

Business Process Modeling for mGovernment Applications

Peggy Schmidt

Institute for Computer Science and Applied Mathematics, Kiel University
Olshausenstr. 40, 24098 Kiel, Germany
e-mail: pesc@is.informatik.uni-kiel.de,
web page: <http://www.is.informatik.uni-kiel.de/~pesc>

Abstract: *mGovernment applications are usually part of greater enterprise environments. Technically, these enterprise applications are called bulk applications because they consist of a variety of complex interrelated business processes. To handle the complexity of these processes suitable modeling techniques are needed. Typically, UML is used for these purposes. In this paper an alternative approach based on the website description language SiteLang is presented. This language supports use case modeling as well as interaction and workflow specification in a unified way using a single dialog type. Due to SiteLang's simple constructs and well defined semantics the language is commonly understandable. SiteLang modeling is explained using a simple example from one of our eGovernment projects.*

Keywords: business process modeling, context, eGovernment, software specifications, storyboarding, SiteLang, web service

1. Introduction

The success of an information system is based on a deep understanding of its application area. Internally, however, things may be quite different. Complex bulk applications require a complex structure of system components. Therefore we need models, which handle this complexity.

Business process modeling is an essential part of understanding the complexities and communication issues inherent in any business. A business process is determined by value chains. However, business process modeling is often simplified, leads to incorrect, inefficient or completely unsuitable process definitions.

This paper presents an approach that reduces the complexity by decomposing complex bulk applications in business processes which are further divided into the components workflow, interactivity and content types. Many process modeling approaches consist of a single flow diagram styled view that provides a basic representation of logical order and the input and output of each process. This is in most cases inadequate. The challenge is to create an accurate and concise model of a business process. The dynamic behavior can be described as a number of interrelated business processes. For such business processes collaborating systems are necessary.

The objective of this paper is to show how SiteLang and UML notations and modeling techniques can support the development of business processes. Business process languages for web service choreography are becoming more and more important in the domain of enterprise integration. An enormous amount of work has been done in business process research.

Modeling techniques like UML allow the design of business processes on different levels. For the description of the behavior use case diagrams, activity diagrams, state diagrams or collaboration diagrams are available. Between these techniques only weak points of contact exist. Transitions between the abstraction levels usually require manual translations by the designer. If rules exist, they are often interpretable.

In contrast to this situation the website description language SiteLang has been created (Thalheim & Düsterhöft, 2001). It has a well-defined semantics and is able to model control flows. This paper shows how a business process model can be obtained from an interface description.

2. Background Material

2.1. Related Work

Formal business process languages like BPML contain semantic definitions that are enforced by the language standard. Other groups, such as the Workflow Management Coalition, publish formal glossaries. Still other groups define terms in ways specific to their particular community. Most of these communities use or define these terms in slightly different ways. BPML is a meta-language for modeling of business processes based on XML. It offers an abstract execution model for transactional and collaborative business processes. BPML defines a business process as an interaction between participants and the execution of activities according to a defined set of rules in order to achieve a common goal.

The theme “workflow management” derived a permanently increasing interest for research and application in the last years. (Kiepuszewski, 2002) analyzes control flows by using workflow patterns. (Kesten & Pnueli, 2002) presents an axiomatic system for quantified propositional temporal logic.

2.2. Business Process Diagram Notations

UML

Dynamic interaction and behavior in UML is divided into three main groups:

1. Sequence diagrams are used to describe interactions between object instances at runtime. A sequence diagram provides a sequential map of messages between objects on a timeline. These diagrams are often used to express case studies.
2. Common activity diagrams describe business processes and user interactions. Activity diagrams are applied to explain how different business processes in a system are composed, how they start, and which parallel processing may occur during execution. Activity diagrams often explain the process that is performed by a use case and how an actor will use the system.
3. State diagrams summarize business process changes all along. Any object that has non-constant instance variables at runtime has some latent state. The actual state of an object relies on the values of its instance variables.

