# Designing mobile-aware adaptive hypermedia

**Toni Alatalo**
Toni.Alatalo@oulu.fi

**Janne Peräaho**
Janne.Peraaho@oulu.fi

A position paper for the 2nd session of the adaptive hypermedia workshop 2001, to be held at the ACM Hypertext conference, August 2001.

Department of Information Processing Sciences, University of Oulu
OWLA project, http://owla.oulu.fi/

## Abstract

*Mobile use presents new challenges for hypermedia design. One of the key elements in mobile-aware hypermedia is adapting to users' situations, but somewhat differently than in the traditional user-modelling centric view of adaptive hypermedia. This paper presents an example of designing new navigational structures for an existing Web service, with the mobile use in mind. A previously presented modelling method for designing adaptive hypermedia is used in the specifications. The system's adaptation to location and time information - but not to any personal user specific properties - is modelled and implemented as a proof of concept demo, as a new layer on top of the existing service. The usefulness of the previously introduced modelling method and other approaches to designing adaptive hypermedia in this context are discussed, and lessons learned from the test and questions for future work pointed out in conclusion*.

## Introduction

In recent years, much attention has been paid to mobile computing within IT industry. Successful mobile IT has been developed including mobile phones, PDAs (Personal Digital Assistants), communicators, and wearable computers. Following the Internet revolution has even been said to be the wireless revolution (Hjelm 2000).

A fundamental attempt to conceptualize mobility in a larger sense has been carried out in Swedish Viktoria Institute, where a reference model has been developed to provide designers with a framework of concepts to understand how people use IT in mobile settings (Dahlbom & Ljungberg 1998). The concerns in the model are environment, modality and application of mobile IT use, each of them having further details. Here the focus is on the first entity - environment - and specifically the physical environment as the use context. Design of mobile applications often aims to *context-awareness*, the ability of application to extract, interpret and use situational information and adapt functionality to the current context of use. Examples of context information are (Korkea-aho 2000): identity, spatial information (location, orientation, speed, acceleration), temporal information (time of the day, date, season of the year), environmental information (temperature, air quality, light or noise level), social situation (who you are with, people that are nearby), resources that are nearby (accessible device and hosts), availability of resources (battery, display, network, bandwidth), physiological measurements (blood pressure, hart rate, respiration rate, muscle activity, tone of voice), activity (talking, reading, walking, running) and schedules and agendas. A concise overall definition: "*context is everything but the explicit input and output"* (e.g. the states that effect the application's behaviour) is given in (Lieberman & Selker 2000).

The current design methods - hypermedia or others - do not currently support the design of adaptivity in general, nor especially contextuality. An exception is the adaptive hypermedia design method (AHDM) which includes the design of user models within the methodology (Koch 1998), but does not concern context-awareness - nor does it support the actual modelling of adaptivity in design specifications. An attempt to address these issues was presented in (Alatalo & Peräaho 2001), but with a user modelling based adaptation example only. Here we experiment with that modelling method further, addressing the issues of mobile use and contextuality from a hypermedia design perspective.

It has been suggested that the *design of context* - i.e. considering context as the object of a design activity - would be "a natural extension of current approaches that treat context as information for the design of the artifacts (design with context) or as the scenario for enacting the design activity (contextual design or design in context)" (Roque & Almeida, 2000). We share this interest, and therefore focus on the design phase that occurs after early analysis and requirements elication, trying to come up with ways to model specifications for actual implementations with the richness of information that can be found surrounding the use. The example this position paper presents is yet a simple one, but we consider it as a proof of concept to begin with.

In the following part, the previously presented design approach and modelling method is recalled. Then the actual design problem - supporting the mobile use of an existing food place information service - is presented as the starting point. The design of the solution is presented phase-by-phase with the according diagrams resulting in the adaptive navigation structure. The work done is discussed regarding the limitations of the study, related efforts and ideas for the future - then concluded in the end.

## Approach

Adaptation can be understood as a process where information, contexts, navigational structures, and/or user interface changes according to (user) profiles and adaptation rules. In these kind of systems profiles and adaptation rules can be static (set once, never change), updatable (change by request), or dynamic (in constant change). Examples of this in adaptive hypermedia are e.g. AHM (da Silva et al 1998) and AHAM (De Bra, Houben & Wu 1999), IMPS SRM (Bordegoni, Bulterman et al 1999) and for an early example in the field of human-computer interaction see (Gargan, Sulliwan & Tyler 1988). We call this *the engine view of adaptation*, see figure 1.

