented in chapter 9. A critical evaluation of the exemplar was given in chapter 11. An example of the application of contextual mediation to web pages was given in chapter 10, illustrating that the techniques are not restricted to the map application domain.

We note from the literature the benefits of timely delivery and limiting density of data on the screen – and our results using basic map data and case studies support this. As the volume and diversity of data available increases these benefits increase. Where maps are simple and sparse the mediation provides little apparent benefit, but imposes only a modest overhead. The map produced will only be better, in the user's judgement, where removal of information is of benefit or the impact of excessive resource use (slow loading, cluttered screen etc.) is greater than the benefit of the information from the extra data loaded. The power of contextual mediation lies in both what is and what is not shown.

We have demonstrated that the techniques described provide significant benefits, even in the current limited form of the prototype application and simple case studies described. The benefits we have demonstrated arise from: use of map data which corresponds to detailed printed maps, screens of PDA to laptop class, network qualities from mobile phone standard up and common tasks such as driving a car. As the range of data available and the ubiquity of use both increase the need and power of the mediation techniques described will become more apparent. The addition of further resource management and the rising use of highly structured data, rich meta data and dynamic adaptation of data further support the mediation process.

12.2 Applicability to Other Application Domains

We have presented in our case studies two example application domains. In general the techniques described are most appropriate to document style applications, with a retrieval / use / request next semantics. Applications which offer highly structured data, such as maps and multimedia documents offer the greatest variety of outcomes from mediation and are therefore most interesting to study. Event-based applications, such as sports result notification services, where events in many individual games or contests may be described in many ways could also benefit. The specification techniques would also be applicable to multimedia applications with flow semantics, however the monitoring and selection techniques would need substantial modification (and much work has been undertaken in the area of multimedia flow QoS management and adapting streaming media).

The variety of applications supported by the application domains already discussed suggest that the technique is quite general – much data is represented in the form of web pages today and so this forms a wide application domain. However, this breadth of subject matter would require significant work on developing type systems. There is research interest in ontologies and the "semantic web" which would provide much of this. The essential needs would be:

a. A general type system as a fall-back, which could be applied in a majority of cases. This need only encompass simple types. In a web page this may be icon, logo, photo, drawing, body text, menu text etc. Much of the time specifications could be developed towards this type system and result in reasonable contextual mediation.

b. A wide range of more detailed type systems describing specific domains of interest, capturing the classifications in different organisations and domains of interest. More sophisticated specifications could be used to reflect those domains of interest which are used frequently.

c. A means of translating between different versions of the above ontologies. This is required to support the almost certain overlaps and divisions between groups in any area of information provision, or collaboration between fields. Separate specifications to reflect essentially similar concepts should not need to be developed.

This work on ontologies and the semantic web is not our main concern, although its crucial to further development of our ideas in the web at large.

12.3 APPLICABILITY TO UBIQUITOUS COMPUTING

Many of the examples presented are described in terms of personal mobile computing devices: PDAs, laptops etc. Ubiquitous computing is wider than this and our work is intended to support a diverse range of contexts of use – indeed this is its main strength. We would expect the technique to be applicable on aeroplane seat-back computers, car dashboard systems, kiosks on the street, shopping centre or other public buildings and other emerging types and locations of devices.

The network resource management presented demonstrates modelling of a wide range of networks and the provision of a useful response even in very limited circumstances. The screen density management has been illustrated on a variety of screen sizes with maximum densities reflecting different conditions of use. Power, CPU time and memory management could all be expected to operate over a range of conditions. These resource management functions abstract away from specific contexts and simply describe the limitations at hand. The various devices which we would hope to apply contextual mediation on will, generally, suffer from limitations in one or more of these resources. Not *all* resources managed have to limit each selection made. The generality resource management offers supports diverse situations, we have simply presented specific examples in common situations with known parameters to ease understanding.

Our specifications are designed to be able to encode limitations and preferences at user and system level. Preferences for semantics will reflect those aspects of context which are facets of the user (e.g. task). Preferences for media quality will arise from a combination of the user’s perception of quality and the limitations of the device. More fundamental properties of media variants will be subject to the user’s needs (audio vs. text, English vs. German) and the system’s limits, e.g. support for specific encodings. For example, the needs which will arise from use of a kiosk in a public place will have similarities to: those reflecting other devices with that type of display (e.g. laptop); having a touch screen (e.g. PDA); working under strong lighting (e.g. use outside); lack of privacy (e.g. use in public) etc. The location of the kiosk may influence the likely tasks to which it is to be applied (e.g. navigation round a shopping centre). It can be seen from this example that an individual context of use will often be able to reuse other specifications reflecting those aspects of context which have already been met elsewhere. Expanding the technique to support new forms of ubiquity need not involve great effort.

12.4 FUTURE WORK

There is clearly work to be undertaken to develop these ideas further. Tasks which have been identified during the writing of this thesis include:

- Further examination of the application of these techniques in a more distributed fashion. We outlined in the thesis our reasons for a client-centric deployment in our initial study. However, the balance of needs may well shift. The system was developed with a view to using interchangeable components and there may well be benefit in deploying some components away from the client, particularly where the client is "thin". Development of alternatives would enable us to better understand the trade-offs. Examples include:
    - Map drawing may occur on a proxy which delivers images. The benefits of on-demand rendering of vector data would then be combined with the lower data transfer costs over a slow link to the client (where the vector data is larger than the image). This also has benefits for CPU or memory limited clients, for whom the vector map drawing may be a problem.
    - Where the last-hop to the client is slow there will be benefits in moving the selection system off the client. The overheads associated with loading meta data and the somewhat memory intensive selection process would then be reduced. This would probably be to a proxy near the client. There are disadvantages related to maintaining accurate and private context and resource models.
    - Where the device is a public device used in passing there may be specifications provided by default by the owner of the device. The appropriateness of overriding these by loading a user's specification from a remote site should be examined. In many cases a combination of the two sets of specifications will be expected to lead to the best result.
- Expand context sensing and description to better automate the input to the specification system. At the same time user interfaces for informing and overriding the specifications derived from context would be desired by many users. These could aid in providing feedback

to adapt specifications to reflect developing user needs. Extending the context awareness of the application should include:

· Studying the interaction between mediation, screen manager and context aware rendering.

· Porting and evaluating the application on a variety of actual devices and networks rather than simply simulations. Tests of the application on PDAs, laptops and kiosks in various environments would be essential in verifying that specifications designed for a context actually provide improved behaviour. This could be coupled with better resource and context sensing. This may build on others work in the areas of power management, vibration and movement detection and sensing of ambient light and/or noise.

· To further develop the location model used. Rather than providing a single location of interest it ought to provide a location of interest which contains other locations. This can provide for paths which one expects to take, start and end points etc.

· Provision for enabling specifications where certainty in an aspect's value is less than 100% and resolving conflicts which may arise from this in the specifications activated.

- The use of goals to order data loading, rather than as a single limit may be interesting to explore. For instance rather than apply a single deadline multiple loads, each within a deadline, may be used to allow prioritised incremental loading of data while providing an deadline on the initial response.
- Modifications to rendering to reflect weights may be interesting. In maps the most important symbols may be highlighted. In web pages text elements may use larger fonts or different ordering to reflect their weights. This will require that screen density management and rendering interact. The HCI issues which arise from this may also be interesting to explore.
- Work is also required on human interface issues and the development of new application domains, including:

· To review selection techniques for ensuring consistent presentation. Both between related features in a document and between successive documents.

· To expand the application to mediate non-rendered data such as hyperlinks. To develop the web mediation and possibly integrate with the map application. Hyperlinks may exist both from the map to web pages and from web pages to the map. The hyperlinks which are active may be mediated to reflect context rather than having a cacophony of irrelevant links. An integrated map / web browser would also lead to the need to consider behaviour for competing applications.

· To continue the work with maps, improving rendering and HCI of the application, expanding data sets, refining specifications and extending the usability study. This would include work on both the map server and the client.

12.5 CLOSING REMARKS

We have described and demonstrated a set of techniques to enable *contextual mediation* to support users of semantically rich data intensive applications on a wide range of devices, using a wide range of networks and engaged in a wide range of tasks. We believe that these techniques offer a general mechanism to support many applications and contexts, while offering a more powerful and sensitive adaptation to context than those in the prior work. In particular our approach is a general one which can make use of specific adaptations without relying on them.

Data forming *documents* is described in meta data, which describes semantics, syntactic properties and structure of data. The selection space formed by structured documents is treated through *weight functions* over *semantic type* and other parameters of abstract media *elements* and *utility functions* over parameters of *variants* which realise elements. The mediation is subject by limits from *goal functions* describing limits on managed resources.

We have described mediation limited by a download deadlines considered by a network throughput and round trip time model. The model offered agile adaptation to substantial changes, even when subject to severe and bursty packet loss. The model was sufficiently accurate that mediation met deadlines in most cases, thus providing a predictable user experience.

We also implemented a screen density model, which aimed to limit clutter on small or otherwise hard to read screens. User tests based on our prototype application and basic map data indicated that the data presented as a result of mediation was useful and generally at least as good as that presented without mediation.

Our exemplar application, of mediated maps offers a full demonstration of the principles described. Application to web browsers and ubiquitous computing in a wider sense has also been considered and is believed to be practical and beneficial.

Although work remains the techniques described have been demonstrated to be beneficial through our initial work. The user's experience of applications in a wide range of contexts can be improved through contextual mediation which reflects the diverse needs and limits of the context by making use of the rich meta data and structure of data.

BIBLIOGRAPHY

1. Abdelsalam, W.A.E.F. “Maintaining Quality of Service for Adaptive Mobile Map Clients” *MSc Thesis* University of Waterloo, Ontario, Canada, 2001
2. Abdelzaher, T.F., and Bhatti, N. “Web Server QoS Management by Adaptive Content Delivery” *Proceedings of International Workshop on Quality of Service (IWQoS’99)* pp.216-225 London, UK, May 1999 (IEEE)
3. Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. “Cyberguide: A Mobile Context-Aware Tour Guide” *ACM Wireless Networks* vol.3, pp.421-433, September 1996 (Baltzer)
4. Abowd, G.D., and Mynatt, E.D., “Charting Past, Present, and Future Research in Ubiquitous Computing” *ACM Transactions Computer-Human Interaction* vol.7, no.1, pp.29-58 March 2000
5. Agrawala, M., Stolte, C. “Rendering Effective Route Maps: Improving Usability Through Generalization” *Proceedings of ACM SIGGRAPH 2001* Los Angeles, CA, USA August 2001
6. Alfano, M., and Sigle, R. “Controlling QoS in a Collaborative Multimedia Environment” *Proceedings of 5th IEEE International Symposium on High Performance Distributed Computing (CHPDC-5)* Syracuse, NY, USA August 1996
7. Aurrecoechea, C., Campbell, A.T., and Hauw, L. “A Survey of QoS Architectures” *Multimedia Systems Journal* vol.6 no.3 pp.138-151 May 1998 (ACM/Springer Verlag)
8. Balachandran, A., Campbell, A.T., and Kounavis, M.E.: “Active Filters: Delivering Scaled Media to Mobile Devices” *Proceedings of the IEEE 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)* pp.125-34, 1997
9. Banerji, A., Cohn, D.L., and Kulkarni, D.C. “Mobile Computing Personae” *Distributed Computing Research Laboratory, Dept. of Computer Science, U. Notre Dame, IN, USA Technical Report* no.93-5
10. Berners-Lee, T. “Semantic Web Roadmap” W3C, September 1998 (from http://www.w3.org/2001/sw/)
11. Bhatti, N., Bouch, A., and Kuchinsky, A. “Integrating User-Perceived Quality into Web Server Design” *Proceedings of 9th International World-Wide-Web Conference (WWW-9)*, Amsterdam, Holland, May 2000
12. Bhatti, S. N., and Knight, G. “Notes on a QoS Information Model for Making Adaptation Decisions” *Journal of Computer Networks*, vol.31, 1999
13. Bickmore, T.W., and Schilit, B.N. “Digestor: Device-Independent Access to the World Wide Web” *Computer Networks and ISDN Systems* vol.29, no.8-13, pp.1075-1082, 1997
14. Björk, S., Holmquist, L.E., Redström, J., Bretan, I., Danielsson, R., Karlgren, J., and Franzén, K. “WEST: A Web Browser for Small Terminals” P*roceedings of ACM Conference on User Interface Software and Technology (UIST'99)*, 1999
15. Blair, G.S., Davies, N., Friday, A., and Wade, S.P. “Quality of service support in a mobile environment: an approach based on tuple spaces” *Proceedings of the 5th IFIP International Workshop on Quality of Service* New York, USA, May 1997, pp.37-48
16. v. Bochmann, G., and Hafid, A. “Some Principles for Quality of Service Management” *Distributed Systems Engineering* vol.4 pp.16-27, 1997
17. v. Bochmann, G., Kerhervé, B., Hafid, A., Dini, P., and Pons, A. “Architectural Design of Adaptive Distributed Multimedia Systems” *Proceedings of International Workshop in Distributed Multimedia Systems Design*, Berlin, Germany, 1996
18. Boll, S., Klas, W., and Wandel, J. “A Cross-Media Adaptation Strategy for Multimedia Presentations” *ACM Multimedia ‘99* Orlando, FL, USA, November 1999
19. Bolliger, J., and Gross, T. “A Framework-Based Approach to the Development of Network-Aware Applications” *IEEE Transactions on Software Engineering* vol.24, no.5, May 1998
20. Bouch, A., and Sasse, M.A. “The Case for Predictable Media Quality in Networked Multimedia Applications” P*roceedings of ACM/SPIE Multimedia Computing and Networking (MMCN'00)* San Jose, CA, USA, January 2000

-

21. Bouch, A., Sasse, M.A. “It Ain’t What You Charge, Its the Way that You Do It: A User Perspective of Network QoS and Pricing” P*roceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM'99)* Boston, MA, USA, May 1999, pp 639-655

22. Bowman, C.M., Danzig, P.B., Manber, U., and Schwartz, M.F. “Scalable Internet Resource Discovery: Research Problems and Approaches” *Communications of the ACM* vol.37, no.8, pp.98-107 1994

23. Brown, P. J., Bovey, J. D., and Chen, X. “Context-Aware Applications: From the Laboratory to the Marketplace” *IEEE Personal Communications*, vol.4 no.5 pp.58-64, October 1997

24. Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. “Easy Living: Technologies for Intelligent Environments” *Proceedings of 2nd International Symposium on Handheld and Ubiquitous Computing (HUC’2000)* Bristol, UK, September 2000 pp.12-29

25. Bulterman, D. C. A. “User-Centered Abstractions for Adaptive Hypermedia Presentations,” *Proceedings of ACM Multimedia '98*, Bristol, UK, 1998

26. Butler, M.H. “Current Technologies for Device Independence” *Hewlett Packard Technical Report* HPL-2001-83 March 2001

27. McCann, J.A., and Crane, J.S. “Kendra: Internet Distribution and Delivery System” *Proceedings of Society for Computer Simulation International EuroMedia Conference*, Leicester, UK, January 1998, pp.134--140

28. McCann, J.A., Howlett, P., and Crane, J.S. “Kendra: Adaptive Internet System” *Journal of Systems and Software* January 2001, Elsevier

29. Cartwright, W., Peterson, M.P., and Gartner, G. (eds) “Multimedia Cartography” (Springer -Verlag 1999)

30. Chalmers, D. “Quality of Service in Mobile Environments” *MSc Thesis* Imperial College, 1998, (also from *http://www.doc.ic.ac.uk/~dc/*)

31. Chalmers, D., and Sloman, M. S., “A Survey of Quality of Service in Mobile Computing Environments” *IEEE Communications Surveys*, *http://www.comsoc.org/pubs/surveys/*, vol.2, no.2. 1999

32. Chalmers, D., Sloman, M.S., and Dulay, N., “Map Adaptation for Users of Mobile Systems” *Proceedings of 10th International World-Wide-Web Conference (WWW-10)*, Hong Kong, May 2001 (ACM)

33. Chandra, S., Ellis, C.S., and Vahdat, A. “Multimedia Web Services for Mobile Clients Using Quality Aware Transcoding” *Proceedings of Workshop on Wireless Mobile Multimedia (WoWMoM’99)* pp.99-108, Seattle, WA, USA 1999

34. Chen, G., and Kotz, D. “A Survey of Context-Aware Mobile Computing Research” *Dartmouth College Technical Report* TR2000-381, November 2000

35. Cheverst, K., Davies, N., Mitchell, K., and Efstratiou, C. “Using Context as a Crystal Ball: Rewards and Pitfalls” *Personal Technologies Journal* vol.5, no.1, January 2001

36. Christel, M.G., Olligschlaeger, A.M., and Huang, C. “Interactive Maps for a Digital Video Library” *IEEE MultiMedia* pp.60-67 January-March 2000

37. Corona, B., Winter, S. “Guidance of Car Drivers and Pedestrians” *AAAI Symposium on Smart Graphics* March 2000

38. Davies, N., Blair, G.S., Cheverst, K., Friday, A. “A Network Emulator to Support the Development of Adaptive Applications” *Proceedings of 2nd USENIX Symposium on Mobile and Location Independent Computing* Ann Arbor, USA, April 1995

39. Davies, N. “The Impact of Mobility on Distributed Systems Platforms” *Proceedings of IFIP/IEEE International Conference on Distributed Platforms* 1996

40. Davies, N., Friday, A., Blair, G.S., and Cheverst, K. “Distributed Systems Support For Adaptive Mobile Applications” *ACM Mobile Networks and Applications, Special Issue on Mobile Computing - System Services* vol.1, no.4 (ACM, 1996)

41. Davies, N., Gellersen, H-W. “Beyond Prototypes: Challenges in Deploying Ubiquitous Systems” *IEEE Pervasive Computing* vol.1 no.1, pp.26-35, January-March 2002

42. Degermark, M., Köhler, T., Pink, S., and Schelén, O. “Advance Reservations for Predictive Service” *Proceedings of Network and Operating Systems Support for Digital Audio and Video (NOSSDAV’95)* (Springer-Verlag, 1995)

43. Delgrossi, L., Halstrick, C., Hehmann, D., Guido, R., Krone, O., Sandvoss, J., and Vogt, C.: “Media Scaling in a Multimedia Communication System” *Multimedia Systems* vol.2 pp.172-180 (Springer-Verlag, 1994)

44. Dey, A.K., and Abowd, G.D. “Towards a Better Understanding of Context and Context-Awareness” *Proceedings of Workshop on The What, Who, Where, When, and How of Context-Awareness, Conference on Human Factors in Computing Systems (CHI 2000)*, The Hague, The Netherlands, April 2000

45. Dublin Core Metadata Initiative *http://purl.org/dc*

-

46. Duchamp, D. “Issues in Wireless Mobile Computing” *Proceedings of 3rd IEEE Workshop on Workstation Operating Systems*, Key Biscayne, FL, USA, April 1992

47. Efstratiou, C., Cheverst, K., Davies, N., and Friday, A. “Architectural Requirements for the Effective Support of Adaptive Mobile Applications” *Proceedings of Middleware 2000*, New York, NY, USA, April, 2000

48. Fink, J., and Kobsa, A. “Adaptable and Adaptive Information Provision for All Users, Including Disabled and Elderly People” *The New Review of Hypermedia and Multimedia* vol.4 pp.163-188 1998

49. Fitzpatrick, T.R., Blair, G.S, Coulson, G., Davies, N., and Robin, P. “Software Architecture for Adaptive Distributed Multimedia Systems” *IEE Proceedings, Software* vol.145, no.5, October 1998

50. Flinn, J., and Satyanarayanan, M. “Energy-Aware Adaptation for Mobile Applications” *Proceedings of 17th ACM Symposium on Operating System Principles (SOSP-17)* Kiawah Island, SC, USA 1999

51. Fox, A., and Brewer, E.A. “Reducing WWW Latency and Bandwidth Requirements by Real-Time Distillation” *Proceedings of 5th International World-Wide-Web Conference (WWW5)* May, 1996, Paris, France

52. Fox, A., Gribble, S.D., Brewer, E.A., and Amir, E. “Adapting to Network and Client Variability via On-Demand Dynamic Distillation” *Proceedings of 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-VII)* Cambridge, MA, USA, 1996

53. Fox, A., Gribble, S. D., Chawathe, Y., and Brewer, E. A., “Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives” *IEEE Personal Communications*, 1998.