The Storyboarding Language SiteLang

This paper is based on the publication of (Düsterhöft, Thalheim 2001). They presented a formalized language SiteLang to support the specification of story boards. Storyboarding is a notion, which permits a dramaturgic modeling of interaction and control flow, developed originally in linguistics and movie business. Storyboarding techniques can be used for specifying business process behavior. It is based on the concepts of story, scenario, scene, dialog step and content types.

The story of interaction between the user and the user interface or the user interface and the workflow is the 'plot' of the narrative work or an account of events. A scenario is a composition of stories, possibly with branches and loops. A story space is a set of scenarios (Thalheim, Düsterhöft, 2001) and can be modeled by means of a many-dimensional graph.

A scene is an abstract characterization of a certain set of activities. A scene is coupled with content types, with a set of implicated actors, the context of the scene, applicable representation styles, and a dialog step expression for the specification of the scene. A content type is a possibly complex data structure generated from the database. Content types correspond to generalized views on databases (Thalheim, Schewe, 2004).

3. Business Process Modelling

3.1. Conceptual Understanding of Business Processes

In order to reduce the complexity of bulks applications, we analyze business processes at first. A business process is a summary of technical knowledge which connects dependent activities and serves for reaching of an economical goal. Business processes stand in temporal and logical dependence to each other.

Business processes can trigger other business process activities or occur in different variants. There are often similar business processes, which possess a homogeneous operational sequence, but are based on different content types. Business processes can be interrupted in the sequence similar to transactions, can be long persisted, be based on other business processes and own different granularity (Thalheim, 2004).

To reduce the complexity of a business process it can be divided into user interface scenarios and workflows scenarios. These scenarios are associated with content types. Within a workflow a content type instance has a lifecycle. Its data changes as the workflow proceeds from one work unit to another.

User interface scenarios give a description of the application from the user's point of view. A content type can be figured out as an electronic document or information in a business process and is defined by a view on some underlying database schema, i.e. it consists of a view schema and defining queries. Such a query must be able to create identifiers in order to create links between the various content types. The view definition is enhanced by functionality, navigational micro behavior, and further associations. A workflow scenario consists of a sequence of scenes, called work units. A scenario can contain branches and loops. Work units are composed of work steps. Work steps contain an abstract semantic specification of a function, which is totally enclosed in a transaction. We add a precondition and a post condition to each work step to define under which conditions the step can be executed and which effect it has. A work unit should be accomplished without interruption. Each work unit has a set of triggering events, called messages.

3.2. Requirements of Business Processes Modeling

We use the term business process architecture to cover the process diagram for processes. A business process diagram is a diagram that shows the interactivity between the user and the user interface as well as the user interface and the control flow of the business process. It shows which sub processes or activities are managed.

For business process analysis and design we need levels. A high-level use case diagram shows main processes called work units. Each main process is in general divided into sub processes which are represented on separate work unit diagrams. Those work units can be refined to sub processes corresponding to sub work units. The smallest sub units we can identify are called work steps. A work step can represent an activity.

In asynchronous processes one process action sends at least one message to another action. It does not wait until it gets a response. The business process interaction model is indicated by its concurrent and reactive behavior. A parallel process is a process in which more than one process sequences are running simultaneously. If a content type instance is being passed from one actor to several others the process is necessarily a parallel process.

A business process diagram describes a generic sequence of process steps. An instance of a business process explains a concrete business process which includes a content type instance, concrete process steps and dedicated decisions.

Message handling enables two work units or content objects to communicate with each other. It allows transformation of the content coming from the sender interface to match the requirements of the receiver interface.

3.3. An Example from eGovernment Modeled with SiteLang

Consider the following example: In a municipality administrative processes have to be automated and integrated. All relevant business processes shall be supported by a homogeneous software system. In order to support employees of the municipality different systems are used or business processes are not supported sufficiently.

Consider the process of an automatic execution of a business trip application. We have the work units “business trip application request” and “approval of business trip”. There are two actors: employee and superior.