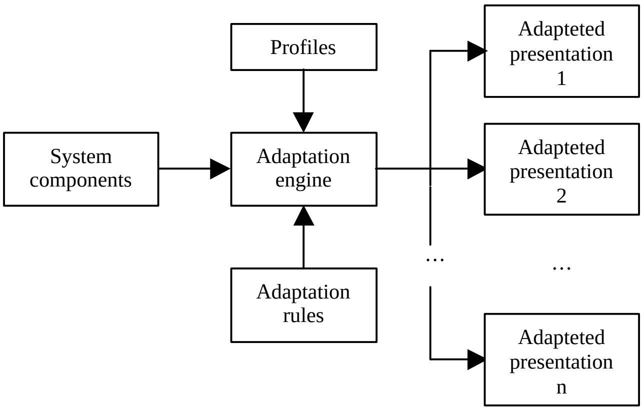



Figure 1. The engine view of adaptation.

The Structural Adaptation (Alatalo & Peräaho 2001) approach to adaptation is quite different. Structural adaptation suggests that adaptation can be hypermedia system's property, too, and not merely a process. In practice, the architectures may well be similar, but as the focus here is on conceptual modelling in design specifications, the different view on adaptation matters when defining what to model.

In the structural adaptation adaptation is integrated in the system's information structures, functions, contexts, navigational structures and user interface. The basic idea is that system can be divided into three parts which together forms an adaptive hypermedia system. The parts are as follows: Adaptors, Heuristics, and Transformants. See Figure 2.

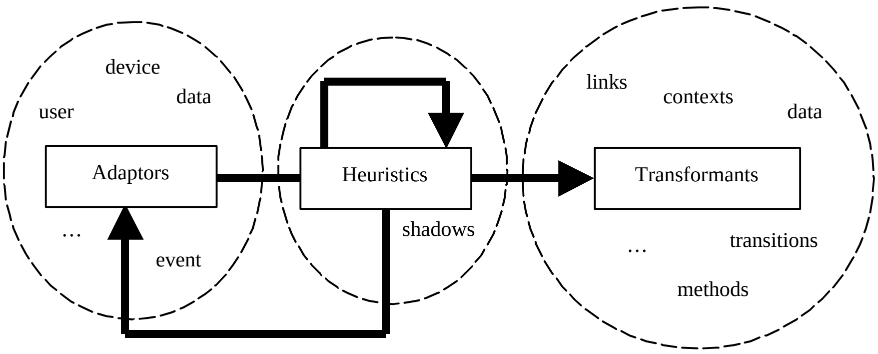



Figure 2. Structural adaptation.

These parts can be understood as specific roles some objects in the system have with respect to adaptivity.

*Adaptors* are defined as a system components to which the rest of the system adapts. In harmony with the object oriented design each adaptor consist of attributes, known as adaptor’s *properties*, and methods. Adaptor’s properties play an important role in the adaptation, since they determine relations between adaptors and system’s adaptive components, as defined by heuristics.

*Transformants* are the name for system’s adaptive components. They adapt to adaptors according to heuristics. Transformants are class, attribute and method constructs containing conditional and non-conditional assosiations. According to transformant structure it can consist of non-adaptive parts (uses non-conditional assosiations), known as the skeleton, and adaptive parts (uses conditional assosiations).

In (Alatalo & Peräaho 2001) seven different transformant categories were presented. We add "Nodes" category to complete the classification. The complete eight category list for hypermedia system's transformants is as follows:

- Data – transformants based on application domain data
- Nodes - transformants based on hypermedia nodes
- Links – transformants based on hypermedia links
- Contexts – transformants based on context
- Transitions – transformants based on links between contexts
- Methods – transformants based on methods
- Adaptors – transformants based on adaptors
- Heuristic rules – transformants based on heuristic rules

Adaptors and heuristic rules acting as transformants is quite rare, but still possible and justified.

*Heuristics* defines two-fold relations between adaptors and transformants: a) to which adaptor's properties transformant has relation to (disclosed with variables) and b) what is the nature of the relation (mathematical and logical dependency).

Heuristics, which are modeled with *shadows*, can be classified based on adaptors to which the heuristic rule is mainly involved. The classification cannot be strict, only suggestive, due to cross references between different adaptors. Nevertheless, giving names, even unaccurate ones, to the shadows does seem support the design work.

The shadow consists of one clause (heuristic rule), containing variables, constants, mathematical- and logical operands, which defines the relation of one or several adaptors and transformants. As a result of this each shadow has a binary output value, 0 (false) or 1 (true), which indicates adaptor's adaptive part's inclusion or disjunction. In other words, shadows acts as a logical glue which holds the transformant’s skeleton and adaptive parts together. Without these links adaptive system can not exists.