54. Fox, A., Goldberg. I, Gribble, S.D., Lee, D.C., Polito, A., and Brewer, E.A. “Experience with Top Gun Wingman: A Proxy-Based Graphical Web Browser for the 3Com PalmPilot” *Proceedings of Middleware '98* Lake District, England, September 1998

55. Friday, A., Davies, N., Blair, G.S., and Cheverst, K.W.J. “Developing Adaptive Applications: The MOST Experience” *Journal of Integrated Computer-Aided Engineering* vol.6 no.2 pp.143-157

56. Furnas, G.W., Landauer, T.K., Gomez, L.M., and Dumais, S.T. “The Vocabulary Problem in Human-System Communication: an Analysis and a Solution” *Communications of the ACM* vol.30, no.11, pp.964-971, 1987

57. Gecsei, J.: Adaptation in Distributed Multimedia Systems *IEEE Multimedia* vol.4 no.2 pp.58-66, April-June 1997

58. Gerfelder, N., Jung, H., Santos, L.M., and Belz, C. “Challenges to Deliver 2D/3D Content for Multimedia Applications in Mobile Environments” *Demonstrating the Future Wireless Information Infrastructure* vol.1, pp.82-89

59. Holtman, K., and Mutz, A. “Transparent Content Negotiation in HTTP” IETF Network Working Group, RFC 2295, 1998.

60. Holtman, K., and Mutz, A. “HTTP Remote Variant Selection Algorithm - RVSA/1.0” IETF Network Working Group, RFC 2296, 1998

61. Hori, M., Kondoh, G., Ono, K., Hirose, S., and Singhal, S., “Annotation-Based Web Content Transcoding,” *Proceedings of 9th International World-Wide-Web Conference (WWW9)*, Amsterdam, Netherlands, May 2000

62. Hoschka, P. (ed) “Synchronized Multimedia Integration Language (SMIL) 1.0 Specification” W3C Recommendation W3C REC-smil-19980615, (W3C 1998)

63. Huang, J., Wan, P.-J., and Du, D.-Z. “Criticality- and QoS Based Multiresource Negotiation and Adaptation” *Journal of Real-Time Systems* 1998

64. McIlhagga, M., Light, A., and Wakeman, I. “Giving Users the Choice between a Picture and a Thousand Words” 1998 *Proceedings of 1st Workshop on HCI and Mobile Devices*, Glasgow, UK, May 1998

65. McIlhagga, M., Light, A., and Wakeman, I. “Towards a Design Methodology for Adaptive Applications” *Proceedings of 4th Annual ACM/IEEE International Conference on Mobile Computing and Networks, (MobiCom’98)* pp.133-144, Dallas, Texas, USA, October 1998

66. Imielinski, T., and Badrinath, B.R. “Mobile Wireless Computing” *Communications of the ACM* vol.37, no.10, 1994

67. Jones, M, Marsden, G., Mohd-Nasir, N., Boone, K. and Buchanan, G. “Improving Web Interaction on Small Displays” *Proceedings of 8th International World-Wide-Web Conference (WWW8)* Toronto, Canada, May 1999

68. Jones, M., Buchanan, G., Thimbleby, H., and Marsden, G. “User Interfaces for Mobile Web Devices” *Position Paper for WWW9 Workshop on Information Design for Mobile Web Services Mobile Workshop* Amsterdam, Netherlands, May 2000

69. Katz, R.H. “Adaptation and Mobility in Wireless Information Systems” *IEEE Personal Communications* vol.1 no.1 pp.6-17, 1995

70. Kerhervé, B., Pons, A., v. Bochmann, G., and Hafid, A. “Metadata Modelling for Quality of Service Management in Distributed Multimedia Systems” *Proceedings of 1st IEEE Metadata Conference* Silver Spring, MD, USA April 1996

-

71. Kermarrec, A.-M., Couderc, P., Banâtre, M. “Introducing Contextual Objects in an Adaptive Framework for Wide-area Mobile Computing” *Proceedings of 8th ACM Special Interest Group on Operating Systems (SIGOPS) European Workshop* September 1998, Sintra, Portugal

72. Kim, M., Noble, B. “Mobile Network Estimation” *Proceedings of 7th International Conference on Mobile Computing and Networking (MobiCom 2001)* July, 2001, Rome, Italy, pp.298-309 (ACM)

73. Klyne, G. “A Syntax for Describing Media Feature Sets” IETF RFC 2533, 1999

74. Koistinen, J., and Seetharaman, A.: “Worth Based Multi Category Quality Of Service Negotiation In Distributed Object Infrastructures” *Hewlett-Packard Research Report* HPL-98-51, 1998

75. Kravets, R., Calvert, K., and Schwan, K. “Payoff Adaptation of Communication for Distributed Interactive Applications” *Journal for High Speed Networking* August 1999

76. van Laerhoven, K., and Aidoo, K. “Teaching Context to Applications” *Personal and Ubiquitous Computing* vol.5 pp.46-49 (Springer-Verlag, 2001)

77. Lassila, O., Swick, R.R. (eds) “Resource Description Framework (RDF) Model and Syntax Specification” W3C Recommendation W3C-rdf-syntax-19990222 (W3C 1999)

78. Lee, C., Lehoczky, J., Rajkumar, R., and Siewiorek, D. “On Quality of Service Optimization with Discrete QoS Options” *Proceedings of IEEE Real Time Technology and Applications Symposium (RTAS’99)* Vancouver, British Columbia, Canada, June 1999 pp.276-286

79. Lehikoinen, J., and Suomela, R. “Accessing Context in Wearable Computers” *Personal and Ubiquitous Computing* vol.6 pp.64-74 (Springer-Verlag, 2002)

80. Leonhardt, U., and Magee, J. “Towards A General Location Service For Mobile Environments” *Proceedings of 3rd IEEE Workshop on Services in Distributed and Networked Environments* Macau, June 1996, pp.43-50

81. Leonhardt, U., and Magee, J. “Multi-Sensor Location Tracking” *Proceedings of 4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’98)* October 1998 Dallas, TX, USA, pp.203-214

82. Loon, T.S., and Bharghavan, V. “Alleviating the Latency and Bandwidth Problems in WWW Browsing” *Proceedings of USENIX Symposium on Internet Technologies and Systems* 1997

83. Lu, S., and Bharghavan, V. “Adaptive Resource Management Algorithms for Indoor Mobile Computing Environments” *Conference of the Association for Computing Machinery's Special Interest Group on Data Communication (SIGCOMM’96)* Stanford, California, USA August 1996 pp.231-242 (ACM)

84. Lu, S., Lee, K-W., and Bharghavan, V. “Adaptive Service in Mobile Computing Environments” *Proceedings of the 5th IFIP International Workshop on Quality of Service (IWQoS’97)* (Chapman & Hall, 1997)

85. Ma, W.-Y., Bedner, I., Chang, G., Kuchinsky, A., and Zhang, H. J. “A Framework for Adaptive Content Delivery in Heterogeneous Network Environments” *Proceedings of Multimedia Computing and Networking (MMCN00)*, San Jose, California, USA, 2000

86. Magee, J., Dulay, N., and Kramer, J. “Regis: A Constructive Development Environment for Parallel and Distributed Programs” Pr*oceedings of the International Workshop on Configurable Distributed Systems (IWCDS)*, Pittsburgh, PA, USA, March 1994

87. Marttila, O., and Vuorimaa, P. “XML Based Mobile Services” *Proceedings of 8th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2000)* Plzen - Bory, Czech Republic, February 2000

88. Mohan, R., Smith, J.R., and Li, C.-S. “Adapting Multimedia Internet Content for Universal Access” *IEEE Transactions Multimedia*, vol.1, no.1, pp.104-114, March 1999

89. Mohan, R., Smith, J.R., and Li, C-S. “Adapting Content to Client Resources in the Internet” *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS)* Florence, Italy, June 1999 pp.303-307 (IEEE)

90. Nagao, K., Shirai, Y., Squire, K. “Semantic Annotation and Transcoding: Making Web Content More Accessible” *IEEE Multimedia* April-June 2001, pp.69-81

91. Neilson, J. “Designing Web Usability” (New Riders Publishing, 2000)

92. Nicholson, D., and Whittle, S. “3G will Crash and Burn Before the Finish Line” *Computing* 13th September 2001, pp.29-32

93. Noble, B.D., Satyanarayanan, M., Narayanan, D., Tilton, J. E., Flinn, J., and Walker, K. R., “Agile Application-Aware Adaptation for Mobility” *Proceedings of 16th Symposium on Operating System Principles (SOSP-16)* Saint-Malo, France, 1997 (ACM)

94. Noble, B.D. “System Support for Mobile, Adaptive Applications” *IEEE Personal Communications* pp.44-49, February 2000

-

95. Open GIS Consortium “Geography Mark-up Language (CML) 2.0” February 2001 (from *http://www.opengis.net/)*
96. Pascoe, J., Ryan, N., Morse, D. “Using While Moving: HCI Issues in Fieldwork Environments” *ACM Transactions on Computer-Human Interaction* vol.7 no.3, pp.417-437, September 2000
97. Petrelli, D., Not, E., Strapparava, C., Stock, O., and Zancanaro, M. “Modelling Context is Like Taking Pictures” *Proceedings of Workshop on "The What, Who, Where, When, Why and How of Context-Awareness" at CHI2000* The Hague, Netherlands, April 2000
98. Petrelli, D., De Angeli, A., Convertino, G. “A User-Centered Approach to User Modelling” *Proceedings of the Seventh International Conference on User Modelling (UM'99)* Banff, Canada, June 1999, pp.255-264 (Springer)
99. Reynolds, F., Hjelm, J., Dawkins, S., and Singhal, S. (eds), “Composite Capability / Preference Profiles (CC/PP): A User Side Framework for Content Negotiation” W3C NOTE-CCPP-19990727, July 1999
100. DeRoure, D., Hall, W., and Hill, G. “A Distributed Hypermedia Link Service,” *Proceedings of 3rd International Workshop on Services in Distributed and Networked Environments* Macau, 1996 (IEEE)
101. Salber, D., Dey, A.K., and Abowd, G.D. “The Context Toolkit: Aiding the Development of Context-Enabled Applications” *Proceedings of Computer-Human Interaction Conference (CHI’99)* Pittsburgh, PA, USA, May 1999 (ACM)
102. Satyanarayanan, M., and Narayanan, D. “Multi-Fidelity Algorithms for Interactive Mobile Applications” *Proceedings of Workshop on Algorithms and Methods for Mobile Computing and Communications (Dial M’99)* Seattle, WA, USA August 1999 (ACM)
103. Schilit, B. N., Adams, N., and Want, R. “Context-Aware Computing Applications” *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications* 1994.
104. Schmidt, A. “Implicit Human Computer Interaction Through Context” *Personal Technologies* vol. 4 no.2&3, pp.191-199, June 2000 (Springer-Verlag)
105. Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., and Van de Velde, W. “Advanced Interaction in Context” *Proceedings of 1st International Symposium on Handheld and Ubiquitous Computing (HUC’99)* pp.89-101, Karlsruhe, Germany, September 1999 (Springer)
106. Smith, J.R., Mohan, R., and Li, C-S., “Content-Based Transcoding of Images in the Internet” *Proceedings of IEEE International Conference on Image Processing (ICIP-98)*
107. Smith, J.R., Mohan, R., and Li, C-S. “Transcoding Internet Content for Heterogeneous Client Devices” *Proceedings of IEEE International Conference on Circuits and Systems (ISCAS) May 1998*
108. Sreenan, C.J., and Mishra, P.P.: Equus “A QoS Manager for Distributed Applications” *Proceedings of the IFIP / IEEE International Conference on Distributed Platforms, Dresden* pp496-509 (Chapman & Hall, 1996)
109. Srivastava, M., and Mishra, P.P. “On Quality of Service in Mobile Wireless Networks” *Proceedings of the IEEE 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV’97)* pp147-58 (IEEE, 1997)
110. Thimbleby, H. “Treat People like Computers? Designing Usable Systems for Special People” *People and Human-Computer-Interaction* ed A. Edwards (Cambridge University Press, 1994)
111. Turner, R.M.: Context-Mediated Behaviour for Intelligent Agents *International Journal of Human-Computer Studies* vol.48, pp307-330 (Academic Press Ltd., 1998)
112. Vanderheiden, G.C. “Anywhere, Anytime (+Anyone) Access to the Next-Generation WWW” *Proceedings of 6th International World Wide Web Conference (WWW6)* Santa Clara, CA, USA, April 1997
113. Veitch, M “Bandwidth limits for 3G” *ZDNet News* Sept. 2001 *(*from *http://news.zdnet.co.uk/story/0,,t269-s2094834,00.html)*
114. Waddington, D.G., and Hutchison, D. “A General Model for QoS Adaptation May 1998” *Proceedings of 6th International Workshop on Quality of Service (IWQoS'98)* Napa, California, USA, May 1998
115. Walpole, J., Krasic, C., Liu, L., Maier, D., Pu, C., McNamee, D., and Steere, D. “Quality of Service Semantics for Multimedia Database Systems” *Proceedings of Data Semantics (DS-8)* Rotorua, NZ, 1999
116. Want, R., Hopper, A., Falcao, V., and Gibbons, J. “The Active Badge Location System” *ACM Transactions on Information Systems* vol.10 no.1, pp 91-102, January 1992
117. WAP Forum, “Wireless Mark-up Language Specification Version 1.2” from www.wapforum.org, 1999
118. Watson, A., and Sasse, M.A. “Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications” *ACM Multimedia ‘98* September 1998, Bristol, UK pp.55-60
119. Watson, A., and Sasse, M.A. “Multimedia Conferencing via Multicast: Determining the Quality of Service Required by the End User” *Proceedings of International Workshop on Audio-Visual Services over Packet Networks (AVSPN '97)* September 1997, Aberdeen, UK, pp.189-194

-

120. Weiser, M. “Some Computer Science Issues in Ubiquitous Computing” *Communications of the ACM*, vol.36, pp.75-85, 1993.

121. Welling, G., and Badrinath, B.R. “An Architecture for Exporting Environment Awareness to Mobile Computing Applications” *IEEE Transactions on Software Engineering* vol.24, no.5, May 1998

122. Wilson, G.M., and Sasse, M.A. “The Head or the Heart? Measuring the Impact of Media Quality” *Proceedings of Conference on Human Factors in Computing Systems (CHI2000)* The Hague, Netherlands, April 2000, pp.117-118 (ACM)

123. Wirag, S. “Modelling of Adaptable Multimedia Documents” *Proceedings of European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS’97)* Darmstadt, Germany, September 1997

124. Wirag, S. “Adaptive Scheduling of Multimedia Documents” *U. Stuttgart, Technical Report* 1997/12, Sept. 1997

125. Witana, V., Fry, M., and Antoniades, M. “A Software Framework for Application Level QoS Management” *Proceedings of International Workshop on Quality of Service (IWQoS'99)* London, UK, 1999 (IEEE)

126. Ye, T., Jacobsen, H. A., and Katz, R. “Mobile Awareness in a Wide Area Wireless Network of Info-Stations,” *Proceedings of 4th International Conference on Mobile Computing and Networking (MobiCom'98)* Dallas, TX, USA, 1998 (ACM)

127. Zenel, B., and Duchamp, D. “Intelligent Communication Filtering for Limited Bandwidth Environments” *Proceedings of the 5th Workshop on Hot Topics in Operating Systems,* Orcas Island, WA, USA, May 1995, pp.28-34 (IEEE)

128. Zhang, H., and Knightly, E.W. “RED-VBR: A Renegotiation-Based Approach to Support Delay-Sensitive VBR Video” *Multimedia Systems* vol.5 pp.164-176 (Springer-Verlag, 1997)

129. Zipf, A. “User-Adaptive Maps for Location-Based Services (LBS) for Tourism” *Proceedings of ENTER 2002. International Congress on Tourism and Communications Technologies in Tourism* Innsbruck, Austria, 2002 (Springer)

APPENDIX 1 LOCATION MODEL

A1.1 A LOCATION SERVICE IN A MAP BASED APPLICATION

The location service implemented is based on that described in [81]. Our location objects described a simple rectilinear area, with no elevation description or sub-structure. The architecture is illustrated along with data flow in figure a1-1. Multiple location information sources describe locations to a location manager. This applies a fusion function in order to describe a best-estimate (both in accuracy and reliability) of the location of interest (generally the current location). Using a history of locations and other data future locations may also be predicted. Applications can subscribe to the location manager to receive update events when the modelled location changes. Our map application does this. The location events arrive at a model of location for display. This may modify the location by expanding one dimension such that the described location reflects the aspect ratio of the display. The map user interface acts as a location source, which may provide updates to the location service as described below.

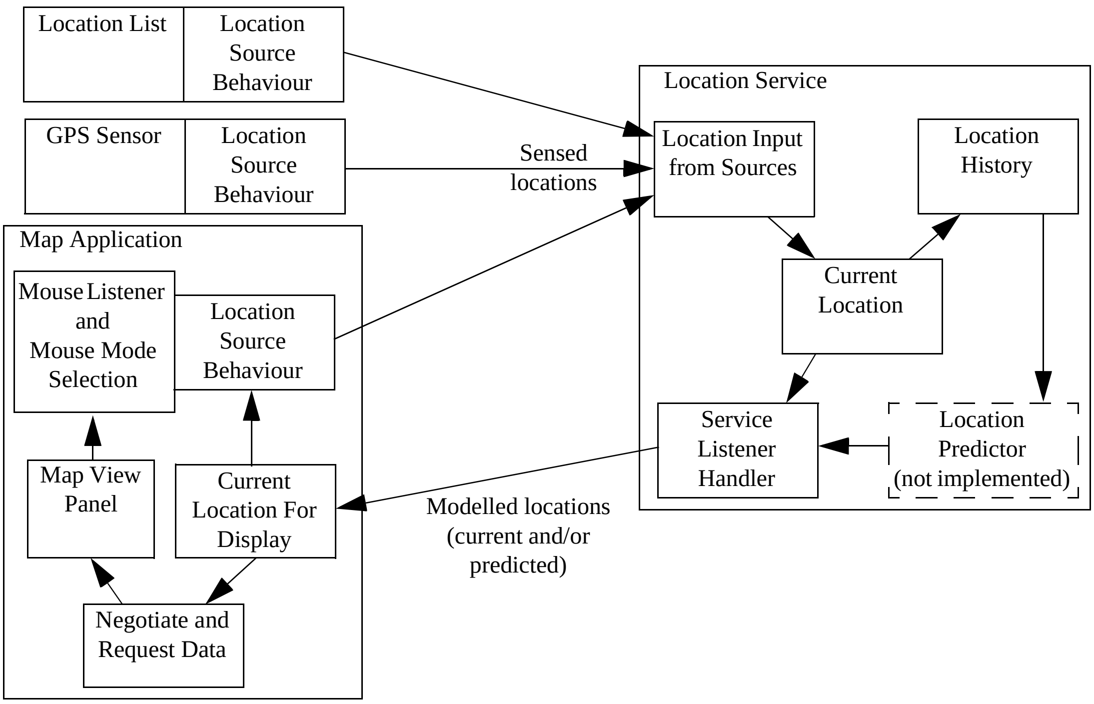


Figure A1-1: Location Service within Map Demo Application

The map application used a single "click" interface to aid portability between devices. Performing accurate selection using a pen interface or touch screen is hard enough, double clicks and right clicks are often impossible to achieve with any accuracy. The selection may be modified using context awareness but for the prototype a simple generic mechanism was chosen. A tool bar for mode selection was therefore required. The mode selection buttons on the top left of the application interface can be seen in figure a1-2. The modes implemented were (left to right): "jump", "centre" and "zoom". Each of these use clicks on the map to modify the location the map application location source provides to the location manager. The "no location" button (to the right of "zoom") causes the map application's location source to register an unknown location and so cause other location sources to override any location the user has described using navigation through the map application.