An application of a business trip can be triggered by a WML or a HTML interface. The WML interface has more dialog steps than the HTML interface. When the actor presses the submit button a new “businessTripID” is generated and the business application data is saved. In the next work unit the superior of the employee has to approve the business trip application. The superior can choose whether he wants to select a business application directly or browse business applications by employees.

The use case story describes the requirements of the business process outward, i.e. from the user’s point of view and gives an overview over the characteristic operational sequence and the actors involved in the business process. A distinction between direct and indirect actors is necessary. Direct participants communicate directly (by using a login) with the system (Oestreich 1999). Indirect actors interact with the system via a direct actor. For example, during the business process of the business trip application the superior has to approve the business trip. Superiors have secretaries. These secretaries may enter decisions of the superior into the system. In this case the secretary is a direct actor and the superior an indirect actor.

Figure 1 uses scenes to show which user interaction, workflow interaction and content objects exist. The actor of the process always appears on the start of a scene. The vertical axis usually shows the flow of time. A loop in a business process diagram is permitted, but violates a strict time flow. Rectangles with rounded corners represent dialog steps that are work units with sub processes. Arrows represent diverse types of flow between dialog steps and scenes. We divide the business process diagrams into user interface scenario that show a user interaction actually performed and workflow scenarios that show how the flow of a set of work units.

The starting point of the modeling of business processes is based on an abstraction layer model (Schewe & Thalheim, 2004). The paper considers the following layers: the business layer, conceptual layer and implementation layer.

The essential activities on the business layer deal with user interactivity and storyboarding, which addresses the design of an underlying application story. On the conceptual layer the activities address the support of scenes by modeling content types. Additionally, we synchronize the workflow with the interactivity in one storyboard. This idea is similar to the model driven architecture (MDA). The core idea of the MDA is the step by step refinement from the analysis to the design based on professional business logic, which is completely independent of the technical implementation and the enclosed IT infrastructure.

The storyboard is illustrated in Figure 1. Each of the diagrams on the left side represents a user interface scenario. Some functions contained in the user interface are repeated unnecessarily. For example, the function which saves the approval of the business trip and reserves the necessary budgetary funds was inserted at different dialog steps. If it has to be changed rapidly the same change has to be done in numerous dialog steps. If the change is done only at one dialog step the behavior of the process becomes inconsistent. Based on scenario descriptions we extract coherent chunks of functions and preconditions. These functions are removed from the user interface scenario and transferred in a workflow unit. The boxes represent dialog steps in the interface scenario and work items in the workflow scenario. The continuous arrows represent allowed control flows between scenes. For example, at the approval scene at the employee search screen you can use the next button to go to the “selecting business trip” screen. Instances of this storyboard are illustrated in Figure 2. We recognize that the user which enters the business data using a notebook has other navigation facilities as the user which enters the data by mobile phone. However, both users trigger the same work unit type. The user interface of an actor with the superior role can trigger work items of a set of business process instances.

