Towards Real-Scale Business Transaction Workflow Modelling*

A.P. Barros[1], A.H.M. ter Hofstede[1], H.A. Proper[2]

[1]Department of Computer Science
The University of Queensland
Brisbane, Qld 4072
Australia

[2]Faculty of Information Technology
Queensland University of Technology
GPO Box 2434, Brisbane 4001
Australia
e-mail: E.Proper@acm.org

PUBLISHED AS:

A.P. Barros, A.H.M. ter Hofstede, and H.A. Proper. Towards Real-Scale Business Transaction Workflow Modelling. In A. Olivé and J.A. Pastor, editors, *Proceedings of the Ninth International Conference CAiSE'97 on Advanced Information Systems Engineering*, volume 1250 of *Lecture Notes in Computer Science*, pages 437–450, Barcelona, Spain, EU, June 1997. Springer Verlag, Berlin, Germany, EU. ISBN 3540631070

Abstract

While the specification languages of workflow management systems focus on process execution semantics, the successful development of workflows relies on a fuller *conceptualisation* of business processing, including process semantics. For this, a wellspring of modelling techniques, paradigms and informal-formal method extensions which address the analysis of organisational processing structures (enterprise modelling) and communication (based on speech-act theory), is available. However, the characterisations - indeed the cognition - of workflows still appears coarse. In this paper, we provide the complementary, empirical insight of a real-scale *business transaction* workflow. The development of the workflow model follows a set of principles which we believe address workflow modelling *suitability*. Through the principles, advanced considerations including asynchronous as well as synchronous messaging, temporal constraints and a service-oriented perspective are motivated. By illustrating the suitability principles and with it the inherent complexity of business transaction domains, we offer timely insights into workflow specification extension, and workflow reuse and deployment.

1 Introduction

The workflow concept, proliferated through the recently emergent workflow management systems (WFMS) (see surveys in [FYW94, WW93, Rod91, GHS95]), advances information systems (IS) implementation models by incorporating aspects of collaboration and coordination in business processes.

*Part of this work has been supported by CITEC, a business unit of the Queensland Government's Department of Public Works and Housing (formerly the Administrative Services Department).

Under traditional implementation models, applications are partitioned into discrete units of functionality, with (typically) operational procedures used to describe how human and computerised actions of business processes combine to deliver business services. Through an endowment of business process execution semantics, workflows permit a greater organisational fit of ISs. Moreover workflows are specified at a level above traditional applications, enabling program binding and access to a loosely-coupled set of databases and files. Therefore, newer applications may be developed out of existing applications to reflect reengineered business processes.

WFMS, e.g. IBM's FlowMark [LR94], allow workflows to be specified and, based on these, provide execution scheduling and run-time event monitoring. As evident in the standardisation of workflow and WFMS concepts, undertaken by the Workflow Management Coalition[1], the focus of workflow specifications is on work *coordination* and less on work specification. Put simply, workflow specifications, as they are currently deployed, are implementation-oriented. The *implementation* level is, of course, orthogonal to that level of specification which deals with the cognition and analysis of business domains, i.e. to the *conceptual* level. The conceptual level is renowned as crucial since its focus on essence facilitates the early and necessary problem solving prior to the later and more error-retentive implementation phases [Dav90]. In addition to *conceptualisation*, effective conceptual modelling techniques provide: a high degree of expressive power thereby minimising/defeasing specification *waterfalls*; an effective *comprehensibility* so that specifications can be developed and communicated with a diverse set of stakeholders; a *formal foundation* whereby both the syntax and semantics of a technique are clear. Also since a "silver bullet" for all domains is no longer considered realistic (see e.g. [BS87, Bro87, ML83]), a *suitability* for its problem domain is required, meaning that a technique's concepts and features should reflect closely those of the problem domain.

Engaging larger "chunks" of business domains into ISs which workflows over and above previous implementation models do, presents a major uncertainty. What characterises a suitable cognition of workflows in the business domain? This uncertainty is evident in the number of different paradigms which have been adopted by workflow-applicable techniques: process-centric, e.g. [DB91]; state-centric, e.g. [DP95]; and actor-centric, e.g. [Die94] (based on the speech-act theory synthesis of [FL80]). In addition to the different paradigms, proposals have been introduced for a greater cognitive effectiveness of methods including integrated enterprise modelling to formalise requirements engineering [BB95, LK95, AMP94] and to enhance IS design mapping, e.g. [Ram94]. Moreover, as identified by the FRISCO group[2], conceptual modelling concepts and techniques are ultimately *constructivist* by nature, warranting on-going reviews and empirical insights for on-going extensions.

The goal of this paper is to provide an empirical insight into the modelling extensions for what we believe to be a common and frequently encountered type of business processing workflow. This relates to operational *business transactions*. As evident from the domain we present, operational business transactions are characterised by a client, a server organisation and supplier (of goods and services to the server) organisations, with multiple interactions between these involving more coordinative processing - typically message (document) passing - and less collaborative processing. In the widely cited characterisation of workflows [GHS95], operational business transactions correspond to *administrative* and *production* workflows. The extensions are structured into a framework of suitability principles which we identified through an assessment of integrated techniques in [BHPC96]. These are: the organisational embedding of conceptual models; their validation (as a broad unit of cognition) through

[1] Refer to http://www.aiai.ed.ac.uk/WfMC/index.html for more details.

[2] FRISCO is an IFIP WG 8.1 Task Group FRISCO, commissioned to create a FRamework of Information Systems COncepts.

scenarios; the insulation of business service requests from the resultant business processing, and its motivation for service modelling within workflow specifications; the incorporation of sometimes neglected aspects, e.g. combining structural and behavioural aspects of processes, interactive aspects, in particular, human to computer interaction (HCI), and temporal aspects, to enhance a workflow's cognition; and the explicit treatment of operational error handling for a workflow's execution resilience. Given its illustrative goal, the paper is not so much concerned with proposing any technique as with the issues of extending techniques to increase their suitability, given a real-scale business transaction domain.

The paper is organised as follows. In section 2, an overview of the road closures domain is presented. In section 3, the highlights of the proposed workflow model are presented along the lines of the suitability principles. In section 4, the paper is concluded with a summary of the findings and open research issues.

2 Case study overview: road closures

As a consequence of the Westminster System used in Australia, the government administration of land falls under a number of statutes (or legislative acts) which involve a number of statutory authorities. In Queensland, the State Government's Department of Natural Resources under the Lands Act[3] is commissioned to grant *tenure* for unallocated state land and reserved land. In a broad sense, this includes: the granting of ownership through freehold titles (i.e. privately owned); the granting of custodianship for some purpose through leasehold titles - leases; the establishment of reserves for national parks and wildlife etc.; and the dedication of roads (which by definition in the Land Act implies public use). These apply to *parcels* of land which are composed of one or more elementary allotments, or *lots*.