In jump mode a click on the map causes the application location source to register a location of the same size as that displayed, immediately adjoining the current location, in the direction of the quadrant the click was registered in. Centre mode sets a location of the same size as that displayed, centered on the click point. Zoom mode responds to both a click and the selection of an area. A click causes a zoom out centered on the click point. The zoomed map covers an area 9 times the size of the current area (3 times the size in each axis). Where an area is selected the location registered is that selected. In all cases the application may expand the location the map requests data for to reflect the aspect ratio of the map display.

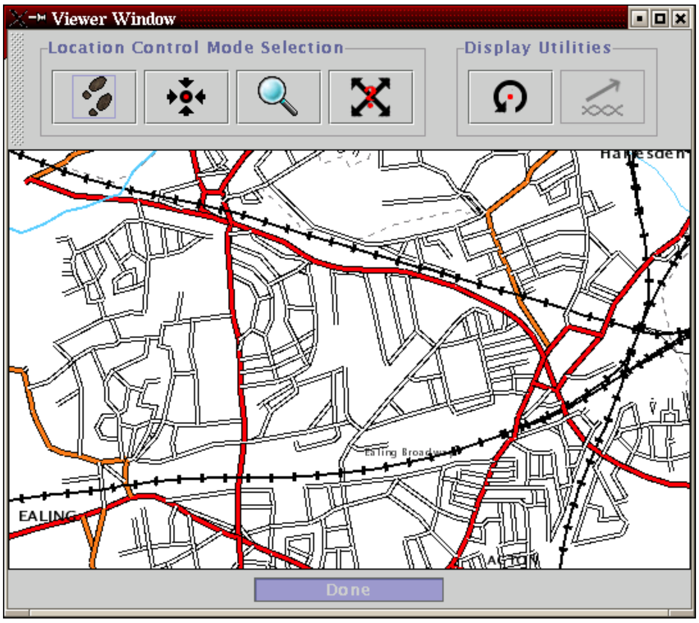


Figure A1-2: Location Service Controls in GUI of Map Demo Application

APPENDIX 2 TYPE, META DATA AND SPECIFICATION ENCODINGS

A2.1 TYPE LANGUAGE

A2.1.1 Type Language Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:complexType name="typeType">
        <xs:sequence>
            <xs:element name="type" type="typeType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:element name="type-domain">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="type" type="typeType" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="name" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="map"/>
                    <xs:enumeration value="system"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

A2.1.2 Type Language Example

The following type definitions are typical of those used to perform the testing in the map application.

```
<type-domain name="system">
    <type name="*"/>
    <type name="structural"/>
</type-domain>

<type-domain name="map">
    <type name="*"/>
    <type name="administrative">
        <type name="county"/>
        <type name="district"/>
        <type name="electoral"/>
        <type name="european"/>
        <type name="national"/>
        <type name="parish"/>
```

```
    <type name="boundary"/>
    <type name="name"/>
    <type name="national-park"/>
</type>
<type name="coastline"/>
<type name="antiquity"/>
<type name="barrier">
    <type name="fence"/>
</type>
<type name="building">
    <type name="airport"/>
    <type name="church"/>
    <type name="glasshouse"/>
    <type name="lighthouse"/>
    <type name="number-name"/>
    <type name="phone"/>
    <type name="pylon"/>
    <type name="radio-mast"/>
    <type name="railway-station"/>
    <type name="urban-area"/>
    <type name="windmill"/>
    <type name="youth-hostel"/>
</type>
<type name="cartographic-notes">
    <type name="alignments">
        <type name="underground"/>
        <type name="overhead"/>
    </type>
    <type name="copyright"/>
    <type name="footnotes"/>
    <type name="gridline"/>
    <type name="survey-benchmark"/>
</type>
<type name="tourist-feature">
    <type name="campsite"/>
    <type name="information-center"/>
    <type name="viewpoint"/>
</type>
<type name="land-use">
    <type name="sports-ground"/>
    <type name="park"/>
    <type name="zoo"/>
</type>
<type name="land-cover">
    <type name="boulder"/>
    <type name="boundary"/>
    <type name="building"/>
    <type name="grass"/>
    <type name="heath"/>
    <type name="marsh"/>
    <type name="rock"/>
    <type name="sand"/>
    <type name="scree"/>
    <type name="scrub"/>
    <type name="woodland">
        <type name="coniferous"/>
        <type name="coppice"/>
        <type name="deciduous"/>
        <type name="mixed"/>
        <type name="orchard"/>
    </type>
    <type name="vegetation"/>
</type>
<type name="marker">
    <type name="boundary-post-stone"/>
    <type name="trigpoint"/>
    <type name="milepost"/>
</type>
<type name="railway">
```

```
        <type name="narrow-guage"/>
    </type>
    <type name="path">
        <type name="national-trail"/>
        <type name="bridleway"/>
        <type name="footpath"/>
    </type>
    <type name="road">
        <type name="a-road"/>
        <type name="b-road"/>
        <type name="byway"/>
        <type name="center-line"/>
        <type name="dual-carriageway"/>
        <type name="junction"/>
        <type name="under-construction"/>
        <type name="motorway"/>
        <type name="minor-road"/>
        <type name="name-number"/>
        <type name="primary-route"/>
    </type>
    <type name="topology">
        <type name="cliff"/>
        <type name="slope"/>
        <type name="spot-height"/>
    </type>
    <type name="water">
        <type name="canal"/>
        <type name="feature"/>
        <type name="ferry"/>
        <type name="flow"/>
        <type name="lake"/>
        <type name="name"/>
        <type name="river"/>
        <type name="sea">
            <type name="high-water"/>
            <type name="low-water"/>
        </type>
    </type>
</type-domain>
```

A2.2 META DATA LANGUAGE

A2.2.1 Meta Data Language Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:complexType name="aType">
        <xs:attribute name="sb" type="xs:short" use="required"/>
        <xs:attribute name="scale" type="xs:int"/>
        <xs:attribute name="price" type="xs:short"/>
        <xs:attribute name="date">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="2000-275-0:0"/>
                    <xs:enumeration value="2001-32-0:0"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="areaType">
        <xs:attribute name="minX" type="xs:int" use="required"/>
        <xs:attribute name="minY" type="xs:int" use="required"/>
        <xs:attribute name="maxX" type="xs:int" use="required"/>
        <xs:attribute name="maxY" type="xs:int" use="required"/>
    </xs:complexType>
```

```
    <xs:complexType name="eType">
        <xs:choice maxOccurs="unbounded">
            <xs:element name="t" type="tType"/>
            <xs:element name="e" type="eType"/>
            <xs:element name="eg" type="egType"/>
            <xs:element name="area" type="areaType"/>
        </xs:choice>
        <xs:attribute name="n" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="egType">
        <xs:sequence>
            <xs:element name="t" type="tType" maxOccurs="unbounded"/>
            <xs:element name="area" type="areaType"/>
        </xs:sequence>
        <xs:attribute name="n" type="xs:hexBinary" use="required"/>
        <xs:attribute name="eno" type="xs:byte" use="required"/>
    </xs:complexType>
    <xs:element name="meta">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="e" type="eType"/>
                <xs:element name="v" type="vType" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="typedomain" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="rType">
        <xs:attribute name="n" type="xs:hexBinary" use="required"/>
    </xs:complexType>
    <xs:complexType name="tType">
        <xs:attribute name="n" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="vType">
        <xs:sequence>
            <xs:element name="r" type="rType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="a" type="aType"/>
            <xs:element name="v" type="vType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="n" type="xs:string" use="required"/>
    </xs:complexType>
</xs:schema>
```

A2.2.2 Meta Data Example

The following meta data is an extract of that returned by the server used to perform the testing in the map application.

```
<?xml version="1.0" standalone="yes"?>
<meta typedomain="map">
<e n="529500177000530499177999-header"><t n="system:structural"/>
<e n="6d7905df"><t n="administrative"/><area minX="530000" minY="170000" maxX="530527"
maxY="177481"/></e>
<eg n="27a305de" eno="2"><t n="building:railway-station"/><t n="railway"/><area
minX="530446" minY="177991" maxX="530446" maxY="178036"/></eg>
<eg n="6af505de" eno="4"><t n="railway"/><area minX="530000" minY="177508" maxX="532101"
maxY="180890"/></eg>
<eg n="4e0ac5d8" eno="16"><t n="road:a-road"/><area minX="530000" minY="177511"
maxX="530494" maxY="178073"/></eg>
<eg n="4f28c5d8" eno="1"><t n="road:b-road"/><area minX="530367" minY="177574"
maxX="530487" maxY="177633"/></eg>
<eg n="65c045da" eno="21"><t n="road:minor-road"/><area minX="530000" minY="177050"
maxX="530532" maxY="177511"/></eg>
<eg n="40500597" eno="8"><t n="road"/><area minX="528920" minY="177082" maxX="529377"
maxY="177496"/></eg>
</e>
```

```
<v n="TQ37"><a sb="4404"/>
<v n="6d7cc5df"><r n="6d7905df"/><a sb="120" scale="10000" date="2000-275-0:0"/></v>
<v n="278305de"><r n="27a305de"/><a sb="92" scale="2500" date="2000-275-0:0"/></v>
<v n="6a52c5de"><r n="6af505de"/><a sb="1480" scale="2500" date="2000-275-0:0"/></v>
<v n="6a5685de"><r n="6af505de"/><a sb="1189" scale="25000" date="2000-275-0:0"/></v>
<v n="6a7b05de"><r n="6af505de"/><a sb="611" scale="250000" date="2000-275-0:0"/></v>
<v n="4e6e85d8"><r n="4e0ac5d8"/><a sb="2898" scale="2500" date="2000-275-0:0"/></v>
<v n="4e1c05d8"><r n="4e0ac5d8"/><a sb="2887" scale="250000" date="2000-275-0:0"/></v>
<v n="4f04c5d8"><r n="4f28c5d8"/><a sb="141" scale="2500" date="2000-275-0:0"/></v>
<v n="652285da"><r n="65c045da"/><a sb="3048" scale="2500" date="2000-275-0:0"/></v>
<v n="652645da"><r n="65c045da"/><a sb="3037" scale="25000" date="2000-275-0:0"/></v>
<v n="65d545da"><r n="65c045da"/><a sb="3004" scale="250000" date="2000-275-0:0"/></v>
</v>
<v n="TQ27"><a sb="4404"/>
<v n="6c73c587"><r n="6c238587"/><a sb="271" scale="50000" date="2000-275-0:0"/></v>
<v n="6c668587"><r n="6c238587"/><a sb="120" scale="500000" date="2000-275-0:0"/></v>
<v n="6088c5b8"><r n="6d2d8587"/><r n="7f3085b9"/><r n="6c238587"/><a sb="81"
scale="250000" date="2000-275-0:0"/></v>
<v n="41bfc597"><r n="40500597"/><a sb="864" scale="250000" date="2000-275-0:0"/></v>
</v>
</meta>
```


A2.3 SPECIFICATION LANGUAGE

A2.3.1 Specification Language DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:complexType name="aspectrangeType">
        <xs:attribute name="min" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="0"/>
                    <xs:enumeration value="16"/>
                    <xs:enumeration value="21"/>
                    <xs:enumeration value="46"/>
                    <xs:enumeration value="51"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="max" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="15"/>
                    <xs:enumeration value="20"/>
                    <xs:enumeration value="200"/>
                    <xs:enumeration value="45"/>
                    <xs:enumeration value="50"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="aspectvalueType">
        <xs:attribute name="value" type="xs:string" use="required"/>
        <xs:attribute name="default" type="xs:boolean"/>
    </xs:complexType>
    <xs:complexType name="functionType">
        <xs:sequence>
            <xs:element name="range" type="rangeType" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="parameter">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="age"/>
                    <xs:enumeration value="distance"/>
```

```
                    <xs:enumeration value="maxweight"/>
                    <xs:enumeration value="scale"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="goalType">
        <xs:sequence>
            <xs:element name="function" type="functionType"/>
        </xs:sequence>
        <xs:attribute name="resource" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="price"/>
                    <xs:enumeration value="screendensity"/>
                    <xs:enumeration value="time"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="matchType">
        <xs:choice>
            <xs:element name="aspectvalue" type="aspectvalueType"/>
            <xs:element name="aspectrange" type="aspectrangeType"/>
        </xs:choice>
        <xs:attribute name="aspect" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="mapuse"/>
                    <xs:enumeration value="speed"/>
                    <xs:enumeration value="task"/>
                    <xs:enumeration value="transport"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="rangeType">
        <xs:attribute name="out" type="xs:decimal" use="required"/>
        <xs:attribute name="to" type="xs:decimal"/>
        <xs:attribute name="rate">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="0"/>
                    <xs:enumeration value="1"/>
                    <xs:enumeration value="1.5"/>
                    <xs:enumeration value="2"/>
                    <xs:enumeration value="3"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="selectufType">
        <xs:sequence>
            <xs:element name="function" type="functionType"/>
        </xs:sequence>
        <xs:attribute name="format" type="xs:string" use="required"/>
        <xs:attribute name="type" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="specType">
        <xs:sequence>
            <xs:element name="match" type="matchType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="weight" type="weightType" maxOccurs="unbounded"/>
            <xs:element name="goal" type="goalType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="selectuf" type="selectufType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="goal" type="goalType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>
```

```
    <xs:element name="specset">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="spec" type="specType" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="typedomain" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="map"/>
                    <xs:enumeration value="system"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="weightType">
        <xs:sequence>
            <xs:element name="function" type="functionType"/>
        </xs:sequence>
        <xs:attribute name="type" type="xs:string" use="required"/>
    </xs:complexType>
</xs:schema>
```