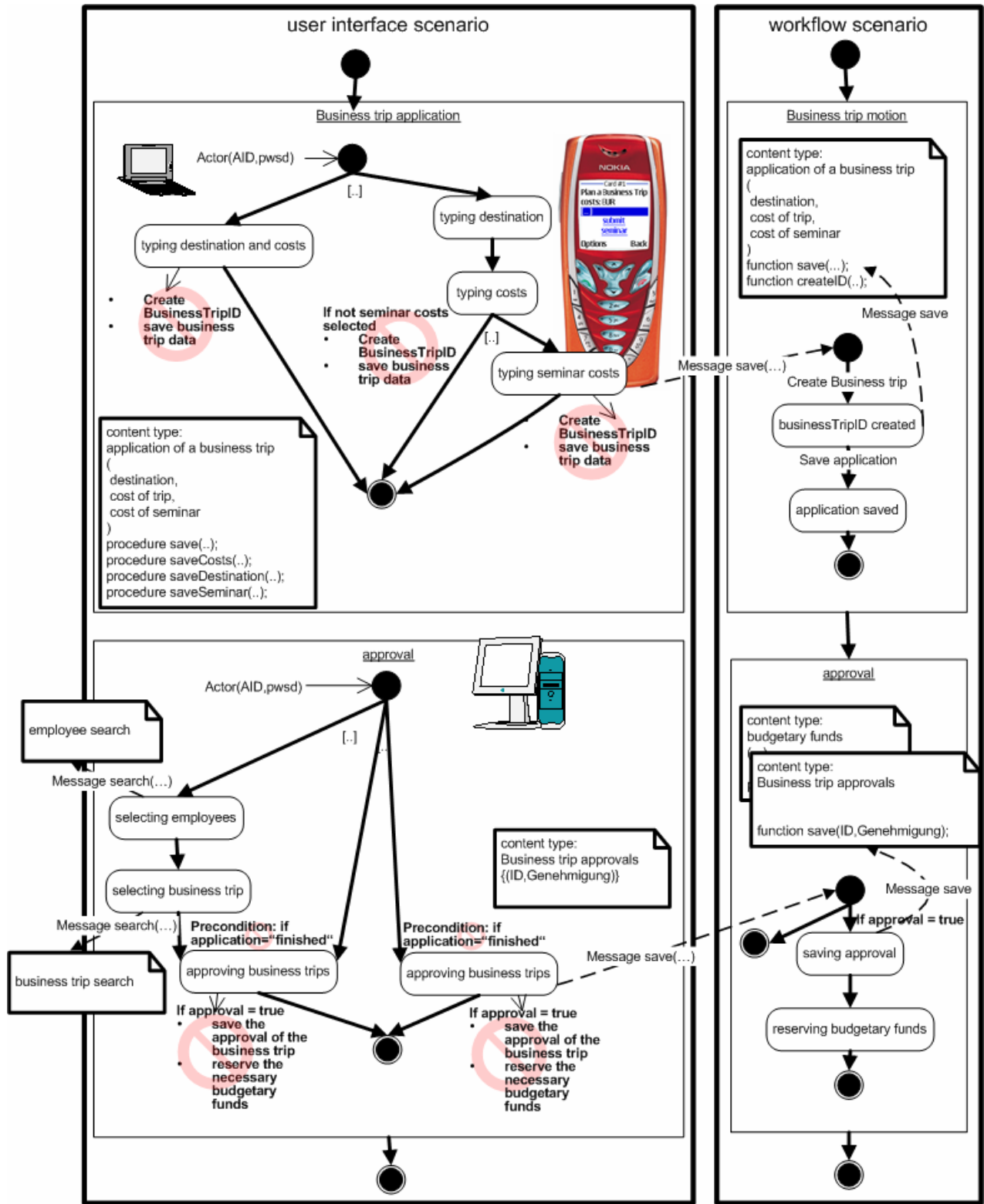


Figure 1 SiteLang specification for the business trip

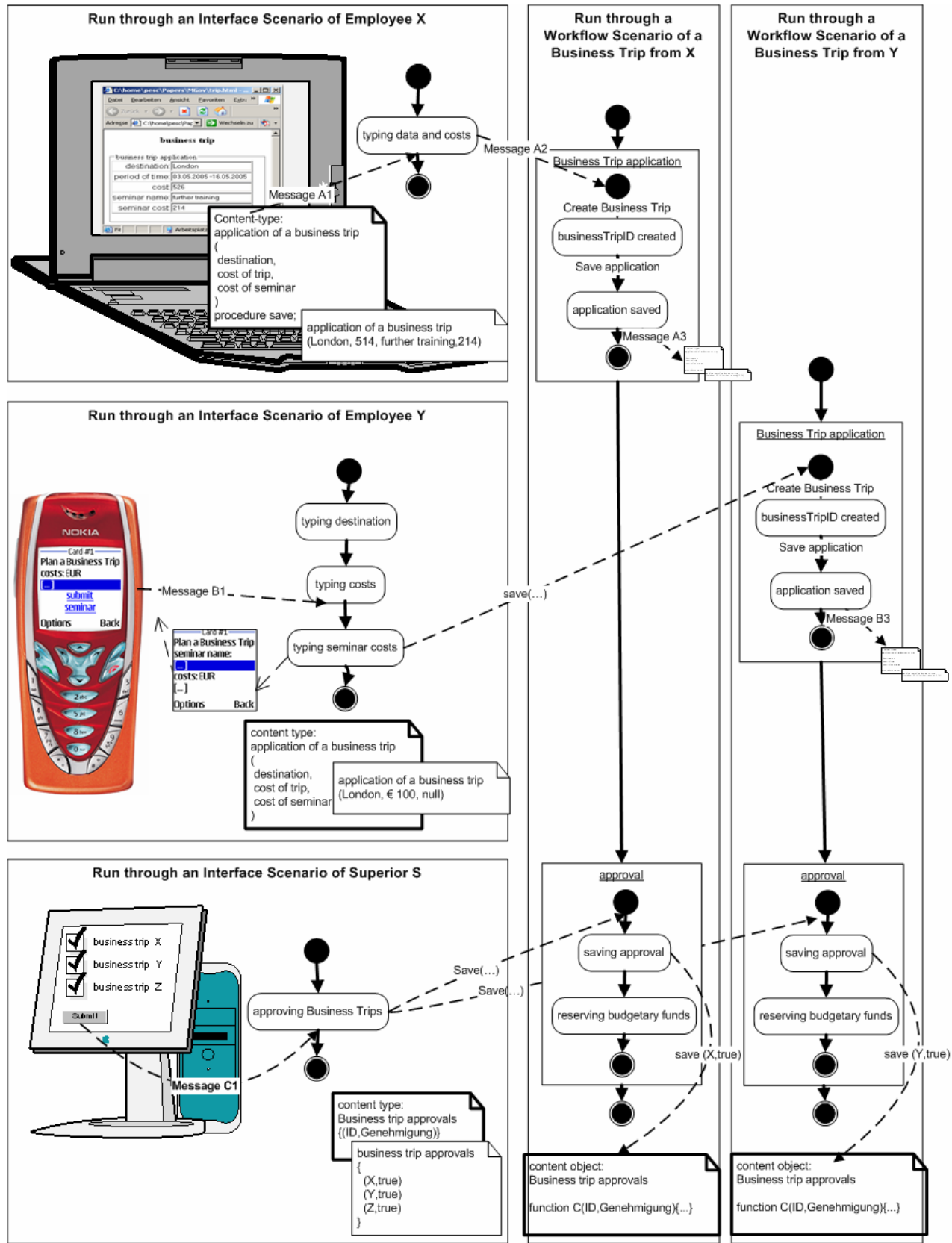


Figure 2 Instance of a SiteLang specification

3.5. New Concepts in the Language SiteLang

Figure 3 shows a Higher Order Entity Relationship Model (HERM) for the concepts of SiteLang. A scenario describes the navigation of an actor through a process. The navigation structure is described by partially ordered dialog steps. A scene subsumes dialog steps belonging to a single plot. Scenes that differ only in a small set of parameters, e.g. the set of enabled actors, can be expressed by scene templates. Composing a scenario for a particular actor includes the instantiation of the associated scene templates by applying concrete values to the template's parameters. Scenario instances are concrete runs through a scenario specification by a concrete user. A dialog step refers to a sequence of functions. Dialog steps and functions will execute if their guard and their accept-on conditions are fulfilled. Functions can be applied in different dialog steps, scenarios, and contexts.

This slightly different view on the SiteLang constructs simplifies the mentioned separation of storyboards and the generation of executable code directly from the SiteLang model but does not change the semantics of the SiteLang constructs.