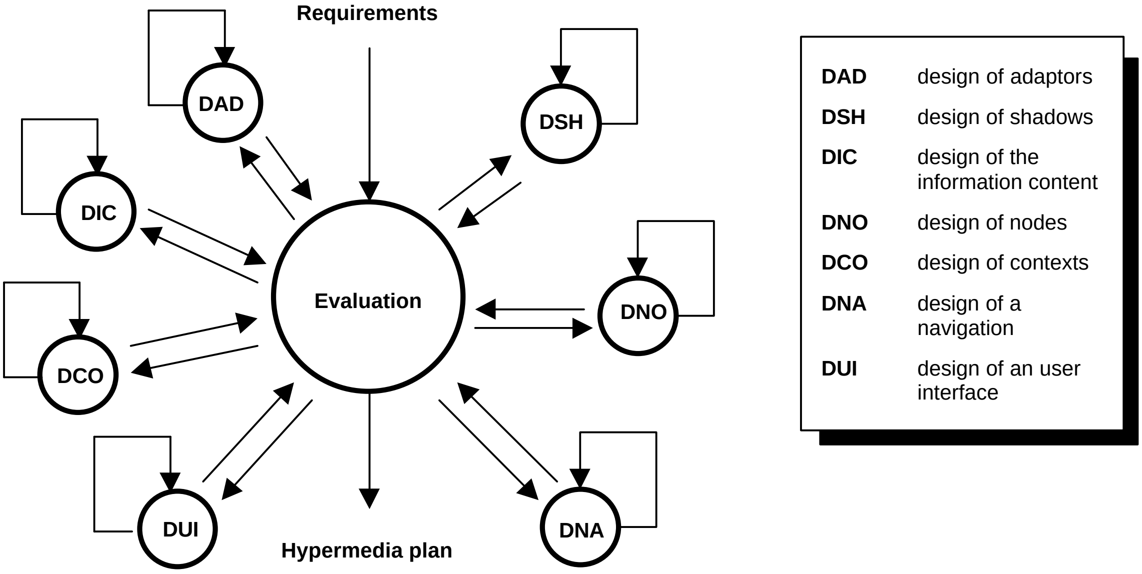



Figure 3. Design tasks of an adaptive hypermedia system.

The design of an adaptive hypermedia system was divided into six tasks and design issues and an incremental iterative design process along with special design tools was recommended for accomplishing the tasks.

The design of hypermedia nodes was part of the navigation design in the previous publication (Alatalo & Peräaho 2001). We now extract the node design into a separate design task: design of nodes. See figure 3. As will be shown later, although presented here as separate concerns, nodes and links can be specified in the same diagram, as in the navigational class diagrams in OOHDM (Schwabe&Rossi 1998) and in the navigational structure diagrams in the UML based method presented in (Hennicker & Koch 2000).

## Design problem

#### Background

The particular design problem at hand is based on an existing service, which provides basic information, detailed descriptions and ratings of food places in the city of Oulu, Finland, on the web. The website is quite a peculiar one, and became an instant success soon after its launch early 2001. The service is run by individual private people, who set it up for fun, without thinking of it as a business nor consulting the actual food place owners. It should be noted, that the scope of the service is not just any food places, and not at all any high-class restaurants, but initially the certain pizzerias that had become somewhat a phenomenon in the city. The slang word for the food in these lower-class nightlife-oriented food places in Oulu is "känkky", hence the URL of the service: http://kaenkky.u--3.com/ (in Finnish only) . Later, the service has grown to cover other similarly used food services such as grills, and finally quite nice restaurants too (although the food there is not considered "känkky").

The content is entirely created by the users, who can submit reviews, and edited by (a team of) administrator(s). As the word spread (mostly on the Internet), the totally unorganized people quickly contributed insightful and greatly humorous stories soon covering the whole range of food places in the region - some enthusiasts had digital cameras with them, so the site has photos of the places and even of the actual meals. In addition, there's a rating function (points from 1 to 5) and a top-five and low-five list derived from the ratings.

Initially, the service was (implicitly) designed for desktop-based use only. Being busy with reseach related to mobile services, we tried using the service with mobile devices too - first with a PDA (Psion5mx running Opera on Epoc) and with a WAP-phone (after tweaking a wrapper for WML on the site). With the PDA, the service was barely usable, but for the WAP-phone with an even smaller display it was just too much. Also with the PDA it helped to know the service from before. To ease the download times and browsing, we created a lite-version of the html-based web service with pictures and other unnesessary information excluded, which was already better for the PDA. But still it was evident that to make the service really usable for the mobile user it would need to be thought over.

#### Requirements for mobile use

To specify the requirements for mobile use of the food place service, we identified two simple use cases (or user stories): 1. A mobile user is hungry, and does not know where to go to eat, so needs to find a place that can satisfy the hunger soon enough but with an adequate quality too. 2. A mobile user is standing by a foodplace, wondering whether it's a nice one - not necessarily being hungry right now, but perhaps trying to decide about going there some time later - and would like to have more information about the place. These cases are depicted in figure 4.

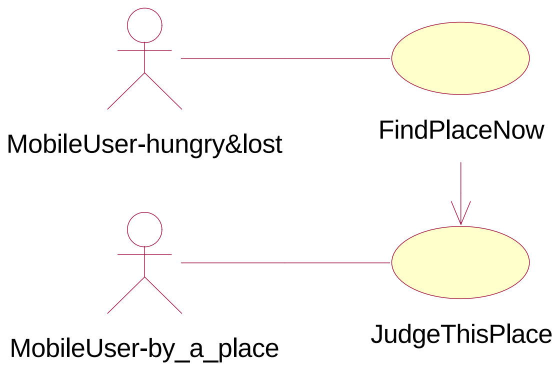



Figure 4. The two use cases, setting requirements for the mobile use

The actual information about the food places can be derived from the existing service via an interface (see figure 5).