A2.3.2 Specification Language Example

The following specification illustrates that in the files used to perform the testing in the map application.

```
<specset typedomain="system">
<spec name="general-0">
    <weight type="*">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="structural">
        <function>
            <range out="0.000001"/>
        </function>
    </weight>
    <goal resource="time">
        <function>
            <range out="60"/>
        </function>
    </goal>
    <goal resource="price">
        <function>
            <range out="300"/>
        </function>
    </goal>
    <goal resource="screendensity">
        <function>
            <range out="2"/>
        </function>
    </goal>
</spec>
</specset>

<specset typedomain="map">
<spec name="general-map">
    <weight type="*">
        <function>
                <range out="0.0001"/>
                 </function>
    </weight>
```

```
    <selectuf format="map" type="*">
        <function parameter="scale">
            <range to="0" out="1"/>
            <range to="5000" out="1"/>
            <range to="1000000" out="0.3" rate="2"/>
         </function>
    </selectuf>
    <selectuf format="map" type="*">
        <function parameter="age">
            <range to="0" out="1"/>
            <range to="525600" out="0.7" rate="1"/>
            <range to="5256000" out="0.4" rate="1"/>
         </function>
    </selectuf>
</spec>
</specset>

<specset typedomain="map">
<spec name="work">
    <match aspect="task">
        <aspectvalue value="work"/>
      </match>
    <weight type="tourist-feature">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="land-use:park">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="antiquity">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="building:airport">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
    <weight type="building:phone">
        <function>
            <range out="0.5"/>
        </function>
    </weight>
    <weight type="building:youth-hostel">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="land-use:zoo">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="administrative:national-park">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="path:national-trail">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <goal resource="time">
        <function>
```

```
            <range out="45"/>
        </function>
    </goal>
    <goal resource="price">
        <function>
            <range out="500"/>
        </function>
    </goal>
         <goal resource="price">
        <function parameter="maxweight">
            <range to="0" out="0"/>
            <range to="0.000000001" out="10" rate="1"/>
            <range to="0.00001" out="600" rate="2"/>
         </function>
         </goal>
</spec>

<spec name="recreation">
    <match aspect="task">
        <aspectvalue value="recreation"/>
      </match>
    <weight type="tourist-feature">
        <function>
            <range out="0.6"/>
        </function>
    </weight>
    <weight type="land-use:park">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
    <weight type="building:railway-station">
        <function>
            <range out="0.6"/>
        </function>
    </weight>
    <weight type="building:phone">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
    <goal resource="time">
        <function>
            <range out="60"/>
        </function>
    </goal>
    <goal resource="price">
        <function>
            <range out="400"/>
        </function>
    </goal>
         <goal resource="price">
        <function parameter="maxweight">
            <range to="0" out="0"/>
            <range to="0.000000001" out="10" rate="1"/>
            <range to="0.00001" out="400" rate="1.5"/>
         </function>
         </goal>
</spec>

<spec name="hiking">
    <match aspect="task">
        <aspectvalue value="hiking"/>
      </match>
    <weight type="tourist-feature">
        <function>
            <range out="0.8"/>
        </function>
    </weight>
```

```
<weight type="land-use:park">
    <function>
        <range out="0.8"/>
    </function>
</weight>
<weight type="barrier">
    <function>
        <range out="0.8"/>
    </function>
</weight>
<weight type="building:railway-station">
    <function>
        <range out="0.7"/>
    </function>
</weight>
<weight type="building:phone">
    <function>
        <range out="0.7"/>
    </function>
</weight>
<weight type="building:youth-hostel">
    <function>
        <range out="0.5"/>
    </function>
</weight>
<weight type="administrative:national-park">
    <function>
        <range out="0.75"/>
    </function>
</weight>
<weight type="land-cover">
    <function>
        <range out="0.9"/>
    </function>
</weight>
<weight type="path">
    <function>
        <range out="1"/>
    </function>
</weight>
<weight type="road">
    <function>
        <range out="0.91"/>
    </function>
</weight>
<weight type="road:byway">
    <function>
        <range out="1"/>
    </function>
</weight>
<weight type="road:minor-road">
    <function>
        <range out="0.95"/>
    </function>
</weight>
<selectuf format="map" type="*">
    <function parameter="scale">
        <range to="0" out="0.5"/>
        <range to="1000" out="0.5"/>
        <range to="10000" out="1"/>
        <range to="50000" out="1"/>
        <range to="100000" out="0.6"/>
        <range to="1000000" out="0.2" rate="1"/>
     </function>
</selectuf>
<goal resource="time">
    <function>
        <range out="70"/>
    </function>
```

```
    </goal>
    <goal resource="price">
        <function>
            <range out="300"/>
        </function>
    </goal>
        <goal resource="price">
        <function parameter="maxweight">
            <range to="0" out="0"/>
            <range to="0.000000001" out="10" rate="1"/>
            <range to="0.00001" out="400" rate="1.5"/>
        </function>
        </goal>
</spec>

<spec name="tourist">
    <match aspect="task">
        <aspectvalue value="tourist"/>
      </match>
    <weight type="tourist-feature">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="coastline">
        <function>
            <range out="0.9"/>
        </function>
    </weight>
    <weight type="barrier">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="land-use:park">
        <function>
            <range out="0.8"/>
        </function>
    </weight>
    <weight type="antiquity">
        <function>
            <range out="0.9"/>
        </function>
    </weight>
    <weight type="building:airport">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
    <weight type="building:church">
        <function>
            <range out="0.75"/>
        </function>
    </weight>
    <weight type="building:lighthouse">
        <function>
            <range out="0.75"/>
        </function>
    </weight>
    <weight type="building:windmill">
        <function>
            <range out="0.75"/>
        </function>
    </weight>
    <weight type="building:phone">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
```

```
    <weight type="building:railway-station">
        <function>
            <range out="0.8"/>
        </function>
    </weight>
    <weight type="building:youth-hostel">
        <function>
            <range out="0.65"/>
        </function>
    </weight>
    <weight type="land-use:zoo">
        <function>
            <range out="0.75"/>
        </function>
    </weight>
    <weight type="administrative:national-park">
        <function>
            <range out="0.6"/>
        </function>
    </weight>
    <weight type="land-cover">
        <function>
            <range out="0.4"/>
        </function>
    </weight>
    <weight type="water">
        <function>
            <range out="0.4"/>
        </function>
    </weight>
    <weight type="path">
        <function>
            <range out="0.5"/>
        </function>
    </weight>
    <selectuf format="map" type="*">
        <function parameter="scale">
            <range to="0" out="0.8"/>
            <range to="5000" out="1"/>
            <range to="200000" out="1"/>
            <range to="1000000" out="0.8" rate="1"/>
         </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="45"/>
        </function>
    </goal>
    <goal resource="price">
        <function>
            <range out="400"/>
        </function>
    </goal>
         <goal resource="price">
        <function parameter="maxweight">
            <range to="0" out="0"/>
            <range to="0.000000001" out="50" rate="1"/>
            <range to="0.00001" out="400" rate="2"/>
         </function>
         </goal>
</spec>
</specset>

<specset typedomain="map">
<spec name="navigatingincar">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
      </match>
    <match aspect="transport">
```

```
    <aspectvalue value="in-car"  default="true"/>
  </match>
<weight type="building">
    <function>
        <range out="0.73"/>
    </function>
</weight>
<weight type="building:airport">
    <function>
        <range out="0.25"/>
    </function>
</weight>
<weight type="building:railway-station">
    <function>
        <range out="0.001"/>
    </function>
</weight>
<weight type="land-cover">
    <function>
        <range out="0.43"/>
    </function>
</weight>
<weight type="marker">
    <function>
        <range out="0.4"/>
    </function>
</weight>
<weight type="railway">
    <function>
        <range out="0.15"/>
    </function>
</weight>
<weight type="path">
    <function>
        <range out="0"/>
    </function>
</weight>
<weight type="road">
    <function>
        <range out="1"/>
    </function>
</weight>
<weight type="road:byway">
    <function>
        <range out="0.15"/>
    </function>
</weight>
<weight type="road:under-construction">
    <function>
        <range out="0.35"/>
    </function>
</weight>
<weight type="topology">
    <function>
        <range out="0.1"/>
    </function>
</weight>
<weight type="water">
    <function>
        <range out="0.1"/>
    </function>
</weight>
<selectuf format="map" type="*">
    <function parameter="scale">
        <range to="0" out="0.5"/>
        <range to="10000" out="0.5"/>
        <range to="50000" out="1"/>
        <range to="100000" out="1"/>
        <range to="1000000" out="0.8" rate="1"/>
```

```
        </function>
    </selectuf>
    <selectuf format="map" type="road">
        <function parameter="scale">
            <range to="0" out="0.6"/>
            <range to="25000" out="0.8"/>
            <range to="50000" out="1"/>
            <range to="100000" out="1"/>
            <range to="1000000" out="0.6" rate="1"/>
        </function>
    </selectuf>
        <goal resource="time">
        <function>
            <range out="50"/>
        </function>
    </goal>
</spec>

<spec name="navigatingonfoot">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
      </match>
    <match aspect="transport">
        <aspectvalue value="on-foot"/>
      </match>
    <weight type="coastline">
        <function>
            <range out="0.85"/>
        </function>
    </weight>
    <weight type="building">
        <function>
            <range out="0.75"/>
        </function>
    </weight>
    <weight type="building:railway-station">
        <function>
            <range out="0.85"/>
        </function>
    </weight>
    <weight type="land-use">
        <function>
            <range out="0.4"/>
        </function>
    </weight>
    <weight type="land-cover">
        <function>
            <range out="0.65"/>
        </function>
    </weight>
    <weight type="land-cover:boulder">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
    <weight type="land-cover:marsh">
        <function>
            <range out="0.9"/>
        </function>
    </weight>
    <weight type="land-cover:rock">
        <function>
            <range out="0.8"/>
        </function>
    </weight>
    <weight type="land-cover:scree">
        <function>
            <range out="0.9"/>
        </function>
```

```
</weight>
<weight type="land-cover:woodland">
    <function>
        <range out="0.75"/>
    </function>
</weight>
<weight type="railway">
    <function>
        <range out="0.9"/>
    </function>
</weight>
<weight type="path">
    <function>
        <range out="1"/>
    </function>
</weight>
<weight type="road">
    <function>
        <range out="0.8"/>
    </function>
</weight>
<weight type="road:motorway">
    <function>
        <range out="0.9"/>
    </function>
</weight>
<weight type="road:byway">
    <function>
        <range out="1"/>
    </function>
</weight>
<weight type="road:minor-road">
    <function>
        <range out="0.9"/>
    </function>
</weight>
<weight type="topology">
    <function>
        <range out="0.85"/>
    </function>
</weight>
<weight type="water">
    <function>
        <range out="0.9"/>
    </function>
</weight>
<selectuf format="map" type="*">
    <function parameter="scale">
        <range to="0" out="0.5"/>
        <range to="1000" out="0.7"/>
        <range to="25000" out="1"/>
        <range to="200000" out="1"/>
        <range to="1000000" out="0.6" rate="1"/>
     </function>
</selectuf>
<selectuf format="map" type="path">
    <function parameter="scale">
        <range to="0" out="0.6"/>
        <range to="5000" out="1"/>
        <range to="25000" out="1"/>
        <range to="50000" out="0.8"/>
        <range to="100000" out="0.5" rate="2"/>
        <range to="1000000" out="0.3" rate="3"/>
     </function>
</selectuf>
<selectuf format="map" type="road">
    <function parameter="scale">
        <range to="0" out="0.6"/>
        <range to="5000" out="1"/>
```

```
            <range to="50000" out="1"/>
            <range to="100000" out="0.7" rate="2"/>
            <range to="1000000" out="0.5" rate="3"/>
         </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="45"/>
        </function>
    </goal>
</spec>

<spec name="navigatingoncycle">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
      </match>
    <match aspect="transport">
        <aspectvalue value="cycling"/>
      </match>
    <weight type="coastline">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
    <weight type="building">
        <function>
            <range out="0.65"/>
        </function>
    </weight>
    <weight type="building:railway-station">
        <function>
            <range out="0.6"/>
        </function>
    </weight>
    <weight type="land-use">
        <function>
            <range out="0.4"/>
        </function>
    </weight>
    <weight type="land-cover">
        <function>
            <range out="0.5"/>
        </function>
    </weight>
    <weight type="land-cover:woodland">
        <function>
            <range out="0.6"/>
        </function>
    </weight>
    <weight type="railway">
        <function>
            <range out="0.65"/>
        </function>
    </weight>
    <weight type="path">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:motorway">
        <function>
            <range out="0.1"/>
        </function>
    </weight>
```

```
    <weight type="road:primary-route">
        <function>
            <range out="0.4"/>
        </function>
    </weight>
    <weight type="road:dual-carriageway">
        <function>
            <range out="0.4"/>
        </function>
    </weight>
    <weight type="road:a-road">
        <function>
            <range out="0.8"/>
        </function>
    </weight>
    <weight type="topology">
        <function>
            <range out="0.75"/>
        </function>
    </weight>
    <selectuf format="map" type="*">
        <function parameter="scale">
            <range to="0" out="0.5"/>
            <range to="5000" out="0.7"/>
            <range to="10000" out="1"/>
            <range to="250000" out="1"/>
            <range to="1000000" out="0.6" rate="1"/>
         </function>
    </selectuf>
    <selectuf format="map" type="road">
        <function parameter="scale">
            <range to="0" out="0.6"/>
            <range to="5000" out="0.7"/>
            <range to="25000" out="1"/>
            <range to="100000" out="1" rate="1"/>
            <range to="250000" out="0.8" rate="1"/>
            <range to="1000000" out="0.5" rate="2"/>
         </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="30"/>
        </function>
    </goal>
</spec>
</specset>

<specset typedomain="map">
<spec name="navigatingslow">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
      </match>
    <match aspect="speed">
        <aspectrange min="0" max="15"/>
    </match>
    <weight type="*">
        <function parameter="distance">
            <range to="500" out="1" rate="0"/>
            <range to="1000" out="0.9" rate="0"/>
            <range to="2000" out="0.7" rate="0"/>
            <range to="5000" out="0.3" rate="0"/>
            <range to="10000" out="0.1" rate="0"/>
            <range to="20000" out="0" rate="0"/>
        </function>
    </weight>
    <weight type="road">
        <function parameter="distance">
            <range to="500" out="1" rate="0"/>
            <range to="1000" out="0.95" rate="0"/>
```

```
            <range to="2000" out="0.35" rate="0"/>
            <range to="5000" out="0.1" rate="0"/>
        </function>
    </weight>
    <weight type="road:minor-road">
        <function parameter="distance">
            <range to="500" out="1" rate="0"/>
            <range to="1000" out="0.92" rate="0"/>
            <range to="2000" out="0.25" rate="0"/>
            <range to="5000" out="0" rate="0"/>
        </function>
    </weight>
    <selectuf format="map" type="*">
        <function parameter="scale">
            <range to="1000" out="0.5"/>
            <range to="10000" out="1" rate="1"/>
            <range to="50000" out="1" rate="1"/>
            <range to="100000" out="0.7" rate="1"/>
            <range to="1000000" out="0.3" rate="1"/>
         </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="30"/>
        </function>
    </goal>
    <goal resource="screendensity">
        <function>
            <range out="2.5"/>
        </function>
    </goal>
</spec>

<spec name="navigatingbrisk">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
      </match>
    <match aspect="speed">
        <aspectrange min="16" max="45"/>
    </match>
    <weight type="*">
        <function parameter="distance">
            <range to="1000" out="1" rate="0"/>
            <range to="2000" out="0.9" rate="0"/>
            <range to="5000" out="0.7" rate="0"/>
            <range to="10000" out="0.3" rate="0"/>
            <range to="20000" out="0" rate="0"/>
        </function>
    </weight>
    <weight type="road">
        <function parameter="distance">
            <range to="1000" out="1" rate="0"/>
            <range to="2000" out="0.76" rate="0"/>
            <range to="5000" out="0.35" rate="0"/>
            <range to="10000" out="0.1" rate="0"/>
        </function>
    </weight>
    <weight type="road:minor-road">
        <function parameter="distance">
            <range to="500" out="1" rate="0"/>
            <range to="1000" out="0.94" rate="0"/>
            <range to="2000" out="0.5" rate="0"/>
            <range to="4000" out="0.1" rate="0"/>
            <range to="8000" out="0" rate="0"/>
        </function>
    </weight>
    <selectuf format="map" type="*">
        <function parameter="scale">
            <range to="1000" out="0.3"/>
```

```
            <range to="25000" out="0.7" rate="1"/>
            <range to="50000" out="1" rate="1"/>
            <range to="100000" out="1" rate="1"/>
            <range to="1000000" out="0.5" rate="1"/>
        </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="25"/>
        </function>
    </goal>
    <goal resource="screendensity">
        <function>
            <range out="2"/>
        </function>
    </goal>
</spec>

<spec name="navigatingfast">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
      </match>
    <match aspect="speed">
        <aspectrange min="46" max="200"/>
    </match>
    <weight type="*">
        <function parameter="distance">
            <range to="2000" out="1" rate="0"/>
            <range to="5000" out="0.9" rate="0"/>
            <range to="10000" out="0.7" rate="0"/>
            <range to="20000" out="0.3" rate="0"/>
            <range to="50000" out="0" rate="0"/>
         </function>
    </weight>
    <weight type="road">
        <function parameter="distance">
            <range to="1000" out="1" rate="0"/>
            <range to="2000" out="0.86" rate="0"/>
            <range to="5000" out="0.4" rate="0"/>
            <range to="10000" out="0.1" rate="0"/>
        </function>
    </weight>
    <weight type="road:minor-road">
        <function parameter="distance">
            <range to="500" out="1" rate="0"/>
            <range to="1000" out="0.88" rate="0"/>
            <range to="2000" out="0.7" rate="0"/>
            <range to="5000" out="0.2" rate="0"/>
            <range to="10000" out="0" rate="0"/>
        </function>
    </weight>
    <selectuf format="map" type="*">
        <function parameter="scale">
            <range to="25000" out="0.3" rate="1"/>
            <range to="50000" out="0.8" rate="1"/>
            <range to="100000" out="1" rate="1"/>
    </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="20"/>
        </function>
    </goal>
    <goal resource="screendensity">
        <function>
            <range out="1.2"/>
        </function>
    </goal>
</spec>
```

```
<spec name="carslow">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
      </match>
    <match aspect="transport">
        <aspectvalue value="in-car" default="true"/>
      </match>
    <match aspect="speed">
        <aspectrange min="0" max="20"/>
    </match>

    <weight type="road">
        <function>
            <range out="0.8"/>
        </function>
    </weight>
    <weight type="road:a-road">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:b-road">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:byway">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
    <weight type="road:dual-carriageway">
        <function>
            <range out="0.92"/>
        </function>
    </weight>
    <weight type="road:motorway">
        <function>
            <range out="0.9"/>
        </function>
    </weight>
    <weight type="road:minor-road">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:primary-route">
        <function>
            <range out="0.95"/>
        </function>
    </weight>
    <weight type="building:number-name">
        <function>
            <range out="0.8"/>
        </function>
    </weight>
    <weight type="road">
        <function parameter="distance">
            <range to="500" out="1" rate="0"/>
            <range to="1000" out="0.95" rate="0"/>
            <range to="2000" out="0.75" rate="0"/>
            <range to="5000" out="0.1" rate="0"/>
            <range to="10000" out="0" rate="0"/>
         </function>
    </weight>
    <weight type="road:minor-road">
        <function parameter="distance">
            <range to="300" out="1" rate="0"/>
```

```
            <range to="1000" out="0.9" rate="0"/>
            <range to="2000" out="0.7" rate="0"/>
            <range to="5000" out="0" rate="0"/>
        </function>
    </weight>
    <weight type="road:b-road">
        <function parameter="distance">
            <range to="500" out="1" rate="0"/>
            <range to="1000" out="0.91" rate="0"/>
            <range to="2000" out="0.7" rate="0"/>
            <range to="5000" out="0.15" rate="0"/>
            <range to="10000" out="0" rate="0"/>
        </function>
    </weight>
    <weight type="road:a-road">
        <function parameter="distance">
            <range to="500" out="1" rate="0"/>
            <range to="1000" out="0.91" rate="0"/>
            <range to="2000" out="0.71" rate="0"/>
            <range to="5000" out="0.2" rate="0"/>
            <range to="10000" out="0" rate="0"/>
        </function>
    </weight>
    <selectuf format="map" type="road">
        <function parameter="scale">
            <range to="1000" out="0.4"/>
            <range to="10000" out="0.7" rate="1"/>
            <range to="25000" out="1" rate="1"/>
            <range to="100000" out="1" rate="1"/>
            <range to="1000000" out="0.6" rate="1"/>
        </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="30"/>
        </function>
    </goal>
    <goal resource="screendensity">
        <function>
            <range out="2"/>
        </function>
    </goal>
</spec>

<spec name="carbrisk">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
    </match>
    <match aspect="transport">
        <aspectvalue value="in-car" default="true"/>
    </match>
    <match aspect="speed">
        <aspectrange min="21" max="50"/>
    </match>
    <weight type="road">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:a-road">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:b-road">
        <function>
            <range out="1"/>
        </function>
    </weight>
```

```
<weight type="road:byway">
    <function>
        <range out="0.5"/>
    </function>
</weight>
<weight type="road:dual-carriageway">
    <function>
        <range out="1"/>
    </function>
</weight>
<weight type="road:motorway">
    <function>
        <range out="0.9"/>
    </function>
</weight>
<weight type="road:minor-road">
    <function>
        <range out="0.6"/>
    </function>
</weight>
<weight type="road:primary-route">
    <function>
        <range out="0.9"/>
    </function>
</weight>
<weight type="building:number-name">
    <function>
        <range out="0.7"/>
    </function>
</weight>
<weight type="road">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.9" rate="0"/>
        <range to="10000" out="0.7" rate="0"/>
        <range to="20000" out="0.3" rate="0"/>
        <range to="50000" out="0" rate="0"/>
    </function>
</weight>
<weight type="road:minor-road">
    <function parameter="distance">
        <range to="400" out="1" rate="0"/>
        <range to="1000" out="0.9" rate="0"/>
        <range to="5000" out="0.7" rate="0"/>
        <range to="10000" out="0" rate="0"/>
    </function>
</weight>
<weight type="road:a-road">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.92" rate="0"/>
        <range to="10000" out="0.82" rate="0"/>
        <range to="20000" out="0.42" rate="0"/>
        <range to="50000" out="0" rate="0"/>
    </function>
</weight>
<weight type="road:b-road">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.92" rate="0"/>
        <range to="10000" out="0.8" rate="0"/>
        <range to="20000" out="0.4" rate="0"/>
        <range to="50000" out="0" rate="0"/>
    </function>
</weight>
<weight type="road:dual-carriageway">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.92" rate="0"/>
```

```
            <range to="10000" out="0.82" rate="0"/>
            <range to="20000" out="0.42" rate="0"/>
            <range to="50000" out="0" rate="0"/>
        </function>
    </weight>
    <selectuf format="map" type="road">
        <function parameter="scale">
            <range to="1000" out="0.4"/>
            <range to="10000" out="0.6" rate="1"/>
            <range to="25000" out="1" rate="1"/>
            <range to="100000" out="1" rate="1"/>
            <range to="1000000" out="0.6" rate="1"/>
         </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="20"/>
        </function>
    </goal>
    <goal resource="screendensity">
        <function>
            <range out="1.4"/>
        </function>
    </goal>
</spec>

<spec name="carfast">
    <match aspect="mapuse">
        <aspectvalue value="navigating" default="true"/>
      </match>
    <match aspect="transport">
        <aspectvalue value="in-car" default="true"/>
      </match>
    <match aspect="speed">
        <aspectrange min="51" max="200"/>
    </match>
    <weight type="road">
        <function>
            <range out="0.7"/>
        </function>
    </weight>
    <weight type="road:a-road">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:b-road">
        <function>
            <range out="0.95"/>
        </function>
    </weight>
    <weight type="road:byway">
        <function>
            <range out="0"/>
        </function>
    </weight>
    <weight type="road:dual-carriageway">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:motorway">
        <function>
            <range out="1"/>
        </function>
    </weight>
    <weight type="road:minor-road">
        <function>
            <range out="0.6"/>
```

```
    </function>
</weight>
<weight type="road:primary-route">
    <function>
        <range out="1"/>
    </function>
</weight>
<weight type="building:number-name">
    <function>
        <range out="0"/>
    </function>
</weight>
<weight type="road">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.9" rate="0"/>
        <range to="10000" out="0.8" rate="0"/>
        <range to="20000" out="0.6" rate="0"/>
        <range to="50000" out="0" rate="0"/>
    </function>
</weight>
<weight type="road:minor-road">
    <function parameter="distance">
        <range to="200" out="1" rate="0"/>
        <range to="1000" out="0.8" rate="0"/>
        <range to="2000" out="0.4" rate="0"/>
        <range to="5000" out="0.05" rate="0"/>
        <range to="10000" out="0" rate="0"/>
   </function>
</weight>
<weight type="road:a-road">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.92" rate="0"/>
        <range to="10000" out="0.82" rate="0"/>
        <range to="25000" out="0.72" rate="0"/>
        <range to="50000" out="0" rate="0"/>
       </function>
</weight>
<weight type="road:motorway">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.925" rate="0"/>
        <range to="10000" out="0.825" rate="0"/>
        <range to="30000" out="0.725" rate="0"/>
        <range to="50000" out="0" rate="0"/>
      </function>
</weight>
<weight type="road:dual-carriageway">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.91" rate="0"/>
        <range to="10000" out="0.81" rate="0"/>
        <range to="25000" out="0.71" rate="0"/>
        <range to="50000" out="0" rate="0"/>
      </function>
</weight>
<weight type="road:primary-route">
    <function parameter="distance">
        <range to="2000" out="1" rate="0"/>
        <range to="5000" out="0.915" rate="0"/>
        <range to="10000" out="0.815" rate="0"/>
        <range to="25000" out="0.715" rate="0"/>
        <range to="50000" out="0" rate="0"/>
      </function>
</weight>
<selectuf format="map" type="road">
    <function parameter="scale">
        <range to="1000" out="0.4"/>
```

```
            <range to="25000" out="0.7" rate="1"/>
            <range to="50000" out="1" rate="1"/>
            <range to="200000" out="1" rate="1"/>
            <range to="1000000" out="0.8" rate="1"/>
        </function>
    </selectuf>
    <goal resource="time">
        <function>
            <range out="20"/>
        </function>
    </goal>
    <goal resource="screendensity">
        <function>
            <range out="1.1"/>
        </function>
    </goal>
</spec>
</specset>
```