In order to grant tenure, the Department of Natural Resources obtains the views of the relevant stakeholders - statutory authorities, affected landholders and affected associations in the community - in order to determine whether proposed and potential use of the parcel affects the surrounding area's current and future land use and the current legislation. The statutory authorities include local governments, electricity power suppliers, telecommunications carriers and environment and heritage regulation authorities.

The process of determining whether tenure should be granted is complex taking from days up to months. During this period, repeated checks are required to ensure that the appropriate requirements are satisfied. These are needed since different actions constantly occur on related aspects of land. For example, during the process of investigating whether a mining lease should be granted, an overlapping part of the land may become heritage-protected while another overlapping part may be needed for a railway corridor. Clearly, the three tenures may result in incompatible land use. Furthermore, repeated interaction with the different stakeholders may be necessary to resolve unsatisfied requirements. During this period also, business processes may change due to changes in legislation as well as in organisational restructures. In short, an effective coordination of processing requires: the integrity of tenure grants to be preserved; insulation from business process change; a minimisation of customer interaction.

The closure of roads is a particular instance of tenure allocation. Under the Lands Act, the Minister for Lands approves road closures. This responsibility may be delegated to specific persons. Apart from

[3] Lands Act 1994; i.e. last issued in 1994.

the legal reasons, the approval (and for that matter the disapproval) of road closures carries important ramifications. For one, the general public have certain rights and expectations to roads. A road should not be closed for reasons which include the current or potential blockage of dedicated access to another parcel(s), the compromise of the transportation network, environmental degradation (e.g. some roads form important corridors and refuges for flora and fauna) and the existence of Native Title[4].

A road may be closed permanently or temporarily. If permanent, it may be subsumed into one or more adjoining parcels. A subsumption into a freehold tenured parcel involves a *Surrender* of the existing title, i.e. a *Certificate of Title*, and the issuing of a new title, i.e. a *Deed of Grant*[5] over the "new" parcel. A subsumption into a leasehold tenured parcel involves an *Adjustment* of the existing lease. For temporary closures, a *Road License* or a *Permit to Occupy*, may be issued over an area, where the State retains the right to re-dedicate the area at any subsequent time. In the case of a Permit to Occupy, the road does not lose its status and the public's access cannot be impeded completely, e.g. a side walk cafe. Roads may be created into parcels without subsumption, in which case a Deed of Grant is issued.

Figure 1 contains the highest level data model expressed using the Conceptual Data Modelling Kernel (CDM) [CP96] using the Object-Role Modelling "flavour" (see [Hal95] for more details). Unlike normal ORM object types, CDM allows object type decompositions, as is the case for all the illustrated object types.

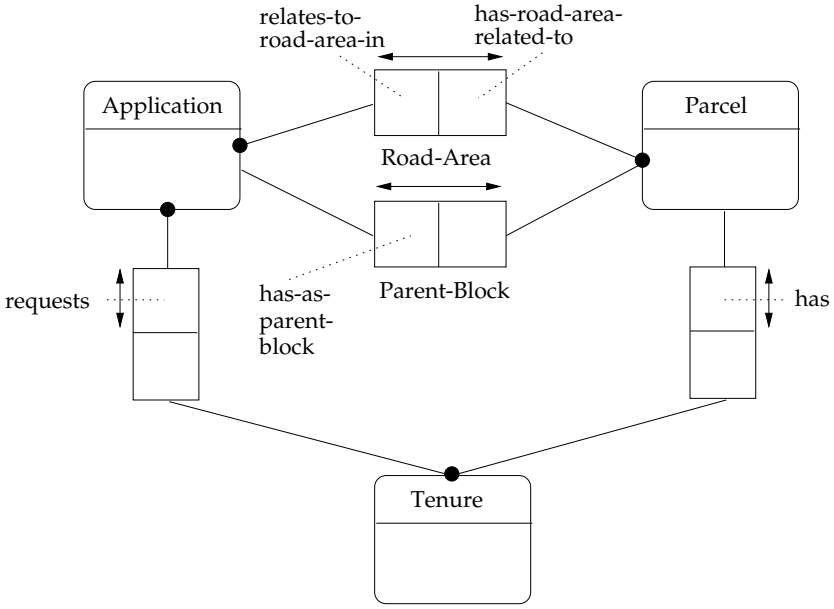

Figure 1: Highest level object model for Road Closures

None of the decompositions other than that for Application in Figure 2 are shown.

[4]Under the Lands Act, parcels without any allocated tenure are deemed to have Native Title, i.e. their use is determined by the Aboriginal people of Australia.

[5]A Deed of Grant is a title which signifies the first tenure of a parcel.

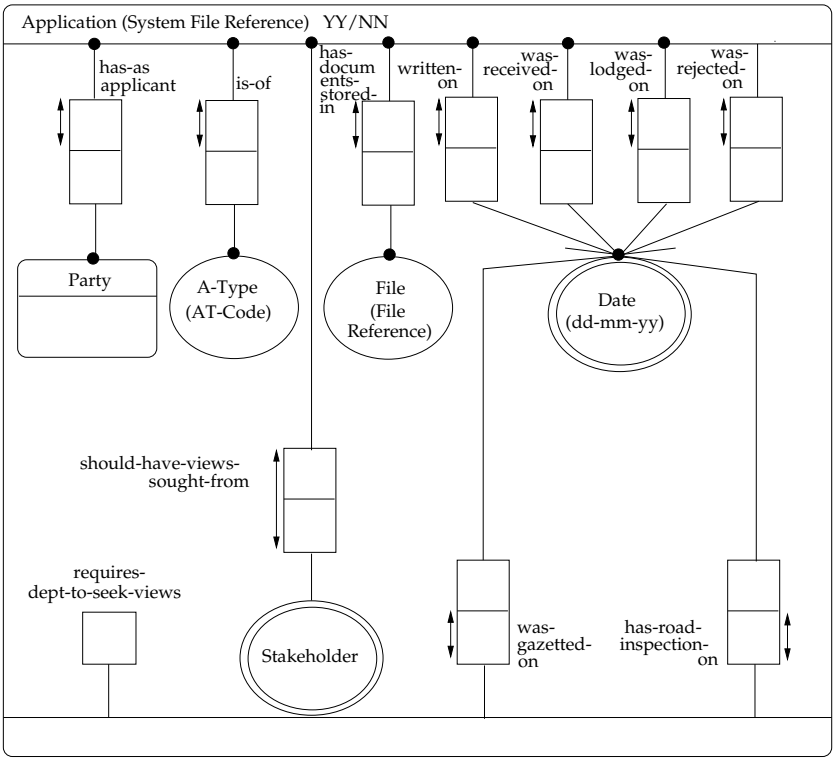

Figure 2: Decomposition of Application

3 Case study workflow modelling

Having described the case study overview, the purpose of this section is to provide an insight into conceptual workflow modelling for domains involving real-scale business transaction processing. This illustrative goal features on the one hand, how workflows can improve this processing and what useful core concepts, elicited from existing techniques, contribute to workflow solutions. Yet on the other hand, an appreciation of the current gaps and uncertainties in workflow specifications is also intended to be conveyed. This aspect is structured into a framework of suitability principles identified as a result of an assessment of techniques conducted in [BHPC96]. Only the highlights of the case study are presented (see [BH96] for full details).