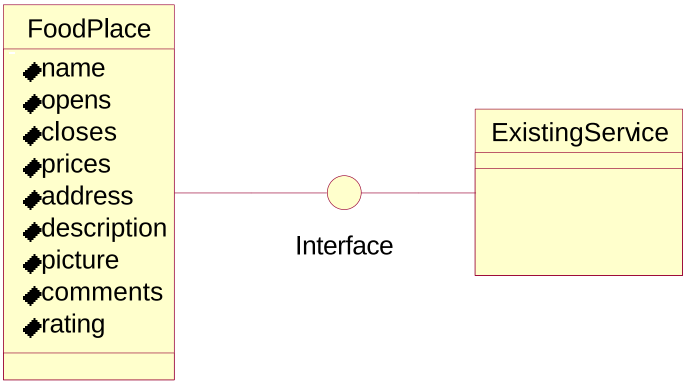



Figure 5. Initial conceptual model, with information derived from the existing service

Following the design process presented before, the next tasks are the design of the information content, design of nodes, design of context, design of navigation, and design of an user interface. We decided to do this using the OOHDM-like (Schwabe&Rossi 1998) method with UML-based notation as presented in (Hennicker & Koch 2000). Their methodology covers the same design objectivities and provides ready-to-use syntax for diagrams. At the same time it allows us to show how an existing hypermedia design model, which was not originally meant for designing adaptive hypermedia, can be complement to meet the challenges set by the adaptive hypermedia systems by extending it with the concepts adaptor and shadow.

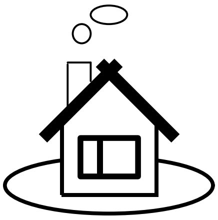
## Solution

We have gathered requirements for food place service, constructed use cases, and drawn a preliminary conceptual model on the basis of the requirements. So far the design process has evolved like in any other hypermedia system design, but now it is time to think system's adaptive qualities more closely.

### Design of Adaptors

Design of food place service's adaptive qualities starts by identifying adaptors. From the use cases we can derive that the system needs to be adaptive with respect to time and place. Time refers to the relationship of current time and food place's opening and closing times, and place refers to the relationship of food place's location and user's location.

Opening and closing times and food place's location are clearly food place's properties and they are definitely needed for adaptation. In the domain model (figure x) we defined the class *FoodPlace*. Now this class can be used as a basis for an adaptor called *FoodPlace*. See figure 6.



Food Place

| FoodPlace |
|---|
| name: String<br>location: Location<br>address: String<br>contact: String<br>opens: Time<br>closes: Time<br>prices: Money<br>description: String<br>picture: Image<br>comments: Set(String)<br>rating: Float |
| TravellingTime(): Time<br>Open(): Boolean |

Figure 6. Adaptors.

We supplement the class with *location* attribute in order to make location adaptation possible, and add two methods to fulfill the time and location adaptation requirements. Method *Open* determines if food place is open in the basis of food place’s opening and closing times and current (system) time. Method *TravellingTime* uses food place’s location and user’s current location while evaluating user’s travelling time and/or distance to the food place.

User's location is needed only for evaluating travelling time between user's current location and the food place, and the user's location is required only when user is requesting recommendation for food place or acquiring food place information which is close by. There is really no need to store any user information, not even the frequently changing location, and we will not therefore need a user adaptor. As there is no information tied to an individual user, there is no "user" in the class diagram, and no user model at all. Instead, in order to adapt, the system requires information about the *use location*. Technically this position information can be e.g. included within the http-request (coming from the client via the network to the server).

#### Design of Shadows

Designing the adaptation, the shadows, is about designing filters, which hide transformant qualities depending on values of properties of adaptors. However, in this case it is not possible to make a clear difference between adaptors and transformants, and for this reason defining the adaptation is a bit tricky and maybe confusing.

It was required that the system recommends food places based on their availability (time and location) and supports quality judgements with food places' ratings and reviews. The second requirement was that system should provide information about food place where user is now or is about to enter.

In the first scenario system should recommend food places in basis of food places' reviews and availability. Earlier we recoqnized the adaptor *FoodPlace* which holds all the information associated to food place, including review, opening and closing times, and location which are needed for making recommendation. Because the result of recommendation is a prioritized group of food places, or more precisely a list of some of the food places' properties, adaptor *FoodPlace* is not only an adaptor but also a transformant. As noted when introducing the design method, this is not contradictary (an object can well have several roles).

To satisfy the needs set by the first use case, one should determine if a food place is currently open and if it is reasonably near. In addition food places need to be ordered by their reviews.