APPENDIX 3 MAP SERVER AND MAP DATA

A3.1 INTRODUCTION

We shall give an overview of the implementation of the map server here. The server is implemented as a Java servlet, this parses the NTF data files on start-up and stores the data in element and variant objects. Lines are further processed to remove detail from the sequence of points, in order to offer higher scale variants. The elements are stored in an RTree to enable efficient look-up. The storage is not highly optimised and future versions may make use of a spatial database. Requests can be for meta data or map data, we describe each below.

A meta data request defines an area of interest and returns a description in XML of the elements and variants in that area. Variants of features are described as being contained by a variant of the header data from the source. More than one header variant may be included in any given request. The header includes origin coordinates, from which feature data describes an offset. The element and variant names are generated from the object hashes to ensure uniqueness, even where multiple data sources might use the same feature codes within the data.

The meta data offered is stored in a hash, to save a more complex search for requested variants when a data request is made. The hash key is the name of the header element, which is derived from the area the data requested is for. This key is used again by the client when data is requested. Requests are periodically removed from the hash, to allow the server to garbage collect the objects and maintain a rapid request look-up.

GZipped or uncompressed data may be requested from the server. Uncompressed data may be useful where the client has limited processor capability to perform decompression, or for debugging. The compression happens on-the-fly, using a compressing stream object in Java. This is necessary to enable a dynamic composition of meta data for an area or data as requested. The fact that the meta data describes the uncompressed size inaccurately is not viewed as a problem – the throughput measurement takes into account average compression. Where streams to a site would be differently amenable to compression, e.g. for a web site with HTML and JPEG images, different throughput models would need to be maintained (or compression abandoned) in order for this approach to work. There is a disadvantage, in that the compression incurs processor overhead on the server and causes additional delay.

A data request provides the area description again, which enables a rapid look-up from recent requests of offered variant objects, the variant names required are then listed. The data from the variant objects described is extracted and combined into a single response. The first variant for each request should be the variant describing the relevant tile header data, this is the only level of structure implemented in the current server (due to limitations in the NTF format, which describes road names with coordinates and does not establish clear links in data with some other potentially structured features). Variants from multiple tiles are requested in multiple (possibly parallel) HTTP GETs. This is important as the servlet does not allow the GET string to exceed 1kB. Where a large number of variants are to be requested multiple requests with the same header have to be made.

A3.2 META DATA GET

An example URL for getting meta data is:

```
http://www-dse.doc.ic.ac.uk/servlets/mapservlet/
mapsrv?&rtype=meta&zip=false&xmin=529500&ymin=177000&xmax=530499&ymax=177999
```

This has the form:

A request to the map servlet: `http://www-dse.doc.ic.ac.uk/servlets/mapservlet/mapsrv?`

Request of meta data: `&rtype=meta`

Don’t use compression: `&zip=false`

A description of the area of interest, in Ordnance Survey grid coordinates: `&xmin=529500&ymin=177000&xmax=530499&ymax=177999`

A3.3 DATA GET

An example URL for getting map data is:

`http://www-dse.doc.ic.ac.uk/servlets/mapservlet/mapsrv?&rtype=data&zip=true&xmin=520001&ymin=180000&xmax=530000&ymax=190000&variants=TQ28,25579ce1,5ab17a32`

This has the form:

A request to the map servlet: `http://www-dse.doc.ic.ac.uk/servlets/mapservlet/mapsrv?`

Request of map data: `&rtype=data`

Use compression: `&zip=true`

A description of the area of interest, which is used to retrieve the details of the variants offered from a hash-table, rather than having to find them by searching the full set of data: `&xmin=520001&ymin=180000&xmax=530000&ymax=190000`.

A comma separated list of the variants required, starting with the containing variant which specifies the tile header data: `&variants=TQ28,25579ce1,5ab17a32` Here just two map data variants are being requested.

A3.4 EXAMPLE NTF DATA

For a detailed explanation of the data format the standard documents and manuals for delivered data should be consulted. The description and example data given here are given to illustrate the data in use.

First in the data are the header sections. These describe the source of the data, the location in the grid of a reference point (from which all other coordinates are given for brevity) and the feature codes which may be found in the data.

```
01ORDNANCE SURVEY                        2000090100000130200V \0%
02Meridian_02.00      DEFAULT_02.00       19920515                    000000001%
00Meridian_02.00      20000901                    00000000000%
40FC004I4    FEATURE CODE\NUMERIC FEATURE CODE\0%
40OD013A13   OSODR\ORDNANCE SURVEY OSCAR DATA REFERENCE\0%
40JN   A*    JUNCTION_NAME\Name Of Road Junction\0%
40LL005I5    LINE_LENGTH\Length Of Line Feature\0%
40NP002I2    NUM_PARENTS\Number Of Parent OSODRs\0%
40PO013A13   PARENT_OSODR\Parent OSODR\0%
40RT001A1    ROUNDABOUT\Roundabout Indicator\0%
40RN   A*    ROAD_NUM\DoT Route Number\0%
40SN   A*    SETTLEMENT_NAME\Settlement Name\0%
40TR001A1    TRUNK_ROAD\Trunk Road Indicator\0%
40LC006I6    LEFT COUNTY\LEFT COUNTY INDICATOR\0%
40RC006I6    RIGHT COUNTY\RIGHT COUNTY INDICATOR\0%
40LD006I6    LEFT DISTRICT\LEFT DISTRICT INDICATOR\0%
40RD006I6    RIGHT DISTRICT\RIGHT DISTRICT INDICATOR\0%
40NM   A*    ADMIN NAME\ADMINISTRATIVE AREA NAME\0%
40PI006I6    GLOBAL ID\ADMIN AREA GLOBAL IDENTIFIER\0%
40DA013A13   DLUA ID\DLUA IDENTIFIER\0%
40PN   A*    PROPER NAME\DEFINITIVE NAME\0%
40RI013A13   RAIL ID\RAILWAY IDENTIFIER\0%
```

```
40SI013A13  STATION ID\STATION IDENTIFIER\0%
40TX   A*   TEXT\INDEPENDENT TEXT\0%
40FA   A13  FOREST ID\FOREST IDENTIFIER\0%
40WA   A13  WATER AREA ID\AREA WATER IDENTIFIER\0%
40WI   A13  WATER LINK ID\LINEAR WATER IDENTIFIER\0%
40HT008I8   HEIGHT\DERIVED HEIGHT IN METRES\0%
053000                        Motorway\0%
053001                        A-road\0%
053002                        B-road\0%
053004                        Minor road\0%
053500                        Road node, unspecified\0%
053501                        Road edge node, unspecified\0%
056401                        County boundary\0%
056403                        District boundary\0%
056405                        County/district boundary\0%
056411                        County seed-point\0%
056415                        District seed-point\0%
056710                        Boundary node, unspecified\0%
056711                        Boundary edge node\0%
056800                        Admin area (neat line)\0%
056200                        Coast\0%
056740                        Coast node, unspecified\0%
056741                        Coast edge node, unspecified\0%
056300                        DLUA Boundary\0%
056310                        DLUA seed-point\0%
056720                        DLUA node, unspecified\0%
056721                        DLUA edge node, unspecified\0%
056801                        DLUA (neat line)\0%
056140                        Railway\0%
056142                        Railway (tunnelled)\0%
056155                        Station, point\0%
056731                        Railway edge node\0%
056730                        Railway node\0%
056551                        Station name\0%
056500                        Names\0%
056223                        Water feature (river small)\0%
056224                        Water feature (river medium)\0%
056225                        Water feature (river large)\0%
056230                        Water feature (hidden)\0%
056231                        Water feature (aqueduct)\0%
056232                        Water feature (dark link)\0%
056243                        Water feature (canal)\0%
056255                        Area water (lake)\0%
056292                        Area water (seed)\0%
056770                        Water feature (node)\0%
056771                        Area water (node)\0%
056772                        Area water (edge node)\0%
056773                        Water feature (edge node)\0%
056803                        Area water (neat line)\0%
056552                        Water Text\0%
056664                        Woodland Boundary\0%
056663                        Woodland seed\0%
056802                        Woodland (neat line)\0%
056750                        Woodland node\0%
056751                        Woodland edge node\0%
056762                        Height Point feature\0%
07TQ29      21000052000000100000000000000001000000052000000001900000000000001%
000000000000000000000000010000000001000000000000000000000020000901200009010%
```

Next comes a series of feature descriptions. Each feature has a reference number and coordinate information, plus optional data such as a road name, references to other feature databases etc. We see here four features. Lines starting “21” give one or more coordinates for the feature. Lines starting “00” continue the previous line. The data is formatted for an 80 column output.

```
23002461002461010024610%
```

```
21002461200030031300594 0044500626 0069700427 0%
14002461ODO3AVV4KJ5G2YVFC3004LL00464PNSILKSTREAM ROAD\NP01POO1AVV4KJ5G2YV0%
2300595600595601005956 0%
21005956200100866002223 0872502120 0876702058 0880601999 0887801913 08926018491%
00 0897301796 0902201743 0909001678 0922301551 0%
14005956FC6140RI52890201918810%
2302009402009401020094 0%
21020094200130053810000 0054809982 0057809933 0062609860 0072509702 00761096241%
00 0077509570 0079509518 0082209461 0084309411 0086509363 0090809271 009261%
0009225 0%
14020094FC6140RI53076801895970%
2300857700857701008577 0%
21008577200190027901539 0034501662 0036301698 0041601801 0045801881 00498019531%
00 0056202075 0058102113 0059902145 0062602198 0065002243 0068902320 007201%
0002380 0075402445 0079102516 0086802664 0088802701 0089602719 0095002819 0%
14008577FC6140RI53062601921980%
```

Finally the data has an end record. We treat this as part of the header data in the variant and request structure.

```
99End Of Transfer Set0%
```

Appendix 3 - Map Server and Map Data

APPENDIX 4 COMPONENT MODEL AND SOFTWARE ENGINEERING NOTES

A4.1 COMPONENT STRUCTURE AND LOADING

A basic component model was developed in order to simplify logical separation of components in the system, and facilitate re-configuration of the application e.g. to change selection algorithms or add location sources. The model used an XML configuration language inspired by Darwin [86], and Java reflection in its implementation. Run-time re-configuration and distribution were not addressed. This is outlined below.

A4.1.1 Components, Interfaces, Instances, Parameters and Bindings

A component was defined by a class, used to instantiate the component. No definition of which other classes are included in the component is used. The component is expected to have a no-attribute constructor, and implement the ComponentI interface. This interface simply provides a close method for termination. It could be used for other control if the model were to be extended. The component is given a name used in references to it within the configuration. e.g.:

```
<component
class="uk.ac.ic.doc.dc.location.LocManager"
name="locmgr" />
```

Interfaces to the component are defined by identifying methods in the component class. An interface consists of a class identification, which is the component to be bound. There may also be a parameter to the interface, defined as for parameters below. Parameters of interfaces allow multiple bindings to an interface to be differentiated. The interface is named. e.g.:

```
<interface
component="locmgr"
name="setViewer”
class="uk.ac.ic.doc.dc.viewer.TabbedPaneViewer
"method="setViewer"
paramtype="string" />
```

Instances of components may be defined and named. e.g.:

```
<instance
component="locmgr"
name="locmgr"/>
```

Parameters may be set on instances. Parameters are integer, double, string or boolean. These provide for setting configuration values without recompilation or modification of the code. e.g.:

```
<param
instance="locsvc-placeui"
method="parseLocationData"
type="String"
value="etc/location/OSGRPlaceSelector.data" />
```

Binding connects an instance and another instance or a component. Binding between two instances provides third-party binding, with the requiror object being passed the providor object. e.g.:

```
<bind
requiror="locmgr"
interface="locmgr/addLocService"
bindType=”static”
providor="locsvc-placeui"
paramvalue="Location Selection" />
```

Note that this binding has an associated parameter, used to differentiate between multiple bindings to the same interface on a requiror. The first-party component object (requiror) can then invoke methods on the second-party component object (providor). If the binding should provide for the second-party object to be able to call the first-party then a second binding is required.

A binding between an instance and a component provides the first-party instance with the class of the second-party component. First-party instantiation and binding may then be performed as needed. Dynamic instantiation is therefore supported. e.g.:

```
<bind
requiror="mapappqos"
interface="mapappqos/setMetaHandler"
bindType=”dynamic”
providor="mapMetaHandler" />
```

(mapappqos is an instance name, mapMetaHandler is a component name).

Both types of binding can take a parameter (as defined in the interface). The binding assumes that the Java method is typed correctly, using the class given in the interface (or a sub-type of this). The interface class definition is not passed into the component.

The configuration file is parsed linearly, and calls made as encountered. Where instantiations and/or bindings and/or parameter settings have dependencies these must be identified and accounted for in the ordering of the definitions in the configuration file.

The component loader can be bound to, using an instance name of “this”. The component loader implements ComponentI, so providing a close method. This then calls the close method on each instance loaded. This allows a clean shut-down of the application. Dynamically loaded components are the responsibility of the creating party.

A4.1.2 Configuration Language

XML is used to encode the configuration (and other data described in this chapter) for convenience in parsing. The XML schema and an example of the encoded data are given below.

A4.1.2.1Configuration Language Schema

```
<?xml version=”1.0” encoding=”UTF-8”?>
<xs:schema xmlns:xs=”http://www.w3.org/2001/XMLSchema” elementFormDefault=”qualified”>
    <xs:complexType name=”componentType”>
        <xs:attribute name=”class” type=”xs:string” use=”required”/>
        <xs:attribute name=”name” type=”xs:string” use=”required”/>
    </xs:complexType>

    <xs:complexType name=”instanceType”>
        <xs:attribute name=”component” type=”xs:string” use=”required”/>
        <xs:attribute name=”name” type=”xs:string” use=”required”/>
    </xs:complexType>

    <xs:complexType name=”interfaceType”>
        <xs:attribute name=”component” type=”xs:string” use=”required”/>
        <xs:attribute name=”name” type=”xs:string” use=”required”/>
        <xs:attribute name=”class” type=”xs:string” use=”required”/>
        <xs:attribute name=”method” type=”xs:string” use=”required”/>
        <xs:attribute name=”paramtype” type=”xs:string”/>
    </xs:complexType>

    <xs:complexType name=”bindType”>
        <xs:attribute name=”requiror” type=”xs:string” use=”required”/>
        <xs:attribute name=”interface” type=”xs:string” use=”required”/>
        <xs:attribute name=”bindType” use=”required”>
            <xs:simpleType>
                <xs:restriction base=”xs:NMTOKEN”>
                    <xs:enumeration value=”dynamic”/>
                    <xs:enumeration value=”static”/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name=”providor” type=”xs:string” use=”required”/>
        <xs:attribute name=”paramvalue” type=”xs:string”/>
    </xs:complexType>

    <xs:complexType name=”paramType”>
        <xs:attribute name=”instance” type=”xs:string” use=”required”/>
        <xs:attribute name=”method” type=”xs:string” use=”required”/>
        <xs:attribute name=”type”>
```

```
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="boolean"/>
                    <xs:enumeration value="double"/>
                    <xs:enumeration value="int"/>
                    <xs:enumeration value="string"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="value" type="xs:string"/>
    </xs:complexType>

    <xs:element name="configuration">
        <xs:complexType>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="component" type="componentType"/>
                <xs:element name="instance" type="instanceType"/>
                <xs:element name="param" type="paramType"/>
                <xs:element name="bind" type="bindType"/>
                <xs:element name="interface" type="interfaceType"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

A4.1.2.2Configuaration Example

The following configuration is typical of those used in testing in the map application.