3.1 Organisational Embedding

The organisational embedding of the workflow, in accordance with the *Organisational Embedding Principle*, is not illustrated. It requires that "a technique should embed all concepts in a conceptual model, directly or indirectly, but without redundancy, into organisational concepts". Typically organisational processing structures, as captured in enterprise models, are refined down to an IS modelling level. Although appearing to state the obvious, the principle does not preclude networked decompositions as appears evident in integrated techniques, e.g. [Ram94], and CAiSE tools, e.g. AD/CYCLE [MMNR90].

For the road closures domain, two high-level business processes, Application Lodgement and Application Investigation have a direct correspondence in the highest level abstraction of the workflow. Two similar business processes in different parts of the department are classified into a third process, Road Closures. From these, necessary actor roles and accessed information repositories are determined; a further basis for organisational embedding. For the road closures, as with other workflows ancillary to it, whether internal or external to the department, service interfaces are created. The reason for this will become clearer later, however for now it should be understood that this provides an organisational embedding for business services. In other words, for the sorts of workflows we are concerned with, business services encapsulate workflows allowing considerably simplified workflow access, globalisation and reuse.

3.2 Scenario Validation

As a further refinement to the scope of the business domain, a more specific articulation of the business processing is required since the same business processes may be used in different workflows. This is clearly evident for Application Lodgement, and Application Investigation which are general for tenure processing. While it may be argued that the specialisations of the main object of interest - tenure Applications - be used to determine the different workflow scenarios, this is restrictive since it is based only on a structural classification. Workflows could also be classified around the types of a set of triggering events - a dynamic classification.

As a general requirement, the *Scenario Validation Principle* states that "a technique should provide an explicit notion of *scenario* for model validation". Validation is concerned with the interpretation of a conceptual model's domain semantics (as distinct from verification which is concerned with its formal semantics). Validation, in its broadest sense implies a mechanism for cognition, and therefore applies to workflow development. There could be many scenario *types* embodying different types of business processing. For example, the type of business processing involved in a senior manager's request results in a more ad-hoc and less precise processing pattern whereas operational business transactions such as road closures are well-understood and more precise in processing nature. It should be clear what types of scenarios a technique should have, in addition to having a suitable set of concepts for those types. We believe that the identification of workflow scenario types in tandem with their modelling suitability will form a major part of future workflow research.

3.3 Cognitive Sufficiency

Following from the previous principle, the *Cognitive Sufficiency Principle* relates to the inclusion of all the concepts which "provide a sufficient cognition of a model such that no assumptions about fundamental aspects of business processing *execution* semantics are required". Notable areas of collective variance in process and workflow modelling techniques, which we identified in [BHPC96], include: data (messaging) *and* control (triggering) flows, human-to-computer interaction points (dialogues) and temporal aspects. These are bourne out by the proposed process models for the road closures business transaction which follow.

3.3.1 Application Lodgement

In Application Lodgement depicted in Figure 3, Application Documents containing a letter and attached documents are lodged at a Service Centre. A Service officer - a particular actor *role* - is responsible for the Application Entry. An Application File, formed to contain the documents, is sent to the relevant Regional Office where it is filed away in the Application Files by Store Application.

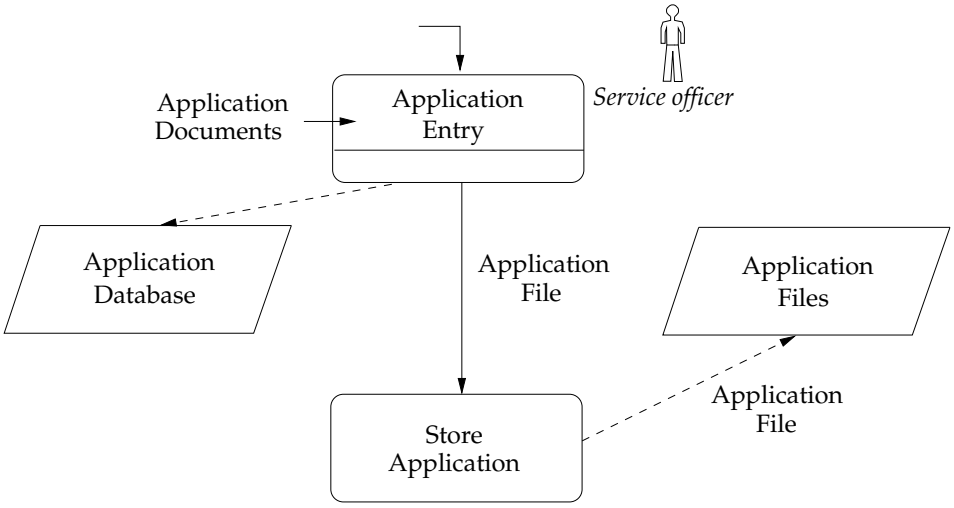

Figure 3: The Application Lodgement process model

The process model resembles in part a Task Structure as introduced in [Bot89], extended in [HN93] and placed into an integrated modelling kernel (with an ORM-like data modelling technique) in Hydra [Hof93]. Processes (boxes) including an initially executed process (denoted by the bent arrow), together with execution triggering (arrows between processes) are shown. Included in process specifications are pre- and post-conditions and component actions for database access. In Hydra, a conceptual specification language LISA-D (see [HPW93]) enables this, given the tight-coupling with data models.

It may be seen that structured process modelling, e.g. the well-known Data Flow Diagrams [You89], concepts such as data flows or *messages*, e.g. Application Documents, add to the cognition of a business transaction's execution semantics. This is because the attributes of the incoming (outgoing) message's type may overlap the attributes of other message types; hence the message type qualifies the pre- (post-) condition further. With the introduction of messages comes the requirement of *messaging*, i.e. information passing. In this case an incoming message is indicated (a small arrow embedded in a process). Also the message Application Files is transferred, seemingly with a trigger. This is just an adopted notational convention since the semantics of messaging are quite different to those of triggering. Messages, afterall, are actually deposited in some container, e.g. an in-tray or a mail-box, from where they are retrieved (perhaps in some predefined order) by the target process. Application Files has been modelled as an object store because of its "persistent" rather than temporary storage nature. Application Database is obviously a computerised object store, and access to it can be formulated in LISA-D given its related schema assignment (recall Figure 2. The inclusion of HCI points (by a bar in a process) provides a further enhancement to execution cognition.