Ordering the food places is not the matter of shadows but the two previous tasks are. (Observe that shadows can be used as search and ordering criteria but not as a method, which produces ordered list of items.) We create two shadows: *Open*, which determines if food place is currently open, and *Near*, which determines if food place is near the user requesting recommendation. See figure 7.

In the second use case, the system should display information about the food place whereby the requesting user  is now. For this purpose we create a shadow *AtDoor* depicted in figure x. Note that this functionality requires quite accurate positioning (current GPS is adequate, and emerging networking technologies such as Bluetooth offer this kind of possibilities), and has to be omitted when there is not such (this detail is not implemented here, but could be easily programmed within the shadow).

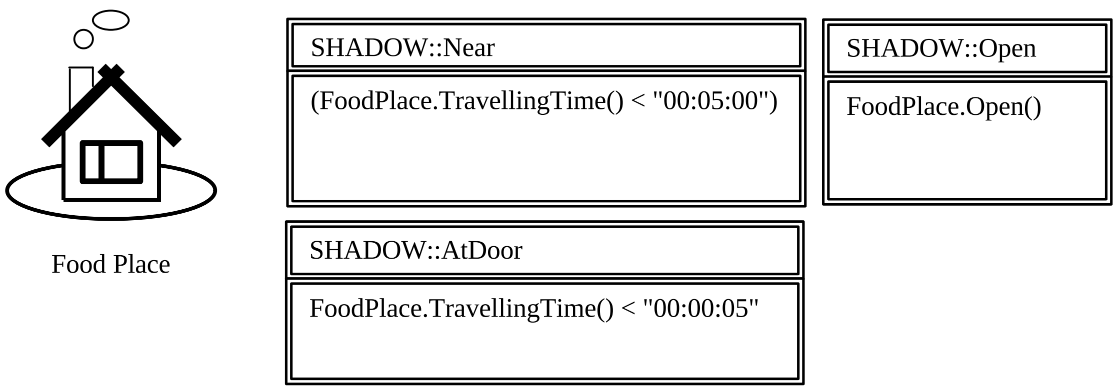



Figure 7. Shadows.

As the focus here is on conceptual modelling, the implementation of shadows is not tied to any specific techniques. One solution is to convert shadows to the existing class's methods. In this case shadows *Near*, *AtDoor* and *Open* can be added to the *FoodPlace* class as methods, to be used for determining if food place is near or open, or if user is standing next to the food place.

We have now enough information to polish the conceptual model presented earlier.

#### Design of Information Content

The concept of location is more complicated than it may look at the first sight. When evaluating travelling times, user and food place location should be expressed with reasonable high accuracy to avoid coarse misjudgements. The accuracy by which the location is expressed varies depending technology in use. For full functionality, we require location information which error marginal is within one to three meters. The most axiomatic way to indicate location is to use longitude and latitude coordinates, which are used here. The users of advanced mobile devices, with accurate positioning and graphical interfaces, may then be e.g. provided with a map where the relevant locations are highlighted - or even a compass-like arrow guiding them to the recommended places. For inaccurate positioning, such as the the nearest GSM base station, the abstraction *Area* is introduced. An area can indicate e.g. a part of the city, when traffic signs and regional information combined with food place's address guide the user the food place's location unequivocally in terms of community planning. Even if there's no GUI (as with many WAP-clients and e.g. speech interfaces), the name of the area and the street address are usable.

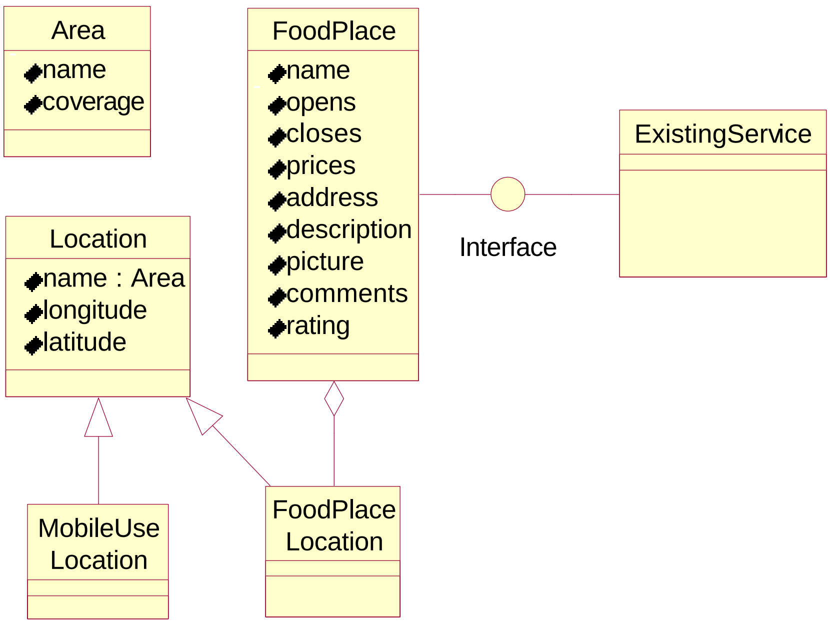



Figure 8. Final Conceptual Model.