```
<configuration>

<component class="uk.ac.ic.doc.dc.logWriter.LogWriter" name="logwriter" />
<instance  component="logwriter" name="logwriter"/>
<param  instance="logwriter" method="setLevel" type="int" value="3" />
<param  instance="logwriter" method="setFile" type="String" value="logs/log" />

<component class="uk.ac.ic.doc.dc.mediaselection.typeSystem.TypeParser" name="types" />
<instance  component="types" name="types"/>
<param  instance="types"method="parseDefData"type="String"value="etc/typedefs/typesys-0" /
>
<param  instance="types"method="parseDefData"type="String"value="etc/typedefs/typesys-map"
/>

<component class="uk.ac.ic.doc.dc.viewer.SinglePaneViewer" name="viewershell" />
<interface component="viewershell" name="setLoader"
class="uk.ac.ic.doc.dc.componentLoader.ComponentI"method="setLoader" />
<instance component="viewershell" name="viewers"/>
<param  instance="viewers" method="setName" type="String" value="Viewer Window" />
<bind   requiror="viewers" interface="viewershell/setLoader" bindType="static"
providor="this" />

<component class="uk.ac.ic.doc.dc.viewer.TabbedPaneViewer" name="controlshell" />
<instance component="controlshell" name="controls"/>
<param  instance="controls" method="setName" type="String" value="Control Window" />

<component class="uk.ac.ic.doc.dc.location.LocManager" name="locmgr" />
<interface component="locmgr" name="setViewer"
class="uk.ac.ic.doc.dc.viewer.TabbedPaneViewer"method="setViewer" paramtype="string" />
<interface component="locmgr" name="addLocService"
class="uk.ac.ic.doc.dc.location.LocationServiceI"method="addLocService" paramtype="string"
/>
<interface component="locmgr" name="addLocationListener"
class="uk.ac.ic.doc.dc.location.LocationServiceListener" method="addLocationListener" />
<instance component="locmgr" name="locmgr"/>
<bind   requiror="locmgr"  interface="locmgr/setViewer" bindType="static"
providor="controls" paramvalue="Location Manager" />
```

```
<component class=”uk.ac.ic.doc.dc.location.OSGRPlaceSelector” name=”locsvc-placeui” />
<instance component=”locsvc-placeui” name=”locsvc-placeui”/>
<param instance=”locsvc-placeui” method=”parseLocationData” type=”String” value=”etc/
location/OSGRPlaceSelector.data” />
<bind requiror=”locmgr” interface=”locmgr/addLocService” bindType=”static”
providor=”locsvc-placeui” paramvalue=”Location Selection” />

<component class=”uk.ac.ic.doc.dc.location.TraceFilePlayer” name=”locsvc-traceplayer” />
<instance component=”locsvc-traceplayer” name=”locsvc-traceplayer”/>
<param instance=”locsvc-traceplayer”method=”setLocationDataFile” type=”String”
value=”etc/location/” />
<bind requiror=”locmgr” interface=”locmgr/addLocService” bindType=”static”
providor=”locsvc-traceplayer” paramvalue=”Location Trace” />

<component class=”uk.ac.ic.doc.dc.mediaselection.playout.Playout” name=”playout” />
<interface component=”playout” name=”setViewer”
class=”uk.ac.ic.doc.dc.viewer.TabbedPaneViewer”method=”setViewer” paramtype=”string” />
<interface component=”playout” name=”setLogWriter”
class=”uk.ac.ic.doc.dc.logWriter.LogWriter”method=”setLogWriter” />
<instance component=”playout” name=”playout”/>
<param instance=”playout” method=”setInitDelay” type=”int” value=”100” />
<param instance=”playout” method=”setRate” type=”int” value=”3500” />
<param instance=”playout” method=”setDropRate” type=”double” value=”0” />
<param instance=”playout” method=”setMultiDropRate” type=”double” value=”0” />
<param instance=”playout” method=”setDropDelay” type=”int” value=”0” />
<param instance=”playout” method=”setTraceFile” type=”string” value=”etc/playout” />
<bind requiror=”playout” interface=”playout/setViewer” bindType=”static”
providor=”controls” paramvalue=”Playout” />
<bind requiror=”playout” interface=”playout/setLogWriter” bindType=”static”
providor=”logwriter” />
<param instance=”playout”method=”startPlayout” />

<component
class=”uk.ac.ic.doc.dc.mediaselection.resourceMgr.bwMgr.SiteBandwidthResourceMgr”
name=”bwmgr” />
<interface component=”bwmgr” name=”setViewer”
class=”uk.ac.ic.doc.dc.viewer.TabbedPaneViewer”method=”setViewer” paramtype=”string” />
<interface component=”bwmgr” name=”setLogWriter”
class=”uk.ac.ic.doc.dc.logWriter.LogWriter”method=”setLogWriter” />
<instance component=”bwmgr” name=”bwmgr”/>
<param instance=”bwmgr” method=”setSlotLength” type=”int” value=”100” />
<param instance=”bwmgr” method=”setSlotCapacity” type=”int” value=”1280” />
<param instance=”bwmgr” method=”setRoundTripTime” type=”int” value=”200” />
<param instance=”bwmgr” method=”setBWJumpSensitivity” type=”double” value=”1.094” />
<param instance=”bwmgr” method=”setBWUpdateRatio” type=”double” value=”0.25” />
<param instance=”bwmgr” method=”setBWHistoryLength” type=”int” value=”3” />
<param instance=”bwmgr” method=”setRTTJumpSensitivity” type=”double” value=”1.094” />
<param instance=”bwmgr” method=”setRTTUpdateRatio” type=”double” value=”0.25” />
<param instance=”bwmgr” method=”setRTTHistoryLength” type=”int” value=”3” />
<bind requiror=”bwmgr” interface=”bwmgr/setViewer” bindType=”static”
providor=”controls” paramvalue=”Network Resource Manager” />
<bind requiror=”bwmgr” interface=”bwmgr/setLogWriter” bindType=”static”
providor=”logwriter” />

<component class=”uk.ac.ic.doc.dc.mediaselection.resourceMgr.screenMgr.MapScreenMgr”
name=”screenmgr” />
<interface component=”screenmgr” name=”setViewer”
class=”uk.ac.ic.doc.dc.viewer.TabbedPaneViewer”method=”setViewer” paramtype=”string” />
<interface component=”screenmgr” name=”setLogWriter”
class=”uk.ac.ic.doc.dc.logWriter.LogWriter”method=”setLogWriter” />
<interface component=”screenmgr” name=”bindToScreen”
class=”uk.ac.ic.doc.dc.mediaselection.appUIs.mapApp.MapApp”method=”bindToScreen” />
<interface component=”screenmgr” name=”setTypes”
class=”uk.ac.ic.doc.dc.mediaselection.typeSystem.TypeParser” method=”setTypes” />
<instance component=”screenmgr” name=”screenmgr” />
<param instance=”screenmgr” method=”setDensityAllowed” type=”double” value=”0.9” />
<bind requiror=”screenmgr” interface=”screenmgr/setViewer” bindType=”static”
providor=”controls” paramvalue=”Screen Resource Manager” />
```

```
<bind   requiror="screenmgr"   interface="screenmgr/setLogWriter" bindType="static"
providor="logwriter"  />
<bind   requiror="screenmgr" interface="screenmgr/setTypes" bindType="static"
providor="types" />

<component class="uk.ac.ic.doc.dc.mediaselection.specMgr.contextMgr.ContextMgr"
name="contextmgr" />
<interface component="contextmgr" name="setViewer"
class="uk.ac.ic.doc.dc.viewer.TabbedPaneViewer"method="setViewer" paramtype="string" />
<instance component="contextmgr" name="contextmgr"/>
<param instance="contextmgr"method="parseContextDefs"type="String"value="etc/context/
ContextDefs.xml" />
<bind   requiror="contextmgr" interface="contextmgr/setViewer" bindType="static"
providor="controls" paramvalue="Context Manager" />

<component class="uk.ac.ic.doc.dc.mediaselection.specMgr.SpecMgr" name="specmgr" />
<interface component="specmgr" name="setViewer"
class="uk.ac.ic.doc.dc.viewer.TabbedPaneViewer"method="setViewer" paramtype="string" />
<interface component="specmgr" name="setContextMgr"
class="uk.ac.ic.doc.dc.mediaselection.specMgr.contextMgr.ContextMgr"
method="setContextMgr" />
<interface component="specmgr" name="setTypes"
class="uk.ac.ic.doc.dc.mediaselection.typeSystem.TypeParser" method="setTypes" />
<instance component="specmgr" name="specmgr"/>
<bind   requiror="specmgr" interface="specmgr/setContextMgr" bindType="static"
providor="contextmgr" />
<bind   requiror="specmgr" interface="specmgr/setTypes" bindType="static" providor="types"
/>
<param instance="specmgr"method="parseSpecDefs"type="String"value="etc/specs/
basespecs.xml" />
<param instance="specmgr"method="parseSpecDefs"type="String"value="etc/specs/
mapbasespecs.xml" />
<param instance="specmgr"method="parseSpecDefs"type="String"value="etc/specs/
maptaskspecs.xml" />
<param instance="specmgr"method="parseSpecDefs"type="String"value="etc/specs/
maptransport.xml" />
<param instance="specmgr"method="parseSpecDefs"type="String"value="etc/specs/
pacespecs.xml" />
<bind   requiror="specmgr" interface="specmgr/setViewer" bindType="static"
providor="controls" paramvalue="Spec Manager" />

<component class="uk.ac.ic.doc.dc.mediaselection.appUIs.mapApp.MapApp" name="mapapp" />
<interface component="mapapp" name="setViewer"class="uk.ac.ic.doc.dc.viewer.ViewerBase"
method="setViewer" />
<interface component="mapapp" name="setLocService"
class="uk.ac.ic.doc.dc.mediaselection.appUIs.mapApp.MapAppLocService"
method="setLocService" />
<interface component="mapapp" name="setQoSMgr"
class="uk.ac.ic.doc.dc.mediaselection.appQoSMgr.mapAppQoSMgr.MapAppQoSMgr"
method="setQoSMgr" />
<interface component="mapapp" name="setLogWriter"
class="uk.ac.ic.doc.dc.logWriter.LogWriter"method="setLogWriter" />
<instance component="mapapp" name="mapapp"/>
<bind   requiror="mapapp" interface="mapapp/setViewer" bindType="static"
providor="viewers"  />
<bind   requiror="locmgr" interface="locmgr/addLocationListener" bindType="static"
providor="mapapp" />
<bind   requiror="mapapp" interface="mapapp/setLogWriter" bindType="static"
providor="logwriter" />
<bind   requiror="screenmgr" interface="screenmgr/bindToScreen" bindType="static"
providor="mapapp" />

<component class="uk.ac.ic.doc.dc.mediaselection.appUIs.mapApp.MapAppLocService"
name="locsvc-mapapp" />
<instance component="locsvc-mapapp" name="locsvc-mapapp"/>
<bind   requiror="locmgr"  interface="locmgr/addLocService"bindType="static"
providor="locsvc-mapapp" paramvalue="Map Application" />
<bind   requiror="mapapp"  interface="mapapp/setLocService"bindType="static"
providor="locsvc-mapapp" />
```

```
<component class="uk.ac.ic.doc.dc.mediaselection.metaData.mapMetaData.MapMetaHandler"
name="mapMetaHandler" />

<!--<component class="uk.ac.ic.doc.dc.mediaselection.appQoSMgr.GroupedSelectByDegCost"
name="selectionalg" />-->
<component class="uk.ac.ic.doc.dc.mediaselection.appQoSMgr.GroupedSelectBySequence"
name="selectionalg" />
<interface component="selectionalg" name="setBWResMgr"
class="uk.ac.ic.doc.dc.mediaselection.resourceMgr.bwMgr.SiteBandwidthResourceMgr"
method="setBWResMgr" />
<!-- <interface component="selectionalg" name="setMoneyResMgr"
class="uk.ac.ic.doc.dc.mediaselection.resourceMgr.moneyMgr.MoneyResourceMgr"
method="setMoneyResMgr" />  -->
<interface component="selectionalg" name="setScreenResMgr"
class="uk.ac.ic.doc.dc.mediaselection.resourceMgr.screenMgr.ScreenResourceMgrI"
method="setScreenResMgr" />
<interface component="selectionalg" name="setSpecMgr"
class="uk.ac.ic.doc.dc.mediaselection.specMgr.SpecMgr"method="setSpecMgr" />
<interface component="selectionalg" name="setTypes"
class="uk.ac.ic.doc.dc.mediaselection.typeSystem.TypeParser"method="setTypes" />
<instance       component="selectionalg"        name="selectionalg"/>
<bind   requiror="selectionalg"   interface="selectionalg/setBWResMgr"  bindType="static"
providor="bwmgr"/>
<!--<bind requiror="selectionalg"   interface="selectionalg/setMoneyResMgr"
bindType="static"  providor="moneymgr"/>-->
<bind   requiror="selectionalg"   interface="selectionalg/setScreenResMgr"
bindType="static"  providor="screenmgr"/>
<bind   requiror="selectionalg"   interface="selectionalg/setSpecMgr"    bindType="static"
providor="specmgr"/>
<bind   requiror="selectionalg" interface="selectionalg/setTypes" bindType="static"
providor="types" />

<component class="uk.ac.ic.doc.dc.mediaselection.appQoSMgr.mapAppQoSMgr.MapAppQoSMgr"
name="mapappqos" />
<interface component="mapappqos" name="setMetaHandler"class="java.lang.Class"
method="setMetaHandler" />
<interface component="mapappqos" name="setPlayout"
class="uk.ac.ic.doc.dc.mediaselection.playout.Playout"method="setPlayout" />
<interface component="mapappqos" name="setTypes"
class="uk.ac.ic.doc.dc.mediaselection.typeSystem.TypeParser"method="setTypes" />
<interface component="mapappqos" name="setSelectionAlgorithm"
class="uk.ac.ic.doc.dc.mediaselection.appQoSMgr.SelectionAlgI"
method="setSelectionAlgorithm" />
<interface component="mapappqos" name="setViewer"
class="uk.ac.ic.doc.dc.viewer.TabbedPaneViewer"method="setViewer" paramtype="string" />
<interface component="mapappqos" name="setMapApp"
class="uk.ac.ic.doc.dc.mediaselection.appUIs.mapApp.MapApp"method="setMapApp" />
<interface component="mapappqos" name="setNetworkResourceManager"
class="uk.ac.ic.doc.dc.mediaselection.resourceMgr.bwMgr.SiteBandwidthResourceMgr"
method="setNetworkResourceManager" />
<interface component="mapappqos" name="setLogWriter"
class="uk.ac.ic.doc.dc.logWriter.LogWriter"method="setLogWriter" />

<instance component="mapappqos" name="mapappqos"/>
<param  instance="mapappqos"method="setMapServBaseURL"type="string"value="http://www-
dse.doc.ic.ac.uk/servlets/mapservlet/mapsrv?" />
<param  instance="mapappqos"method="setUseZIP"type="boolean"value="true" />
<bind   requiror="mapapp"  interface="mapapp/setQoSMgr"bindType="static"
providor="mapappqos" />
<bind   requiror="mapappqos" interface="mapappqos/setLogWriter" bindType="static"
providor="logwriter" />
<bind   requiror="mapappqos"  interface="mapappqos/setMapApp"bindType="static"
providor="mapapp" />
<bind   requiror="mapappqos"      interface="mapappqos/setMetaHandler" bindType="dynamic"
providor="mapMetaHandler" />
<bind   requiror="mapappqos" interface="mapappqos/setPlayout" bindType="static"
providor="playout" />
```

```
<bind  requiror="mapappqos" interface="mapappqos/setTypes" bindType="static"
providor="types" />
<bind  requiror="mapappqos" interface="mapappqos/setSelectionAlgorithm"bindType="static"
providor="selectionalg" />
<bind  requiror="mapappqos" interface="mapappqos/setViewer" bindType="static"
providor="controls" paramvalue="Meta Data" />
<bind  requiror="mapappqos" interface="mapappqos/setNetworkResourceManager"
bindType="static"  providor="bwmgr" />

<param instance="viewers" method="setXSize" type="int" value="400" />
<param instance="viewers" method="setYSize" type="int" value="400" />
<param instance="viewers" method="showUI" />
<param instance="controls" method="setXSize" type="int" value="400" />
<param instance="controls" method="setYSize" type="int" value="400" />
<param instance="controls" method="showUI" />