3.3.2 Application Investigation

A more complicated process model is depicted in Figure 4 for Application Investigation. In brief, it consists of a number of internal checks to determine whether the Application is valid, a Preparation for a more detailed investigation, Seek Views and Process Views of the Stakeholders as a part of the detailed investigation, and a Site Inspection as another part. A decision is then made to Approve (an) Offer which if negative, results in either a rejection or request for further information/action through Suspend Processing, or if positive results in a preparation to make the offer through Effect Offer Approval. In the description that follows, only Initial Review Passed?, Preparation and Seek Views are further elaborated on.

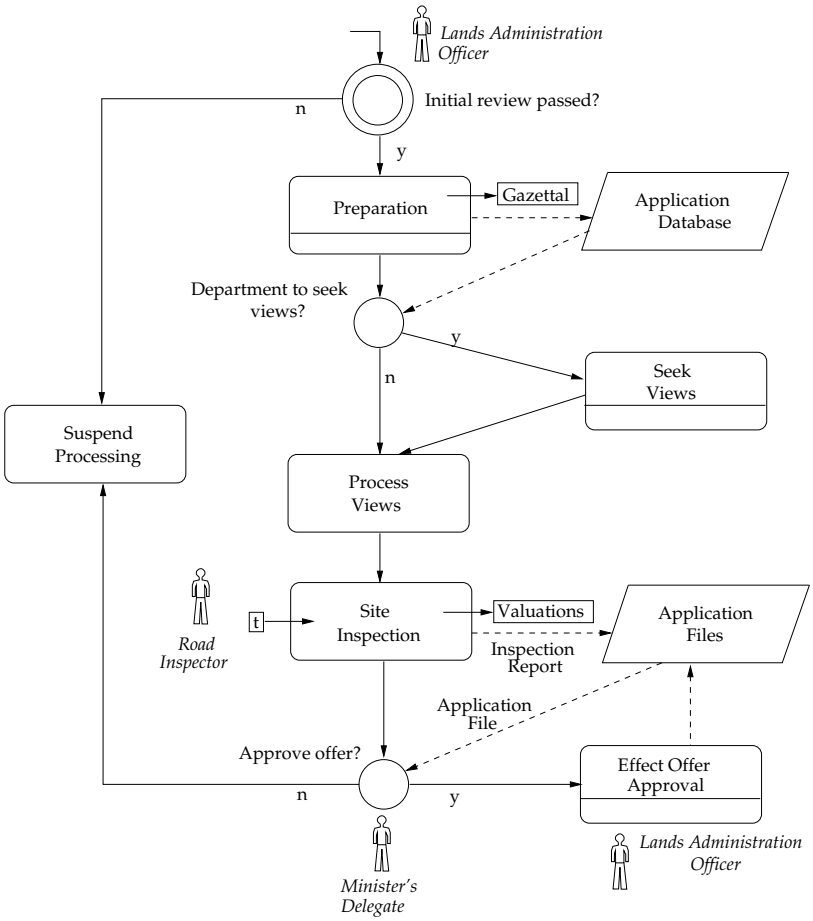

Figure 4: Decomposition of Application Investigation

The example of the internal checks presents the need for an extension to decision handling in traditional modelling. Under Hydra Task Structures for example, decisions yield either a positive or negative outcome, given their rules. Moreover an outcome can terminate execution, returning control to the supertask. It is evident through this part of road closures, as depicted in Figure 5 that decisions in real-scale business transactions may be based on sub-decisions - possibly a whole network of decisions with their own execution dependencies. In this case, the decisions are all *simple*, are executed in parallel and have an implied synchronisation of their outcomes. A network of decisions - simple or *complex* -

have execution triggers between them. We propose terminating *aborts* for decision outcomes to extend expressive power, since the outcome of a given decision, of itself, may be satisfactory to determine the outcome of the complex decision.

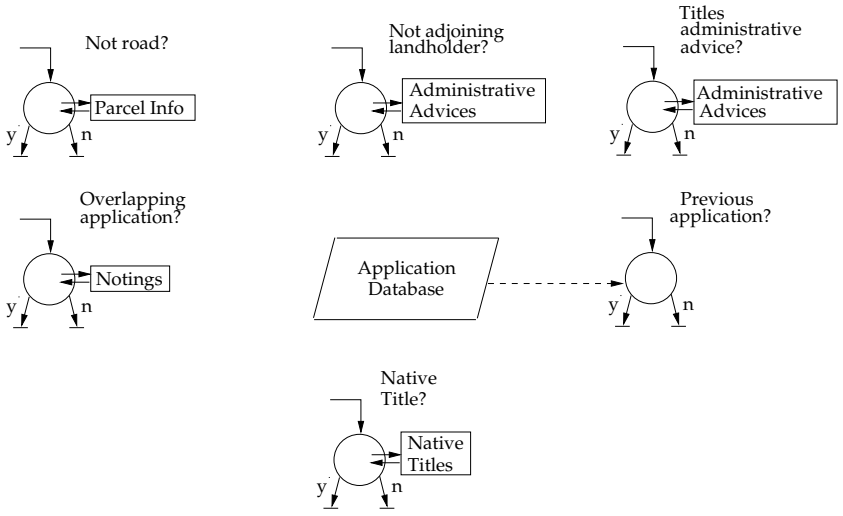

Figure 5: Decomposition of Initial Review Passed

A further extension to decision handling is the accommodation of messaging. In Figure 5, most decisions require data from messages for the decision rules. Also the messaging of "remote" services (boxes attached to the messaging arrows) is illustrated. Unlike the previously discussed form of messaging which was *asynchronous*, the depicted messages are *synchronous*. That is, a message is sent out and an incoming message is anticipated (hence two embedded arrows). From the time that the message is sent out to the time that the message is received, no execution proceeds. Highly expressive conceptual specification languages allow quite sophisticated rules to be formulated, as is evident in the following LISA-D formulation for the positive outcome of Previous Application? (assuming a two year threshold):

Application(has-as-parent-block Parcel **CONTAINING** Lot
elementary-surveyed-unit-of Parcel
has-road-area-related-to Current-App
AND ALSO
received-on Date $<$ Date marks-receipt-of Current-App
AND ALSO
received-on Date $\geq$ Date marks-receipt-of Current-App $-$ 2 years)

Like a simple decision, the result of a complex decision is either a positive or negative outcome. A negative outcome results in the execution of Suspend Processing which is not depicted. In brief, it either results in a rejection of the application or a request for further information. In either case, the appropriate notifications are sent out to the Interested Stakeholders (i.e. those who provided some form of response to the notifications of road closure intentions). In the case of rejection, when the last of the notifications is sent out, an abort message is raised so that all active processes involved in processing this application are closed. Abort handling, generalised into operational error handling, is the subject of the *Execution Resilience Principle* (see section 3.5).