The polished conceptual model is depicted in figure 8. Food place has a geographical location for evaluating user's travelling time to the food place and showing the exact position when possible, and regional location so that it can be found without a special equipment.

#### Design of nodes, contexts and navigation

In terms of our own adaptive hypermedia system design process model (see figure 3), we have designed adaptors, shadows and the system's information content. Design tasks still uncompleted are design of nodes, design of contexts, and design of navigation. (Note that there is an user interface design task, too, but we decided to exclude it from this paper.) These tasks will be completed next in the form *of Adaptive Navigational Structure Model and Adaptive Navigational Class Model* using the UML-based method presented in (Hennicker & Koch 2000). In OOHDM, the respective design phase is navigational design, and the corresponding diagrams are called the navigational class diagram and the context diagram (Schwabe&Rossi 1998). Besides nodes and links, also access structures (such as indices) are designed in this phase, and here they are our first concern.

Two kinds of queries are needed: the one which lists maximum of ten priorised food places and the one which fetches information attached to a single food place, to a food place which is next to the user (if one is really close, as defined in the shadow AtDoor).

The first query satisfies the needs of an user who is requesting food place recommendation, where as the second query services user who is by a certain place. The navigational structure model depicted in figure 9 specifies this design. In the diagram, we use the stereotypes for query (the questionmark), index (with the little box with lines) and navigational class (with the empty box) as defined in (Hennicker & Koch 2000).

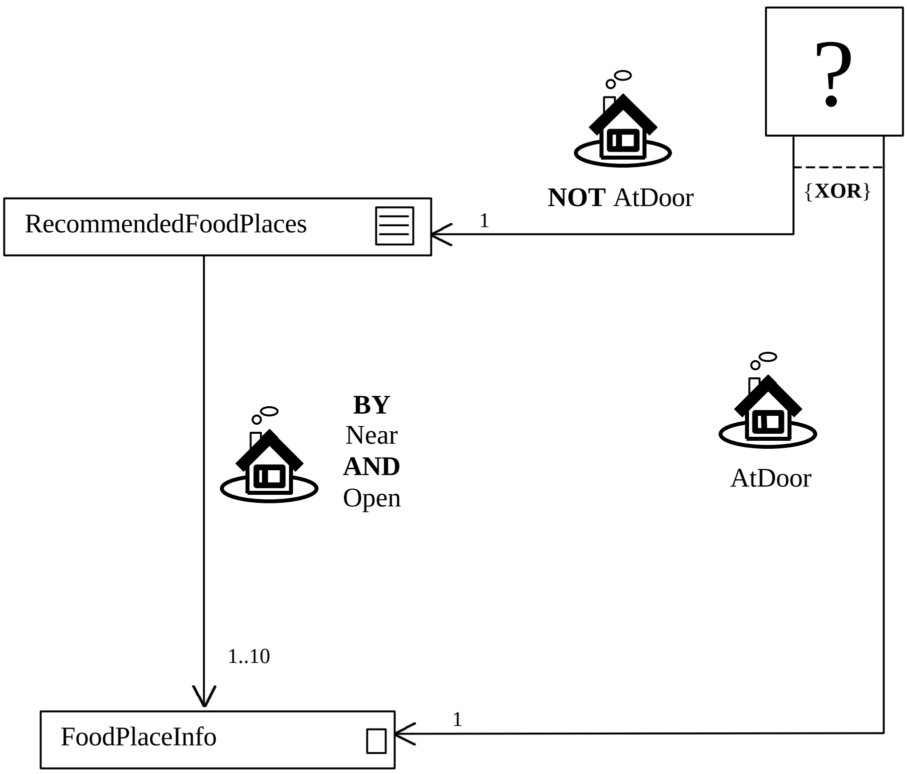



Figure 9. Adaptive Navigational Structure Model (or Adaptive Context Diagram).

Choice of a query depends on user's location. If shadow *AtDoor* is true, system displays information about food place which is next to the requesting user. If shadow *AtDoor* is false (there is no food places nearby), a list of food places is constructed. This list is ordered by by shadows *Near* and *Open* so that food places which are near the user and which are currently open appear on the top of the list.

Index *RecommendedFoodPlaces* and node *FoodPlaceInfo* need to be elaborated. We have done this in a way that can be seen in figure 10., again following (Hennicker & Koch 2000).