</configuration>
```

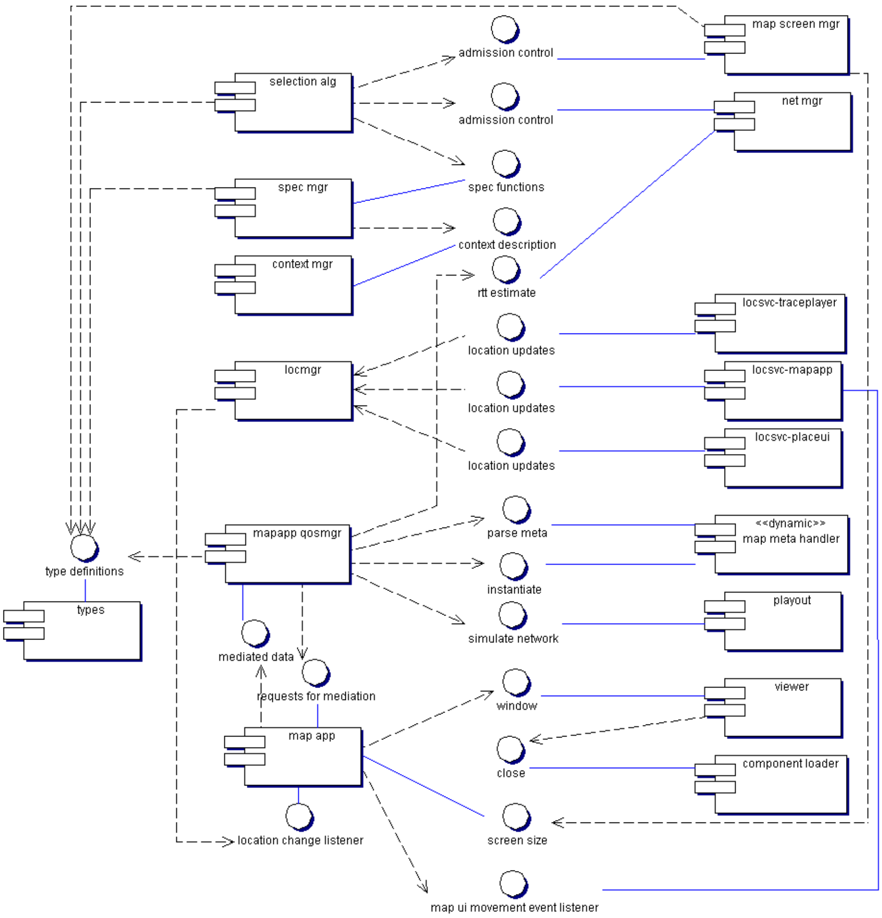


Figure A4-1: Component Bindings in Example Configuration

A4.2 Software Engineering Notes

There follows UML diagrams, depicting package structure, inheritance and dependencies for the code developed for the map application. The are over 22,000 lines of Java code (including comments and white space) which we do not propose to describe in detail. An overview of the software engineering behind the application is provided to aid understanding of the application only.

The package `uk.ac.ic.doc.dc` forms the practical basis for the project.

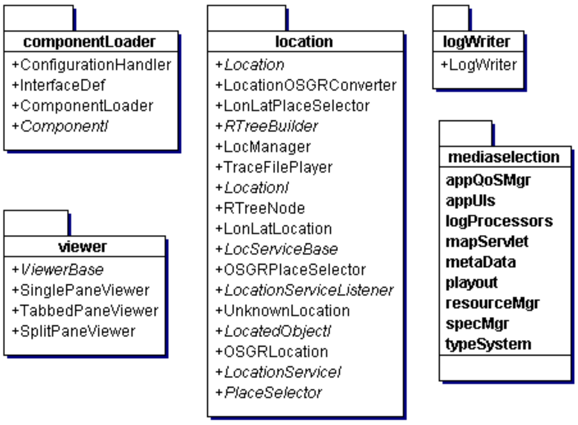


Figure A4-2: Package uk.ac.ic.doc.dc

The component loader, as described in section a4.1 is essentially formed from an XML parser which employs reflection to load the components into the Java virtual machine as specified and set up relationships between them.

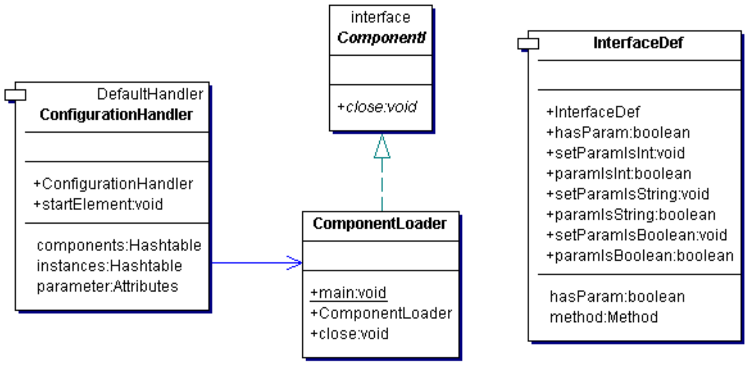


Figure A4-3: Package uk.ac.ic.doc.dc.componentLoader

Package `uk.ac.ic.doc.dc.mediaSelection` forms a holding package for the majority of the code dealing with mediation, the map application and map server.

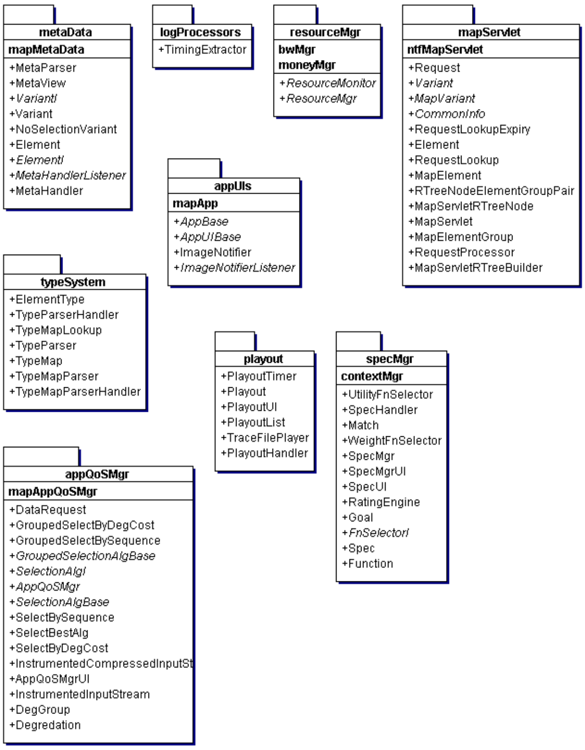


Figure A4-4: Package uk.ac.ic.doc.dc.mediaSelection

`uk.ac.ic.doc.dc.mediaSelection.appUIs` provides base classes for applications to extend which structure interaction with the application QoS manager and common application tools such as saving an image from the display and providing a progress bar.

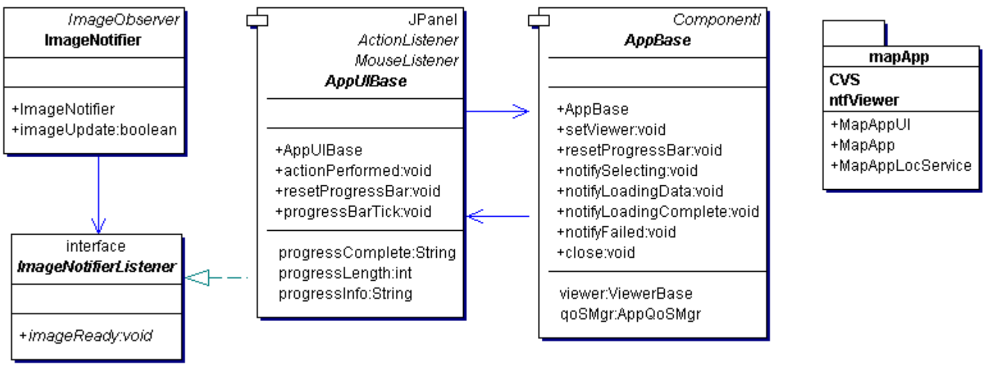


Figure A4-5: Package uk.ac.ic.doc.dc.mediaSelection.appUIs

The uk.ac.ic.doc.dc.mediaSelection.appUIs.mapApp package and its sub-package ntfViewer provide the display of the map data, navigation using the screen as a location service (as described in appendix 1).

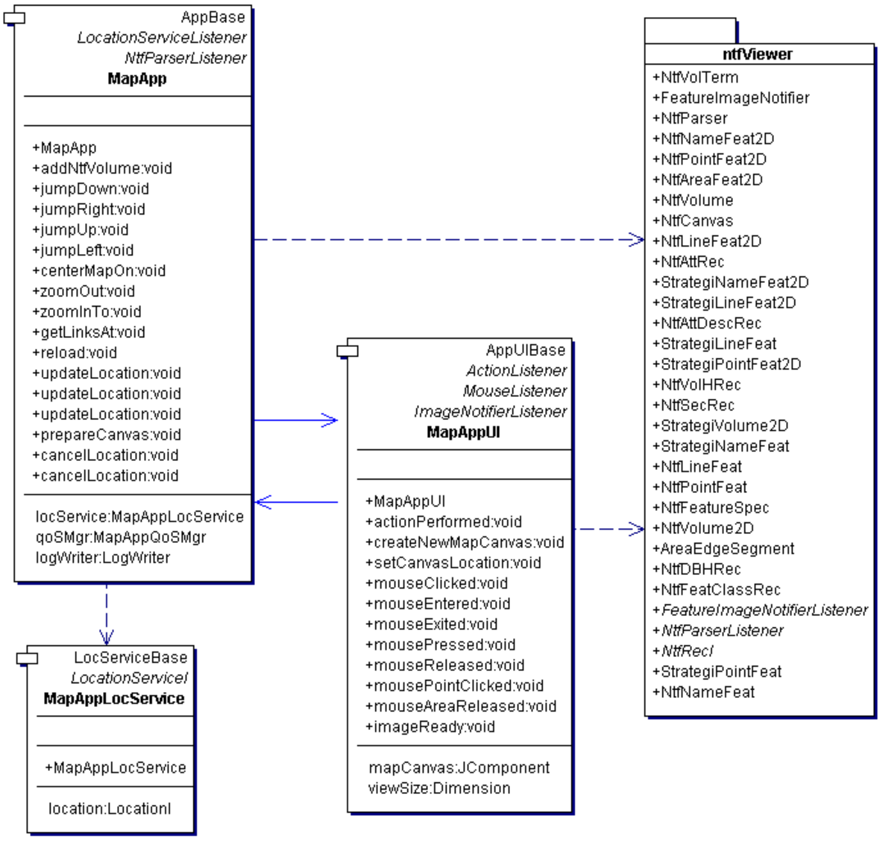


Figure A4-6: Package uk.ac.ic.doc.dc.mediaSelection.appUIs.mapApp

`uk.ac.ic.doc.dc.mediaSelection.appQoSMgr` includes the majority of the mediation process. `AppQoSMgr` is a base class providing an interface to applications and common programming structures. `InstrumentedInputStream` and its compressed sub-class, together with `DataRequest` are used to handle data loading and measuments of network performance. The remaining classes, based on `SelectionAlgBase` and `Degredation` are used to perform the selection according to the required algorithms, with or without grouping of elements in the degredation path. The algorithm used is loaded in the configuration.

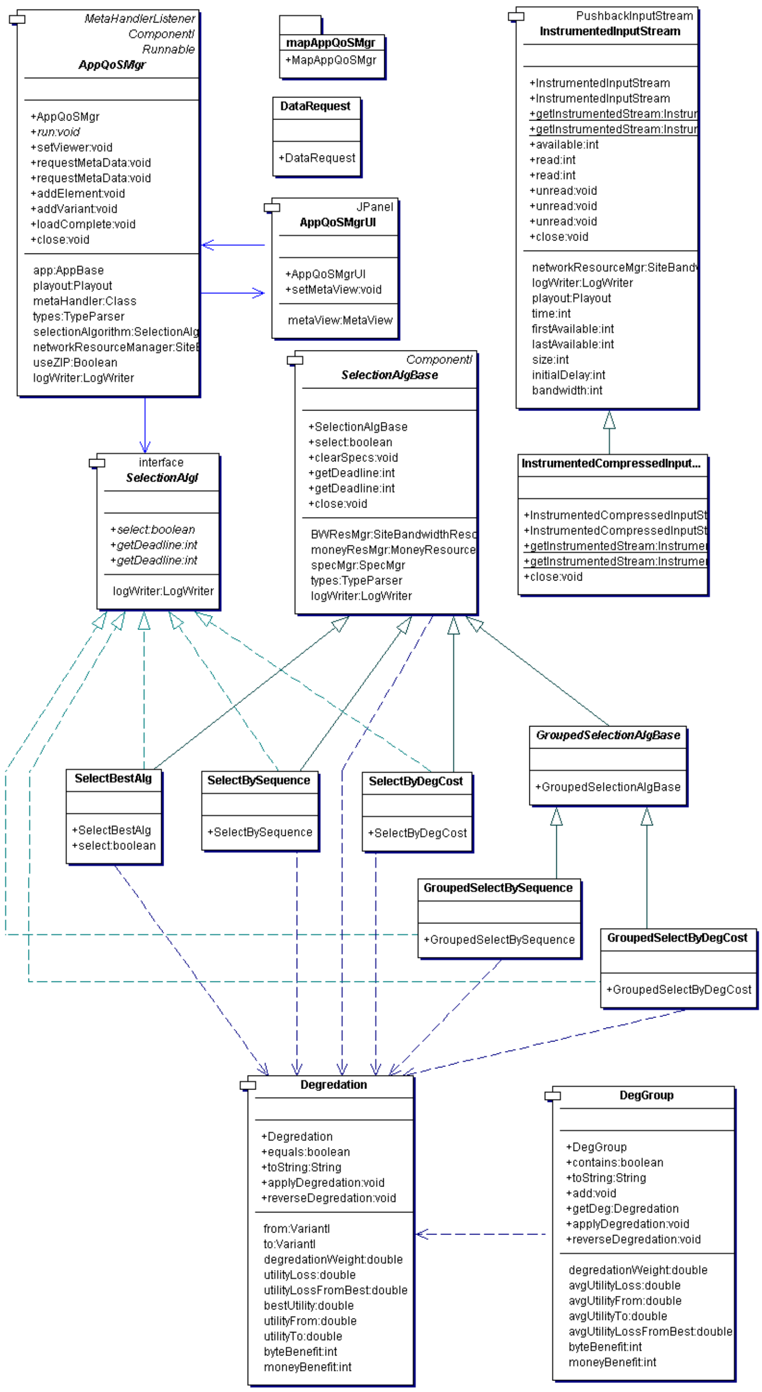


Figure A4-7: Package uk.ac.ic.doc.dc.mediaSelection.appQoSMgr

`uk.ac.ic.doc.dc.mediaSelection.appQoSMgr.mapAppQoSMgr.MapAppQoSMgr` is a sub-class of the base AppQoSMgr which provides interactions specific to the map application, with an understanding of the form of the data requests to the server and how the returing data is used by the application.

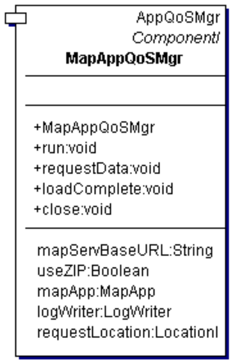


Figure A4-8: Package uk.ac.ic.doc.dc.mediaSelection.appQoSMgr.mapAppQoSMgr

Package `uk.ac.ic.doc.dc.mediaSelection.specMgr.contextMgr` provides the models of discrete and continuous context for different context aspects. Definitions of allowed values for aspects are loaded using the `ContextHandler`.

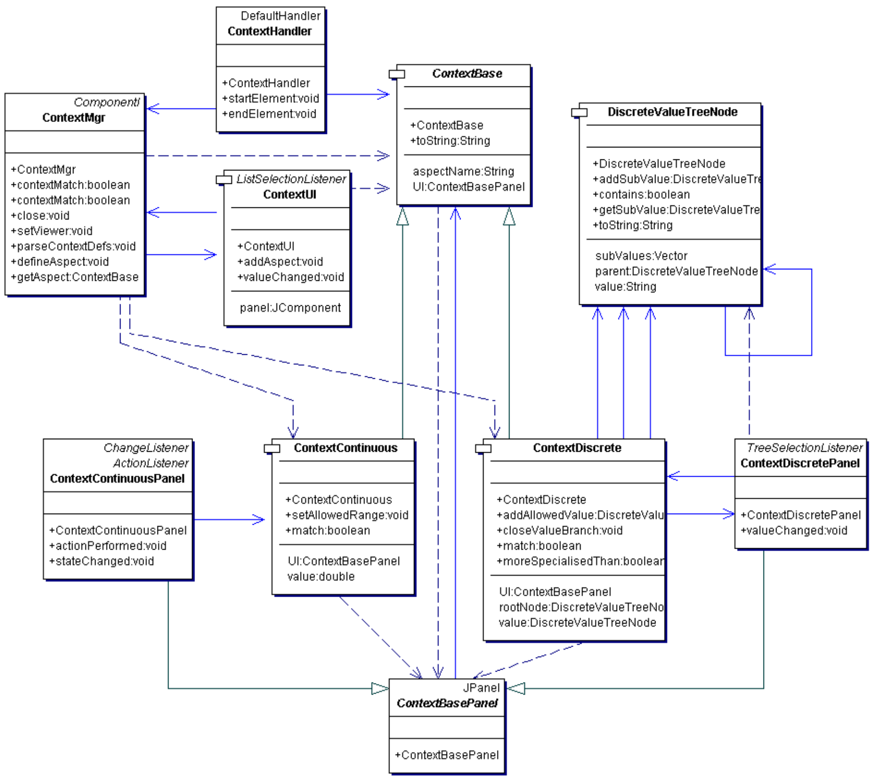


Figure A4-9: Package uk.ac.ic.doc.dc.mediaSelection.specMgr.contextMgr

The type system is provided by classes in `uk.ac.ic.doc.dc.mediaSelection.typeSystem`. This includes loading type definitions, translating between type systems, e.g. NTF feature codes, and providing the element type class used in the mediation process.

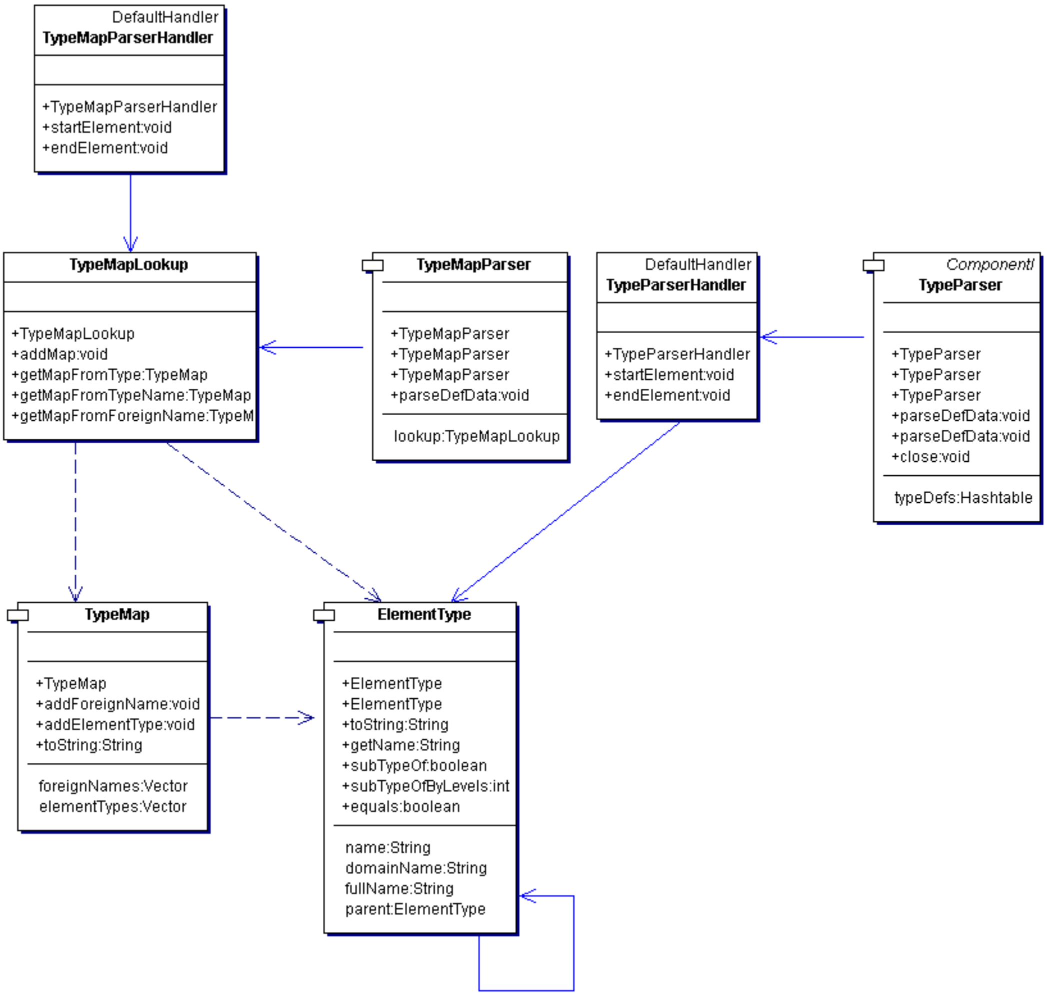


Figure A4-10: Package uk.ac.ic.doc.dc.mediaSelection.typeSystem