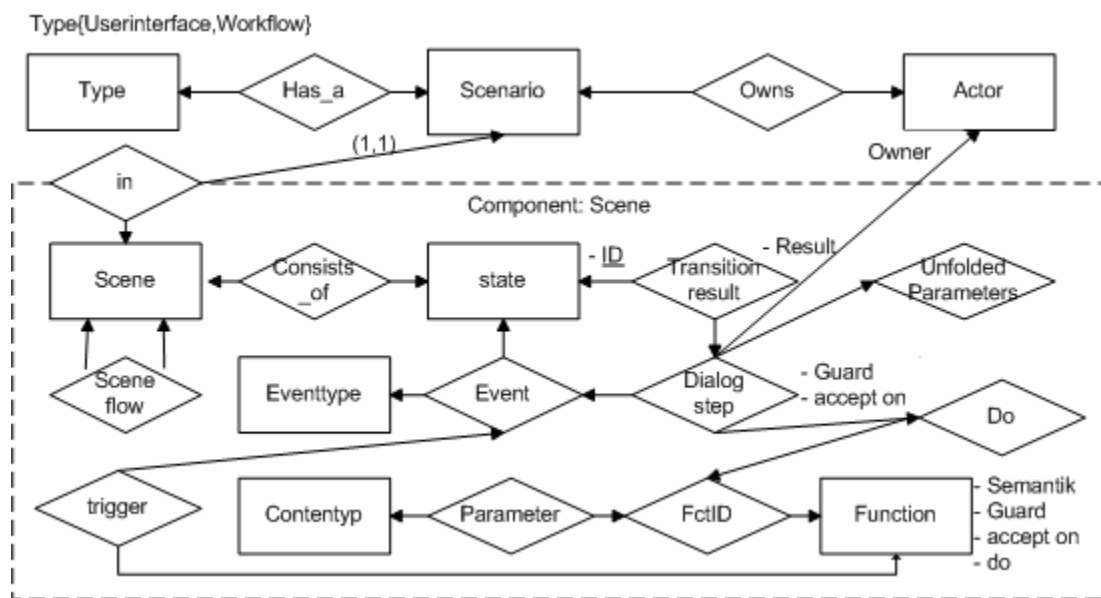


Figure 3: HERM for the language concepts of SiteLang

Figure 4 depicts a schema for the management of scenario runs. This is necessary if the current state of a scenario run should be persistent or if the exclusive execution of a dialog step has to be enforced. With this information we can compute the next dialog step that has to be enabled. Based on the knowledge about enabled dialog steps we can generate portfolio scenes for each user. This portfolio consists of all dialog steps which can be triggered by a user.

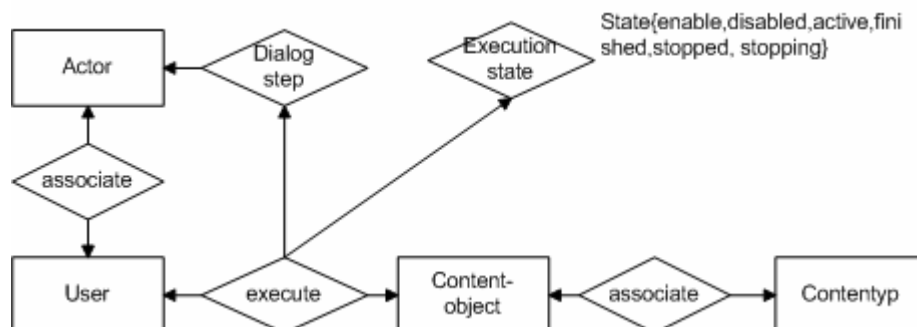


Figure 4: HERM for SiteLang instance management

3.4. Modeling the example with UML

SiteLang integrates use case, interaction and state-transition modeling. In principle UML also provides suitable constructs for modeling business processes.

Unfortunately, these constructs are not integrated into a single diagram type. In the following the paper shows how the scenario from the last section can be described using UML. The result is partially presented in figures 5, 6, 7, and 8.

A use case is a sequence of work units that offer an appreciable value to an actor. In the example there are the use cases “business trip application” and “business trip approval”.

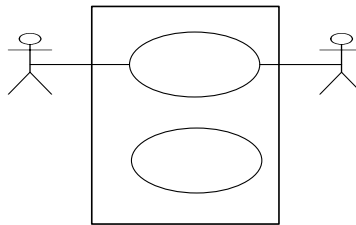


Figure 5 Use case diagram for the business trip

An activity diagram (Object Management Group 2003) can be used for modeling scenes.