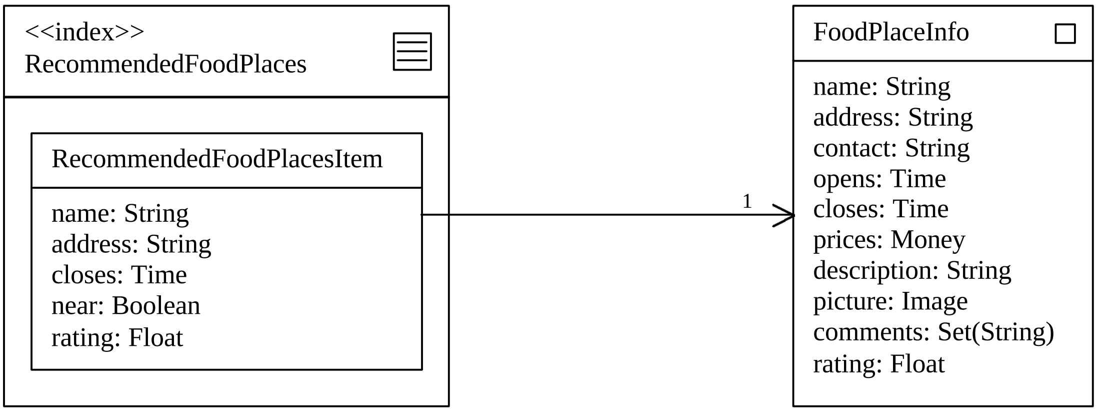



Figure 10. Adaptive Navigational Class Model.

Index *RecommendedFoodPlaces* consists of *RecommendedFoodPlacesItem* items. Each item contains food place's name, postal address, closing time, rating, and a piece of information indicating if food place is somewhere near. Link connecting item to a node (*FoodPlaceInfo*), containing detailed information about food place, is anchored to the food place's name (which is the default behaviour defined in (Hennicker & Koch 2000)).

The very same node (i.e. navigational class *FoodPlaceInfo*, with the content from the existing service) is pointed to by the query which displays information about food place whereby the user is. This completes the location- and time-adaptive design of the navigation for mobile use of the existing food place service.

#### User Interface considerations

The specification of the actual user interfaces for the system is outside the scope of this paper, but here are some remarks concerning especially mobile use and the assumptions made in the work. The mobile devices are expected to be of various kinds, and used in many different ways. The examples already mentioned in the description of the work done: PDAs with Web-browsers, phone-embedded WAP-clients and voice interfaces (e.g. over a phone line while driving a car), already have very different capabilities. The diversity may grow even greater with the advent wearable computing (possibly augmented reality goggles) or, on the other hand, the disappearing interfaces of ubiqutous computing. The hypermedia design models used here separate the content, navigational structures (e.g. nodes as views to databases) and the presentation for the user, enabling user-interface metaphora independent design. E.g. the system at hand here uses existing content, adds what's needed to support mobile use, and allows for different user interfaces.

## Discussion

Here we have presented a case where there is adaptivity in a hypermedia application without modelling the user, using the location and time information instead. Admittedly, however, user oriented adaptation would be interesting in this case, too. For example users could specify their individual tastes regarding the food, or preferred areas/locations and perhaps eating times - or the system could be set up to learn them (semi)automatically based on the behaviour of the users. Also social (collaborative) filtering could be used to e.g. acquire recommendation based on ratings by users with similar tastes etc. This point of not forgetting the user, but modelling them as well as the context, has been elaborated elsewhere (Jameson 2001). On the other hand, it can be argued that the location information as used in our example is user information - it is the current location of the user, after all. But the emphasis here is that the system keeps no record of the usage, and none of the adaptation is done with regard to any personal characteristics - just the use location. This makes a point when concerning privacy, which is often a hot topic when location information is used. In this example, the location information is never connected to the individual.

Thinking of the design approach, and specifically the modelling method, there are many questions. What is the actual scope of these models - do these diagrams capture the essence of the design problem at hand? In the end, after trying several different ways (even an activity diagram), we would argue that the solution here is an elegant one and our previously presented addition to support modelling adaptivity - the shadows - integrate well, complementing the UML-based notation for the OOHDM-like modelling presented in (Hennicker & Koch 2000). Compared with the original example in (Alatalo&Peräaho 2001) a notable point is that there we modelled adaptive content (a single node) whereas here there's adaptive navigation, too (thanks to the flexibility of UML class diagrams that can be used for both purposes when modelling hypermedia).

The results are limited as the case is still a simple one, if not trivial, but we are already working on a considerably more complex application. Another remaining issue is the relation of our approach to previous ones, such as the Adaptive Hypermedia Architecture Model (AHAM, De Bra, Houben & Wu 1999).

In the future, there are additional challenges still in achieving more complex ("deeper") adaptivity, perhaps along the lines explored in an attempt to create open-ended semiosis for adaptive knowledge management (Rocha 2000). A potential approach for designing complex adaptivity may also lie in multi-agent systems, and from a modelling perspective the proposed Agent-UML extension is an interesting one (Odell, Van Dyke and Bauer 2000). There even the concept of emergence has been noted, but no way to actually design it proposed. The work on open-ended semiosis mentioned above, perhaps combined with the recent development in understanding the underlying monumental works of von Neumann on evolution of complexity (McMullin 2000), may offer solutions. In the meanwhile, the more basic kind adaptivity - such as the simple example here - combined with good design, is certainly enough to keep us busy.