Returning to Initial review passed? in Figure 4, its positive outcome results in the Preparation for a detailed investigation of the Application. This involves publication of road closure intention in the Government Gazette; done through the Gazettal service (external to the department). The department seeks the views of stakeholders if it is required to do so. This is done through Seek Views, depicted in Figure 6, is executed.

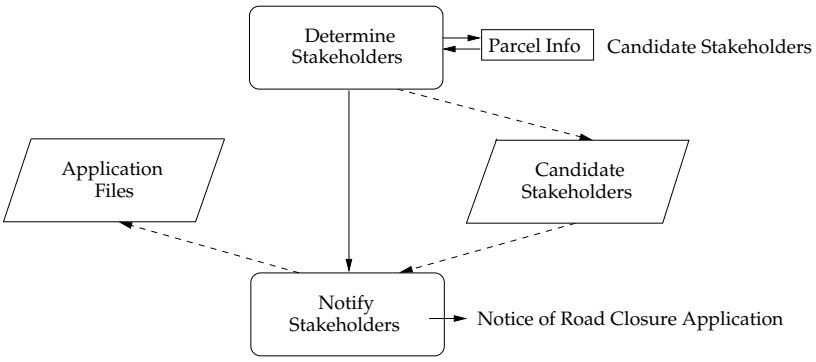

Figure 6: Decomposition of Seek Views

First the Candidate Stakeholders need to be determined. These are obtained through Parcel Info (an external service which accesses a Cadastral database identifying the surrounding parcels, utilities etc.). Then the contents of the message needs to be inserted into Candidate Stakeholders object store. Rather than place a HCI point Determine Stakeholders, the whole process by ensuring that the message's schema, depicted in Figure 7 is covered by the object store's schema:

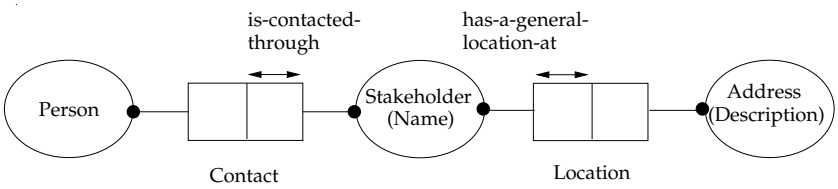

Figure 7: Schema associated with the Candidate Stakeholders message

Then the following LISA-D update becomes possible (Contact and Location, denotations of fact types, are the same for the object store schema):

ADD Contact **IN** Candidate Stakeholders **TO** Contact
ADD Location **IN** Candidate Stakeholders **TO** Location

A Notice of Road Closure is then sent to each Stakeholder, illustrating how *bulk* messaging can be incorporated in LISA-D:

SEND Notice of the Road Closure **TO EACH** Person is-contact-for Stakeholder s
AT Address is-contact-for Stakeholder s

Now the sending of the messages is required to occur no later than one day after the date of gazettal. This illustrates the need for a *temporal constraint* in the postcondition in Seek Views:

END-DATE(Seek Views) $\leq$ Date is-gazetted-date-of Application Curr-Application $+ 1$

As an example of a temporal constraint on preconditions, a Site Inspection is not allowed to occur more than prior to two months before the intention for road closure has been "gazetted":

START-DATE(Road Inspection) $\geq$ Date is-gazetted-date-of Application Current-App $+ 2$ months

START-DATE and **END-DATE** indicate the need for temporal functions which provide the start and end dates of process object execution. This implies that certain execution statistics about process objects should be maintained. This allows time durations to also be used within constraints, for example, for "timeouts". Also process execution dependency can further be qualified through temporal constraints. For example: run a number of processes at some time, simultaneously (parallelism); or within a time duration of each other (sequence); run a process repeatedly within a certain time period or cyclically at time points (repetition). Such constraints can apply to messaging as well, e.g. contingent service access for process objects if messages have not returned within certain times. It is striking to note for the specifications of other types of domains considerable treatment of temporal aspects is available, e.g. in model-based formal specification languages e.g. PAISley [Zav86], algebraic specification languages e.g. Real-Time Process Algebra [BB91], and Petri net based approaches e.g. ExSpect [HSV89]. For business domains, however, little evidence exists for temporal *processing* constraints (for an indication see the survey in [TL91]). For any such extensions of workflow specifications, it should be noted that determination of completeness in a temporal constraints language is an open issue. Despite this, we believe that with the increased use of workflows and distributed service access (e.g. an open distributed Traders environment), temporal aspects particularly those involving temporal durations will increasingly become important.

3.3.3 Road closure

Road Closure is the last of the top-level processes. Its decomposition, depicted in Figure 8, demonstrates two needs for *execution synchronisation*. Firstly, the synchroniser construct (a triangle) may be used to ensure that prior to subsequent processing, a number of triggers should be reached (the first synchroniser of Figure 8). Secondly, when two or more execution paths should be simultaneously (but time independently) started (the second synchroniser of Figure 8). Using this, a "fork" operation can be specified (where one path returns control to the source execution point).

3.4 Service Information Hiding

A key distinction in business transactions is that between business services and business processes. This is, in fact, a generalisation of the distinction between events and processes. An event, afterall, is associated with some intention, and more than one event may share the same intention. In a business sense, intentions are aggregated into business services. As such, business services are a described (external) organisation of functionality which do not do anything as such, other than being associated with process interactions. This may involve external access such as client requests and responses from outside