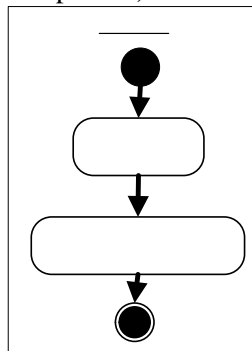


Figure 4 Activity diagram

A sequence diagram is a description of the messages passed between object instances. Sequence diagrams can be used to model usage scenarios between actors, user interfaces, workflows and content objects. Figure 5 shows an UML sequence diagram for the use case “business trip approval”. A business process sequence diagram helps to visualize and to validate the logic of a use case scenario. Figure 6 uses stereotyped icons to display particular objects. The boxes across the top of the diagram show the superior actor, the user interface, the workflow, and the content object as well as the messages passed between these objects.

bus
ap

* *

employee

bus
a

ap

save

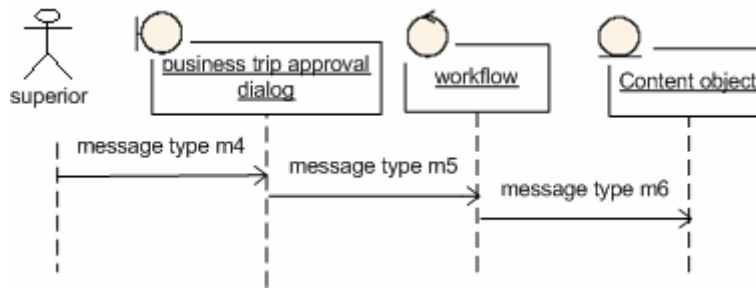


Figure 5 Sequence diagram

Figure 6 shows a state diagram for the content type “business trip”. The arrows represent transitions from one state to another. For example, when a business trip is in the “initiating” state, it can either be approved or cancelled. A problem is that the controller state is mixed with the content type instance state. This diagram is redundant because the information is already given within the activity diagrams.

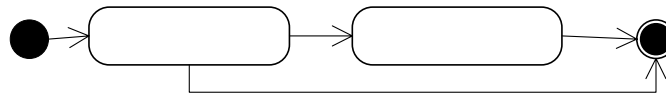


Figure 6 State diagrams

4. Conclusions

On the basis of the decomposition of complex applications into business processes the paper showed how a business process can be represented in different levels of abstraction. The explained modeling techniques demonstrate that SiteLang is a suitable language for a structured description of complex applications. Using a single diagram type interaction as well as workflows can be expressed. The paper shows that SiteLang supports all features necessary for business process modeling available in UML.

References

- Kiepuszewski. B., 2002, Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows , M.Sc. A dissertation presented to the Faculty of Information Technology Queensland University of Technology in fulfilment.
- Kesten Y. & Pnueli A., 2002, Complete proof system for QPTL. Journal of Logic and Computation, 12(5): pp. 701-745.
- Oestereich, B., 1998, Objektorientierte Softwareentwicklung, R.Oldenbourg Verlag München Wien
- Object Management Group, 2003, <http://www.omg.org> (2005/02/14)
- Schewe, K.-D. & Thalheim, B., 2004, Web Information Systems: Usage, Content, and Functionality Modelling, Report 0405, Kiel University, Institute for Computer Science and Applied Mathematics.
- Thalheim, B., 2003, Co-Design of Structuring, Functionality, Distribution, and Interactivity for Large Information Systems, Report 15/03, Brandenburg University of Technology at Cottbus.
- Thalheim, B. et.al., 2004, Website Modeling, Website Orchestration, and Website Management, Proceedings of LIT 2004.
- Thalheim, B. & Düsterhöft, A., 2001, Conceptual Modeling of Internet Sites, Proceedings of ER 2001.

<http://www.bpmi.org> (2005/02/14)

<ftp://ftp.omg.org/pub/docs/bom/00-12-11.pdf>. (2005/02/22)