The type system is provided by classes in `uk.ac.ic.doc.dc.mediaSelection.specMgr`. This includes loading spec definitions and providing the weight, utility and goal function selector classes used in the mediation process. `SpecMgr` and `RatingEngine` together provide the mechanism for passing over the trees of elements and variants and assigning weights or utility functions to them according to context.

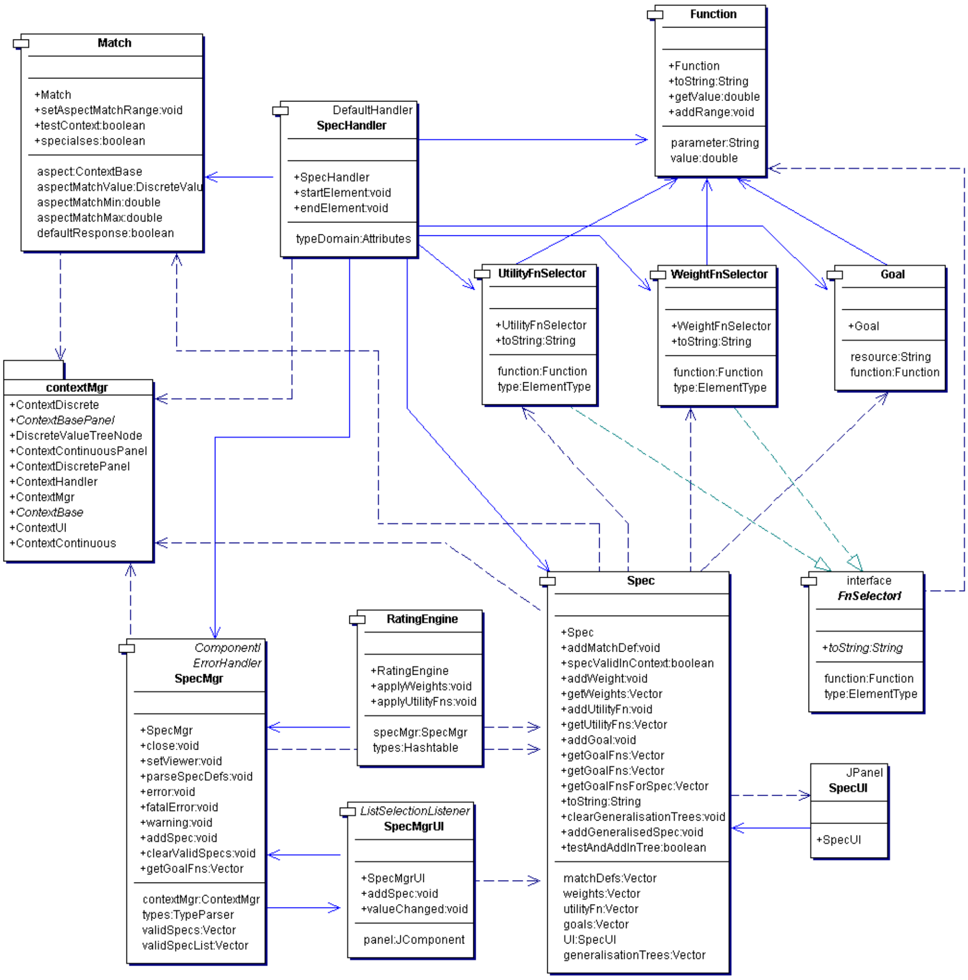


Figure A4-11: Package uk.ac.ic.doc.dc.mediaSelection.specMgr

The package `uk.ac.ic.doc.dc.mediaSelection.metaData` is used in the mediation process (not in the server) to parse the incoming meta data and form trees of elements and variants. The `NoSelectionVariant` is used in the selection process to represent no selection, with a fixed utility and size of zero. The `mapMetaData` package extends the `metaData` package to provide parsing, elements and variants which represent issues in map data, including location and scale.

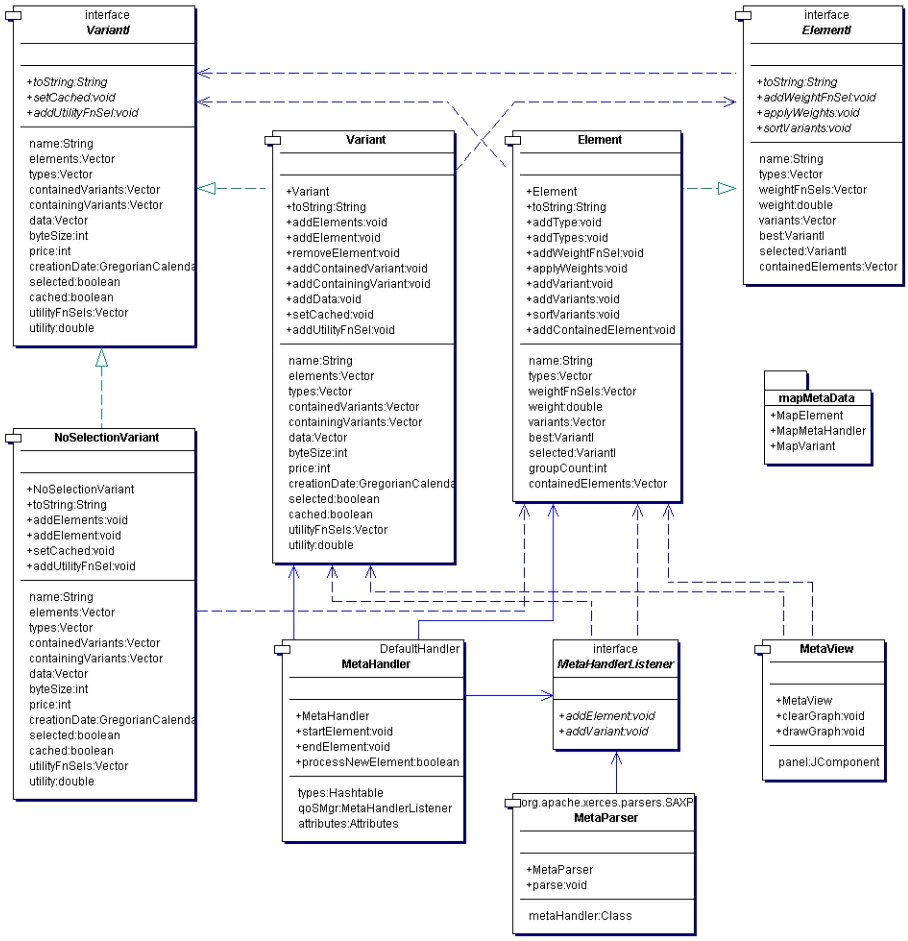


Figure A4-12: Package uk.ac.ic.doc.dc.mediaSelection.metaData

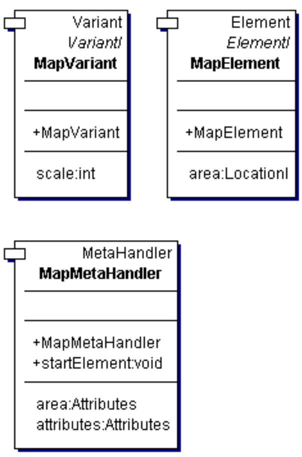


Figure A4-13: Package uk.ac.ic.doc.dc.mediaSelection.meatData.mapMetaData

The network simulation is contained within `uk.ac.ic.doc.dc.mediaSelection.playout`. The `InstrumentedInputStream` passes calls for data through `Playout`. The request is converted into a `PlayoutHandler` recorded on the `PlayoutList`. The `PlayoutTimer` with input from the `PlayoutUI` or `TraceFilePlayer` provide the network simulation.

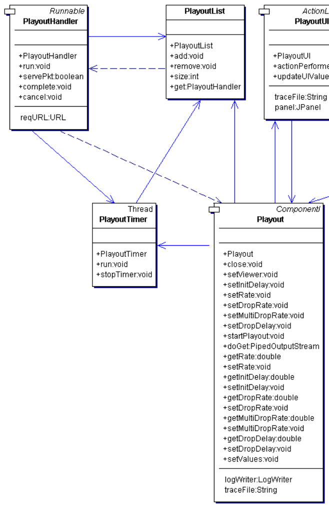


Figure A4-14: Package uk.ac.ic.doc.dc.mediaSelection.playout

`uk.ac.ic.doc.dc.mediaSelection.resourceMgr` contains the three resource managers developed, and some base classes which these use. The money resource manager is not described in detail in this thesis.

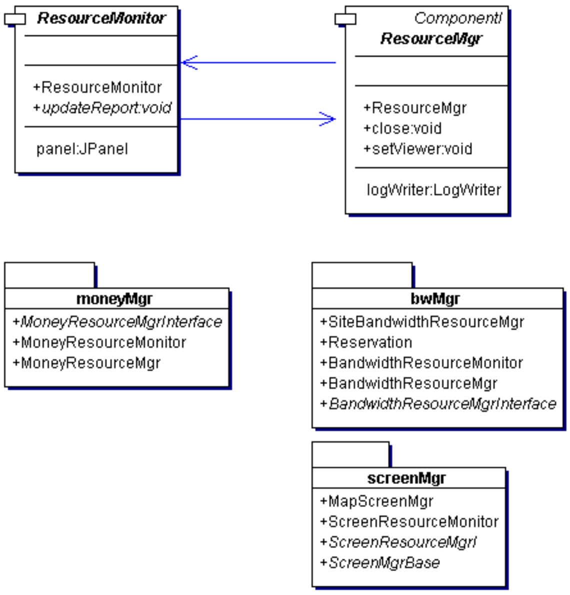


Figure A4-15: Package uk.ac.ic.doc.dc.mediaSelection.resourceMgr

The `bwMgr` package under `resourceMgr` provides both the models of throughput and round trip time (per URL stem in `BandwidthResourceMgr` and `SiteBandwidthResourceMgr`) and the throughput reservation system (`BandwidthResourceMgr` and `Reservation`). A user interface is also provided through `BandwidthResourceMonitor`.

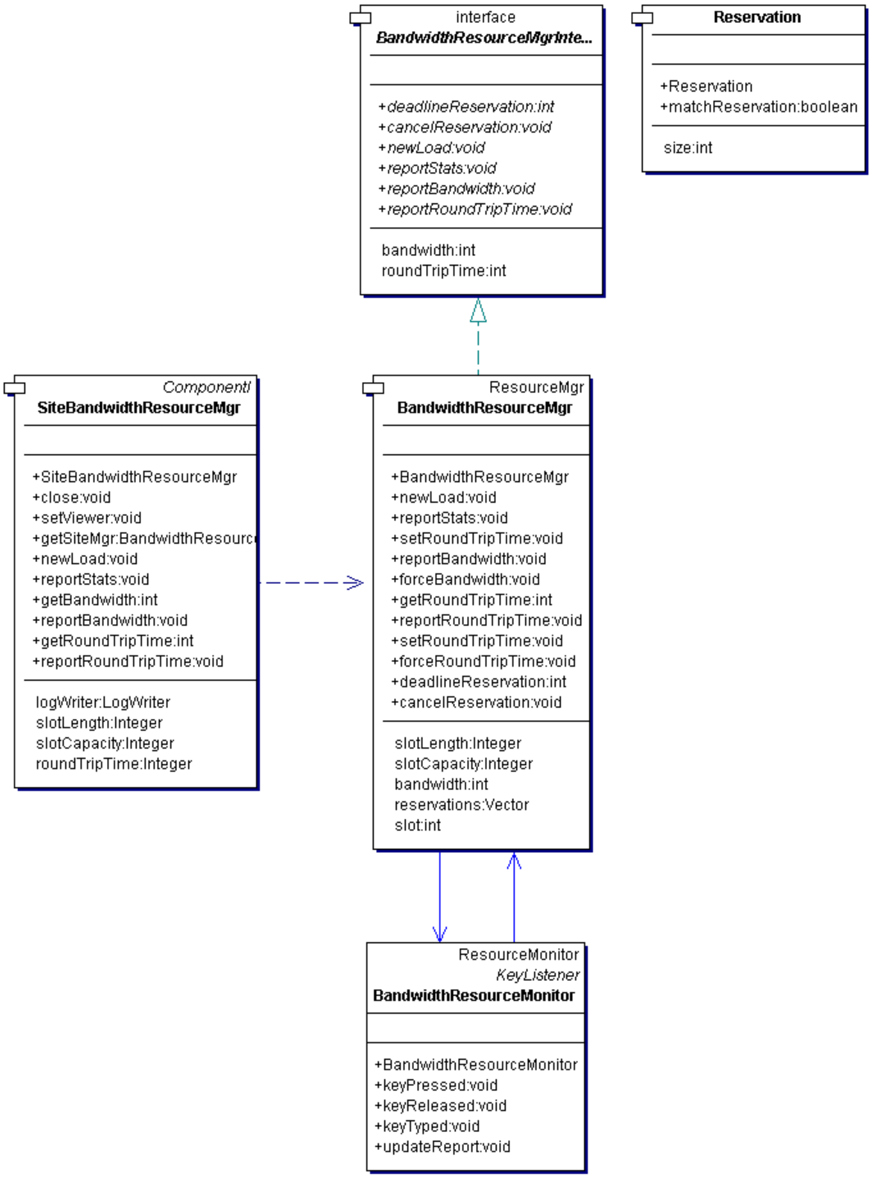


Figure A4-16: Package uk.ac.ic.doc.dc.mediaSelection.bwMgr

The `uk.ac.ic.doc.dc.mediaSelection.screenMgr` package provides the generic screen density admission control functions through `ScreenMgrBase` with interface to the map application through `MapScreenMgr`.

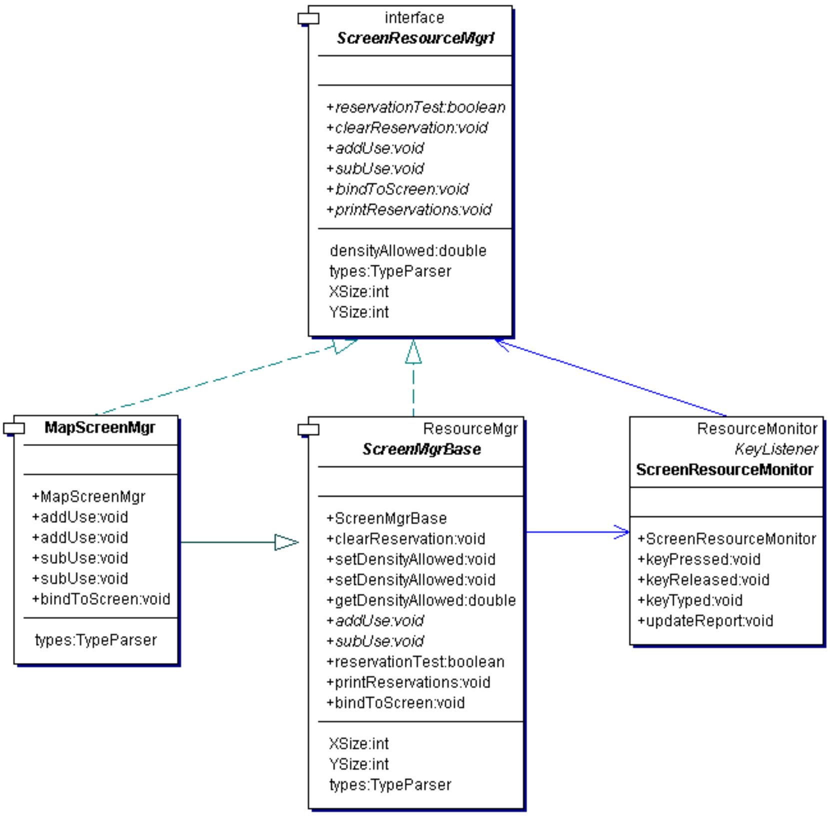


Figure A4-17: Package uk.ac.ic.doc.dc.mediaSelection.screenMgr

The location service and models of location and distance used in element weight evaluation and the map server (including an RTree) are collected in `uk.ac.ic.doc.dc.location`. Support for reading a “script” of locations into the location service comes from `TraceFilePlayer`. Conversion between Ordnance Survey Grid coordinates and Longitude / Latitude are provided in `LocationOSGRConverter`. Further discussion of these functions can be found in appendix 1.

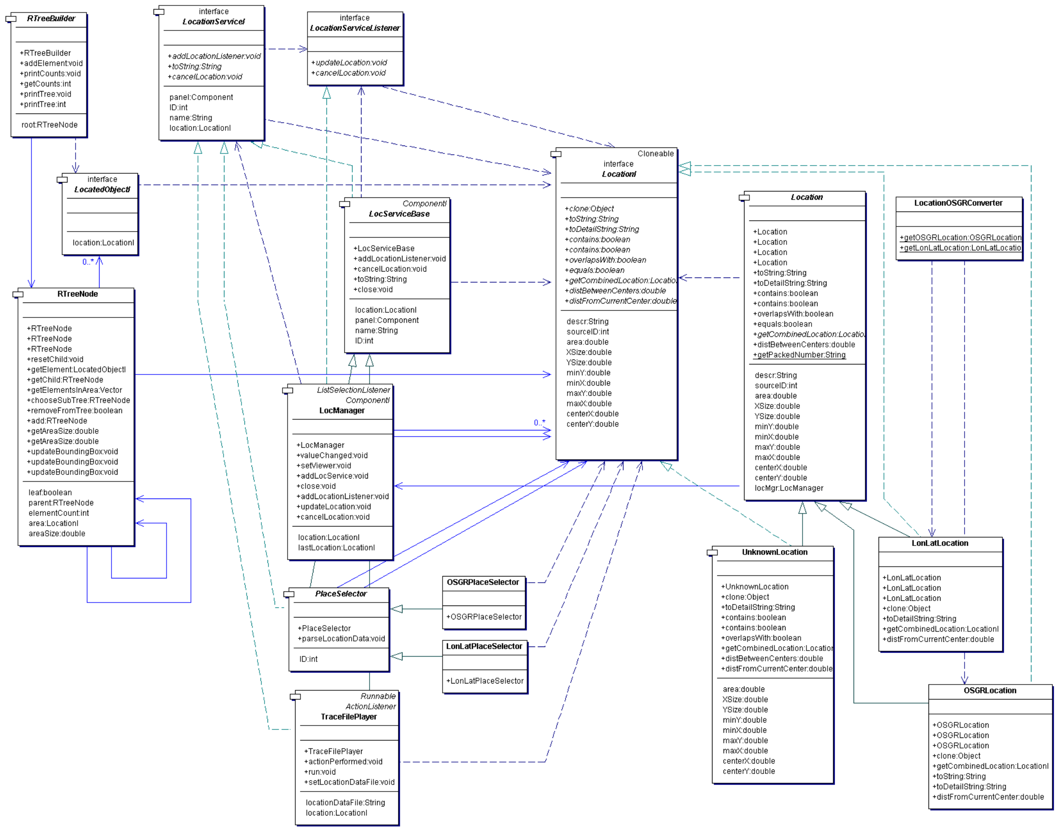


Figure A4-18: Package uk.ac.ic.doc.dc.location

`uk.ac.ic.doc.dc.mediaSelection.mapServlet` implements a Tomcat Java Servlet which interprets data requests and returns meta data or map data as appropriate. Internally data is stored as elements and variants, which are generated at initialisation by parsing the NTF files. This servlet in implemented for testing purposes and does not represent the most elegant solution to serving map data.

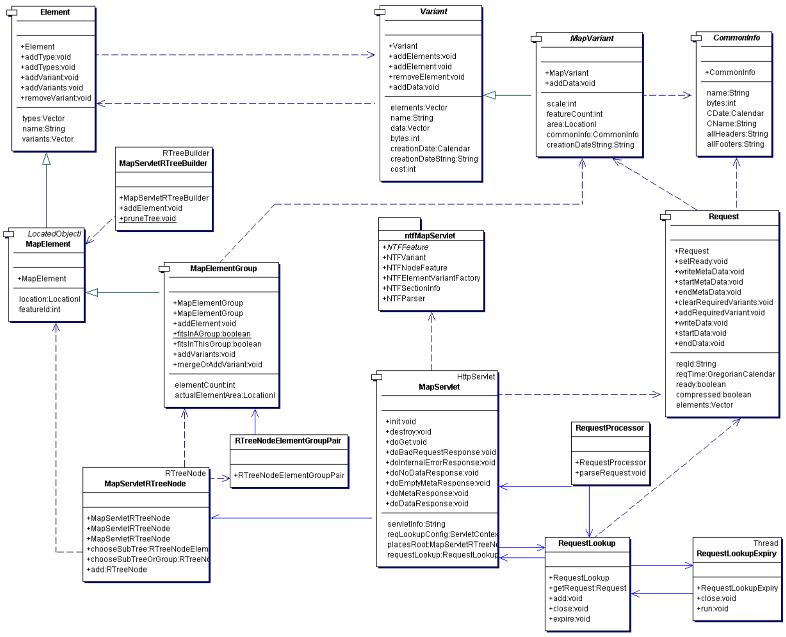


Figure A4-19: Package uk.ac.ic.doc.dc.mediaSelection.mapServlet