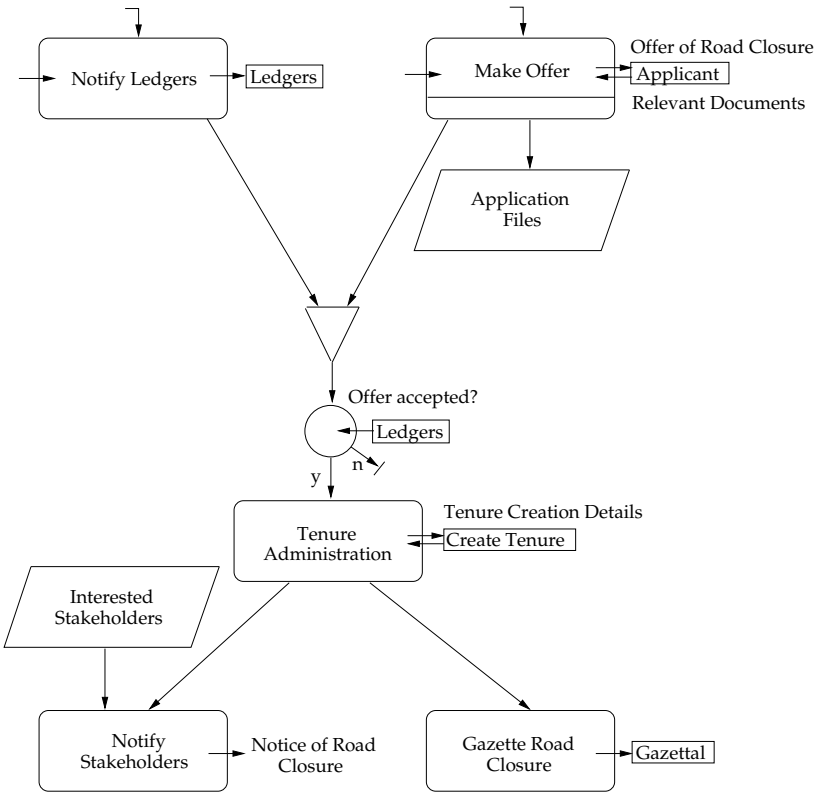

Figure 8: Decomposition of Road Closure

organisations, or internal access to services in different parts of an organisation. Inherently, they have a set of *states*, e.g. initiated, processing, rejected, and each state is associated with, and dependent on, a particular course of action resulting from a particular event. Business processes on the other hand, are a prescribed (internal) organisation of functionality reflecting the mechanisms by which business services are delivered. Unlike services, they perform concrete actions, (e.g. data transformations, updates and retrievals) and their states are (relatively speaking) dependent on the success of their processing.

The *Service Information Hiding Principle* requires that "a technique should allow the formulation of service requests to be independent of their actual processing". This is to avoid the problem of the *direct* triggering of processes given the context of triggering. From the point of view of the environment or from different parts of an organisation, the actual business processes triggered for some business service request are inconsequential for the formulation of the request. An implication is that when processes are reengineered, the actual request is not affected. It can be seen that this is an application of the well-known *Information Hiding Principle* in software design.

A convenient way to model a service is an object. Objects, afterall, encapsulate processes, and their behaviour is described through a lifecycle of states and state transitions (see e.g. [RBP$^+$91, SM88]). Moreover the event-condition-action (ECA) paradigm which has been adopted for active rule specification in database systems, e.g. [CN90], and conceptual specification languages, e.g. [LMS$^+$91], can easily be adapted for event specifications. A service model for the road closures business transaction is depicted in Figure 9. The service model consists of: normal states (large polygons); special states indicating the "birth" of a service (small unshaded polygon) and the "death" of a service (small black-

shaded polygon); and state transitions (arcs).

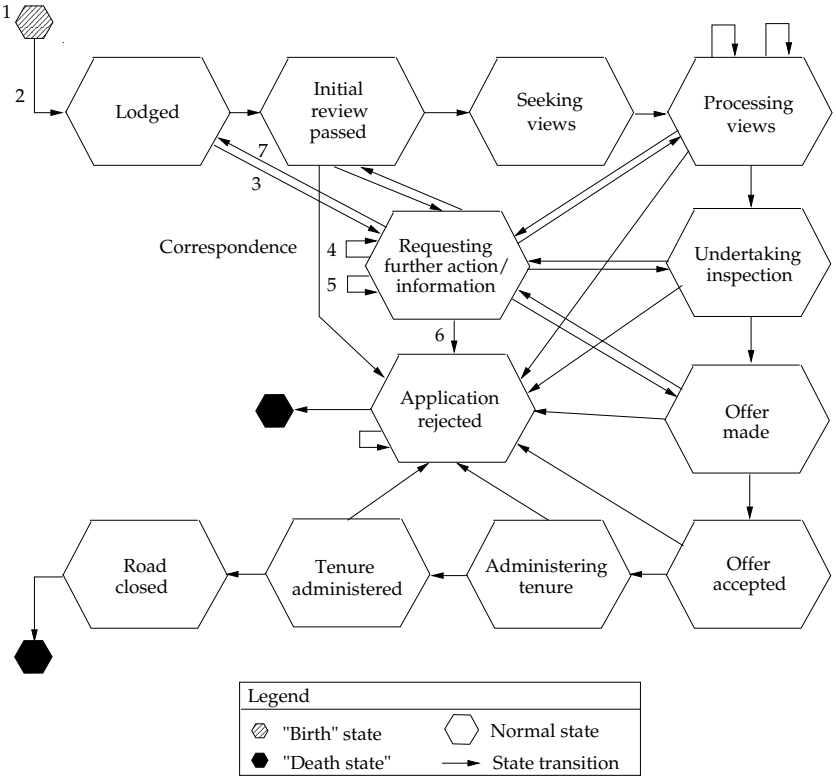

Figure 9: Service Model for Road Closures

The first event is the arrival of the message Application Documents. This is an example of a *messaging event*. It is distinguished from an actual external event e.g. the signing of a contract for an estate development. In general, the inclusion of such external events do not seem necessary for workflow specifications, although they could be captured through an event dependency formalism separate to the workflow specification. This first event leads to the service object (instance) creation in the "birth state" state. Upon this creation, it sends the message to Application Entry thereby triggering the workflow described in Figure 3:

WHEN Application Documents **RECEIVED**
THEN SEND Application Documents **TO PROCESS** Application Lodgement

The transition to the Lodged state occurs when the Application object is first entered into the Application Database (2) - an example of a *database state event*. The expression is formulated using LISA-D demonstrating how conceptual data specification languages can be used by ECA languages. The predicate, in this case, is an arbitrary fact type with a mandatory role since this will evaluate to "true" after the Application is stored in the database. No **THEN** part follows since the workflow execution still continues (without the need for invoking further processing). This further illustrates the importance of service states capturing the required perceptions of service stakeholders, independent of the underlying workflow execution:

WHEN Application received-on Date

A problem may found in the Application (this relates to the internal checks done as part of the decision Initial review passed? described in Figure 5). In this case, a Request Further Action/ Information may executed. Its issue of a Notice of Further Action/Information message - a messaging event but this time outgoing - is detected by the service object for the next state transition (3):

WHEN Notice for Further Action/Information **SENT**

The subsequent Correspondence, like all incoming messages from the environment, is sent to the service object, and so the service object further activates processing. The Correspondence should be examined, and so no state change results (4):

WHEN Correspondence **RECEIVED**
THEN SEND Correspondence **TO PROCESS** Examine a Correspondence

Although not included in the process model description, Correspondence may not be satisfactory in which case the Suspend Processing may be reinvoked with the Minister's Delegate's decision to Reject Application? - a *processing event* involving negative decision termination:

WHEN DECISION Reject Application? **REJECTED**

or a positive decision termination resulting in the transition to the Application rejected state (6):

WHEN DECISION Reject Application? **ACCEPTED**

The above ECA rules provide some indication of the types of events which a service can react to. In the full case study, the need for events types were identified, typified by: process objects commencing execution; process objects failing to commence execution over a different numbers of times, and the occurrence of terminating aborts.