## Conclusion

In this paper, we have used a previously presented approach for designing adaptive hypermedia to address the requirements set by mobile use. It was shown, that the method could be used to model specifications for a time and location adaptive system too, although it was originally illustrated with a user adaptive (e.g. personalization) example only. This also makes a case concerning the definition of adaptive hypermedia, which currently includes user modelling only. Although the adaptivity in this case is also user-centered, in the sense that the hypermedia navigation changes according to the location of the user, it may be argued that it is not user adaptivity, as there in fact is no user model and none of the behaviour depends on any characteristics of individual users (that are not even recognized). This emphasis on *context awareness*, (being in this case specifically *mobile aware*), is a strong trend in information systems and human-computer interaction design, with hopefully fruitful connections with the hypermedia community in the future - not forgetting the user modelling aspects already elaborated in adaptive hypermedia, either.

## References

Alatalo, T. & Peräaho, J. (2001). A Modelling Method for Designing Hypermedia. A position paper for the Third Workshop on Adaptive Hypertext and Hypermedia, the UM2001 session. <http://owla.oulu.fi/publications/modelling_method_for_AH-position1_AHWS2001.pdf>

De Bra, P., Houben, G. J., Wu, H. (1999). AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. The 10th ACM Conference on Hypertext and Hypermedia, Hypertext '99.

Bulterman, D. C. A., Rutledge, L., Hardman, L.,van Ossenbruggen, J. (1999). Supporting Adaptive and Adaptable Hypermedia Presentation Sematics. The 8th IFIP 2.6 Working Conference on Database Semantics (DS-8): Semantic Issues in Multimedia Systems. January 5-8, 1999.

Dahlbom, B. & Ljungberg, F. (1998). Mobile Informatics. Scandinavian Journal of Information Systems. Vol. 10, no. 1&2.

Gargan Jr, R. A., Sullivan, J. W., Tyler, S. W. (1988). Multimodal Response Planning: An Adaptive Rule Based Approach. CHI '88 Conference Proceedings. May 15-19, 1988.

Hennicker, R. & Koch, N. (2000). A UML-based Methodology for Hypermedia Design. In A. Evans,

S. Stuart, and B. Selic, editors, UML'2000 - The Unified Modeling Language - Advancing the Standard, volume 1939 of Lecture Notes in Computer Science, York, England, October 2000. Springer Verlag.

Hjelm. J. (2000). Designing Wireless Information Services. New York: John Wiley & Sons.

Jameson, A. (2001). Modelling Both the Context and the User. Personal Technologies, 5, 1.

Koch, N. (1998). Towards a Methodology for Adaptive Hypermedia Systems Development. In *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Proceedings of the* 6th Workkshop, ABIS-9*8,* U. Timm and M. Roessel (Eds.)

Korkea-aho, M. (2000). Context-Aware Applications Survey [online]. Mari Korkea-aho. <http://www.hut.fi/~mkorkeaa/doc/context-aware.html>.

Lieberman, H. & Selker, T. (2000) Out of Context: Computer Systems that Adapt To, and Learn From, Context. CHI 2000 Workshop Research Directions in Situated Computing <http://www.daimi.au.dk/~mbl/chi2000-sitcomp/pdf/Cheverst.pdf>

McMullin, B. (2000) John von Neumann and the Evolutionary Growth of Complexity: Looking Backwards, Looking Forwards... Presented at Artificial Life VII August 1-6, 2000, Portland, Oregon. <http://www.eeng.dcu.ie/~alife/bmcm-2000-01/>

Odell J., Van Dyke Parunak H. and Bauer B.(2000) "Extending UML for Agents". AOIS Worshop at AAAI 2000, Austin, TX, 2000.

Rocha, Luis M. (2000). "Adaptive recommendation and open-ended semiosis". International Journal of Human-Computer Studies. (In Press) <http://www.c3.lanl.gov/~rocha/ijhms_pask.html>

Roque, L. & Almeida, A. (2000) Towards the Design of Context. CHI 2000 Workshop Research Directions in Situated Computing. <http://www.daimi.au.dk/~mbl/chi2000-sitcomp/pdf/Roque.pdf>

Schwabe, D. & Rossi, G. (1998). Developing Hypermedia Applications using OOHDM. Workshop on Hypermedia Development Processes: Methods and Models, Hypertext'98. <http://www.inf.puc-rio.br/~schwabe/papers/ExOOHDM.pdf.gz>

da Silva, D.P., Van Durm, R., Duval, E., Olivie, H. (1998). Adaptive Navigational Facilities in Educational Hypermedia. The Ninth ACM Conference on Hypertext and Hypermedia. Hypertext 98. June 20-24, 1998.