3.5 Execution Resilience

The *Execution Resilience Principle* requires that "a technique should support the handling of operational errors, so that business processing execution may be verified as being resilient". Although traditional process modelling techniques do not deal with this aspect, a recently proposed workflow modelling technique, e.g. [CCPP95], provides basic mechanisms for exception handling. A more detailed treatment of non-deterministic failures has recently become the subject of workflow implementation specifications. In particular, the traditional transaction model (more recently described in [GR93]) with its ACID properties (atomicity, consistency, isolation and consistency) has been extended for workflow execution semantics; see survey of transactional workflows issues in [Kim94] (pp. 596-598).

Under the traditional model, a transaction binds a set of database operations into an atomic unit of execution. Following the requirement of *failure atomicity*, a transaction's changes to a database(s) are *committed* if the execution is successful or *rolled-back* if not. Workflows are more complex structures than traditional transactions, and it is unacceptable that the failure of any one of its processes results in the rollback of the entire workflow. Although not described here, the need for two different recovery strategies is apparent from the road closures example. One is *rollforward recovery* which seems appropriate when a failure occurs outside the control of the system (e.g. system crashes). For this a *redo* or a contingent process is run; a greater contingency flexibility can be introduced by specifying a range of contingent processes qualified by the number of startup failures of the process. In other words, rollforward recovery ensures workflow *durability*. The other is a *rollback recovery* which occurs when a failure is generated by the system (e.g. a terminating abort). For this an *undo* possibly through the execution of some other process, i.e. a *compensation* occurs. *Compensations* are necessary since a process can commit and release its resources prior to a workflow reaching a termination state, therefore allowing other processes to access its updates. In sum, rollback recovery ensures workflow *consistency*. In addition to the rollback of process objects within a decomposition, service states also provide an extra level of rollback granularity. Of course, a declarative, ECA type of language is conducive for the specification of recovery actions given the different failure events.

4 Epilogue

As bourne out in the CAiSE experience, the complexity of problem domains together with the productivity required for the development of their IS solutions and the quality of maintaining those solutions thereafter, makes automated support compelling. The focal nature of tools, however, can lead to the perception that tools "become" the methods and techniques that they support. Tools, and the improvement of tools as such, do not necessarily improve the understanding of problem domains. This is very much dependent on the quality of methods and techniques available in the tools. This invaluable lesson is undoubtedly timely for workflow technology where tools providing analysis, design, implementation and even run-time execution have recently become available prior to a well-formed insight into how well a workflow is actually positioned over business processing. Indeed much of the workflow ontology standardised by the WMC is clearly implementation-oriented and still leaves the critical issue of workflow conceptualisation open.

In one form or another, the requirements which lead to effective conceptual modelling are that: techniques should adhere strictly to the conceptual level; should provide a high degree of expressive power; should at the same time facilitate comprehensibility; and should be backed up by a solid formal foundation whereby both the syntax and semantics are clearly defined. Equally importantly, a technique should be suitable for its problem domain, meaning that its concepts and features reflect closely those of the problem domain.

Rather than propose yet another technique with yet another new set of concepts, this paper sought to provide an insight into what areas, amongst others, workflow modelling techniques need to improve on given a real-scale problem domain. Conceptual modelling is vastly developed and so we advocated a synthetic approach to the proposal of a workflow model for the domain. The concepts and paradigms of the model are only important in so far as they reflect the framework for workflow modelling *extension* that we proposed. This framework comprises a set of suitability principles which we previously identified through an assessment of a number of techniques. In doing so, we intended to convey the inherent

complexity of problem domains requiring workflow management. Moreover, attention was restricted to what we believe to be the most basic types of workflows, i.e. operational business transactions.

Through the principles of *Organisational Embedding* and *Scenario Validation*, the not often salient aspect of aligning workflow models with enterprise concepts and an enterprise cognition was described. For the *Cognitive Sufficiency Principle*, we anchored into an essentially behavioural process model - typifying workflow models in commercially available tools like IBM's FlowMark and DEC's LinkWorks - messages and message handling, object store access, HCI points and temporal aspects. Rather than loosely-couple these concepts, the benefit of scaling a tightly-coupled technique (Hydra) was most demonstrated through the degree of expressiveness possible with its specification language, LISA-D. We believe this insight into process semantics complements the current strenghts of process execution semantics in workflow specifications.

We recognised in the *Service Information Hiding Principle* that the interfaces to workflows, in a business sense denoted through business services, should be insulated from any knowledge of the workflows. Like business services in the real-world, their customers and users should not have to know how they are implemented in formulating service requests. A service concept was therefore motivated and for it the declarative specification benefits of object behaviour, i.e. a state-centric (ECA) paradigm seems most appropriate. Finally, through the *Execution Resilience Principle*, we discussed that recovery management aspects of workflows have to be addressed by techniques. We advocate a rollforward recovery strategy for non-failure aborts and a rollback recovery strategy for failure aborts; all of which integrates the well-established transaction concepts of *undo* and compensations, and *redo* and contingencies.

As a result of this work, we believe several future research issues are pertinent. Firstly, the classification of workflow scenario types requires further research including much empirical exposure. A restriction of attention to particular types, rather than the outright development of "silver bullets", we believe, will fertilise an effective development of techniques. Secondly, as with the development of any technique, the ramifications of amalgamating concepts and proposing new features should be understood formally. In particular, the formal semantics of workflow specifications should be given due attention as these specifications involve highly concurrent process behaviour (including synchronous and asynchronous communication), complex constraints and database operations. Informal specifications of such a complex nature are bound to be ambiguous. Furthermore, without a formal semantics verification and validation are not possible. In [HOR96] a number of verification issues in workflow specifications are identified and their complexity analyzed. An open issue is the determination of completeness of temporal constraints in a workflow model. We discussed some constraints on process model objects, but this treatment was illustrative and partial. Temporal requirements in service provision, as sometimes identified in service performance indicators, is but one scope of extension.

References

[AMP94] A.I. Anton, W.M. McCraken, and C. Potts. Goal decomposition and scenario analysis in business process reengineering. In G. Wijers, S. Brinkkemper, and I. Wasserman, editors, *Proceedings of the Sixth International Conference CAiSE'94 on Advanced Information Systems Engineering*, volume 811 of *Lecture Notes in Computer Science*, pages 94–104, Utrecht, The Netherlands, June 1994. Springer-Verlag.

[BB91] J.C.M. Baeten and J.A. Bergstra. Real Time Process Algebra. *Formal Aspects of Computing*, 3:142–188, 1991.

[BB95] A.T. Berztiss and J.A. Bubenko. A software process model for business reengineering. In *Proceedings of Information Systems Development for Decentralized Organizations (ISDO95), an IFIP 8.1 Working Conference*, pages 184–200, Trondheim, Norway, August 1995. Chapman & Hall.

[BH96] A.P. Barros and A.H.M. ter Hofstede. A Business Transaction Workflow Case Study: Road Closures in Queensland. Technical Report 393, Department of Computer Science, University of Queensland, Brisbane, Australia, December 1996.

[BHPC96] A.P. Barros, A.H.M. ter Hofstede, H.A. Proper, and P.N. Creasy. Business Suitability Principles for Workflow Modelling. Technical Report 380, Department of Computer Science, University of Queensland, Brisbane, Australia, August 1996.

[Bot89] P.W.G. Bots. *An Environment to Support Problem Solving*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 1989.

[Bro87] F.P. Brooks Jr. No silver bullet: essence and accidents of software engineering. *IEEE Computer*, 20(4):10–19, April 1987.

[BS87] D. Benyon and S. Skidmore. Towards a Tool Kit for the Systems Analyst. *The Computer Journal*, 30(1):2–7, 1987.

[CCPP95] F. Casati, S. Ceri, B. Pernici, and G. Pozzi. Conceptual Modeling of Workflows. In M.P. Papazoglou, editor, *Proceedings of the OOER'95, 14th International Object-Oriented and Entity-Relationship Modelling Conference*, volume 1021 of *Lecture Notes in Computer Science*, pages 341–354. Springer-Verlag, December 1995.

[CN90] S. Chakravarthy and S. Nesson. Making an object-oriented DBMS active: design implementation and evaluation of a prototype. In *Proceedings of the International Conference on Extended Database Technology (EDBT)*, Venice, Italy, April 1990.

[CP96] P.N. Creasy and H.A. Proper. A Generic Model for 3-Dimensional Conceptual Modelling. *Data & Knowledge Engineering*, 20(2):119–162, 1996.

[Dav90] A.M. Davis. *Software Requirements: Analysis & Specification*. Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[DB91] R.C.J. Dur and P.W.G. Bots. Dynamic Modelling of Organizations Using Task/Actor Simulation. In R.L. Crosslin and H.G. Sol, editors, *Proceedings of the Second International Working Conference on Dynamic Modelling of Information Systems*, pages 49–71, Amsterdam, The Netherlands, 1991. Elsevier Science Publishers.

[Die94] J. Dietz. Business modelling for business redesign. In *Proceedings of the 27th International Conference on System Sciences*, pages 723–732, Honolulu, Hawaii, August 1994.

[DP95] V. De Antonellis and B. Pernici. Reusing specifications through refinement levels. *Data & Knowledge Engineering*, 15(2):109–133, April 1995.

[FL80] F. Flores and J.J. Ludlow. Doing and Speaking in the Office. In *Decision Support Systems: Issues and Challenges*. Pergamon, 1980.

[FYW94] G. Fitzpatrick, Y. Yang, and J. Welsh. Supporting Cooperative Work Processes - A Survey of Systems and Issues. Technical Report 304, Department of Computer Science, University of Queensland, Brisbane, Australia, June 1994.

[GHS95] D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modelling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, April 1995.

[GR93] J. Gray and A. Reuter, editors. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, San Mateo, California, 1993.

[Hal95] T.A. Halpin. *Conceptual Schema and Relational Database Design*. Prentice-Hall, Sydney, Australia, 2nd edition, 1995.

[HN93] A.H.M. ter Hofstede and E.R. Nieuwland. Task structure semantics through process algebra. *Software Engineering Journal*, 8(1):14–20, January 1993.

[Hof93] A.H.M. ter Hofstede. *Information Modelling in Data Intensive Domains*. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, 1993.

[HOR96] A.H.M. ter Hofstede, M.E. Orlowska, and J. Rajapakse. Verification Problems in Conceptual Workflow Specifications. In B. Thalheim, editor, *Proceedings of the 15th International Conference on Conceptual Modeling (ER'96)*, volume 1157 of *Lecture Notes in Computer Science*, pages 73–88, Cottbus, Germany, October 1996. Springer-Verlag.

[HPW93] A.H.M. ter Hofstede, H.A. Proper, and Th.P. van der Weide. Formal definition of a conceptual language for the description and manipulation of information models. *Information Systems*, 18(7):489–523, October 1993.

[HSV89] K.M. van Hee, L.J. Somers, and M. Voorhoeve. Executable Specifications for Distributed Information Systems. In E.D. Falkenberg and P. Lindgreen, editors, *Information System Concepts: An In-depth Analysis*, pages 139–156. North-Holland/IFIP, Amsterdam, The Netherlands, 1989.

[Kim94] W. Kim, editor. *Modern Database Systems: The Object Model, Interoperability and Beyond*. Addison-Wesley, Reading, Massachusetts, 1994.

[LK95] P. Loucopoulos and E. Kavakli. Enterprise modelling and the teleological process design and database design. *International Journal of Cooperative Information Systems*, 4(1):45–79, January 1995.

[LMS+91] P. Loucopoulos, P. McBrien, F. Schumaker, B. Theodoulidis, V. Kopanas, and B. Wangler. Integrating database technology, rule based and temporal reasoning for effective information systems. *Journal of Information Systems*, 1:129–152, February 1991.

[LR94] F. Leymann and D. Roller. Business Process Management with FlowMark. In *Proceedings of IEEE COMPCON 94*, pages 230–234, San Francisco, California, March 1994. IEEE Computer Society Press.

[ML83] J.L. Malouin and M. Laundry. The Mirage of Universal Methods in Systems Design. *Journal of Applied Systems Analysis*, 10:47–62, 1983.

[MMNR90] V.J. Mercurio, B.F. Meyers, A.M. Nisbet, and G. Radin. AD/CYCLE strategy and architecture. *IBM Systems Journal*, 28:170–187, 1990.

[Ram94] G.J. Ramackers. *Integrated Object Modelling, an Executable Specification Framework for Business Analysis and Information System Design*. PhD thesis, University of Leiden, Leiden, The Netherlands, 1994.

[RBP+91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson. *Object-Oriented Modeling and Design*. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[Rod91] T. Rodden. A survey of CSCW systems. *Interacting with Computers*, 3(3):319–353, 1991.

[SM88] S. Shlaer and S.J. Mellor. *Object-Oriented Systems Analysis: Modeling the World in Data*. Yourdon Press, New York, New York, 1988.

[TL91] C.I. Theodoulidis and P. Loucopoulos. The Time Dimension in Conceptual Modelling. *Information Systems*, 16(3):273–300, 1991.

[WW93] D.G. Wastell and P. White. Using process technology to support cooperative work: Prospects and design issues. In D. Daiper and C. Sanger, editors, *CSCW in Practice: An Introduction and Case Studies*, pages 105–126. Springer-Verlag, London, United Kingdom, 1993.

[You89] E. Yourdon. *Modern Structured Analysis*. Printice-Hall, Englewood Cliffs, New Jersey, 1989.

[Zav86] P. Zave. Salient Features of an Executable Specification Language and Its Environment. *IEEE Transactions on Software Engineering*, 12(2):312–325, 1986.