# Context-Aware Web Information Systems

**Aleksander Binemann-Zdanowicz[2], Roland Kaschek[1], Klaus-Dieter Schewe[1], Bernhard Thalheim[3]**

[1] Massey University, Department of Information Systems, Private Bag 11 222, Palmerston North, New Zealand, Email: [r.h.kaschek|k.d.schewe]@massey.ac.nz
[2] Computer Science Institute, Brandenburg University of Technology at Cottbus, PostBox 101344, D-03013 Cottbus, Germany, Email: binemann@informatik.tu-cottbus.de
[3] Computer Science Institute, Christian-Albrecht-University Kiel, Germany, Email thalheim@is.informatik.uni-kiel.de

**Abstract**

Apart from completeness usability, performance and maintainability are the key quality aspects for Web information systems. Considering usability as key implies taking usage processes into account right from the beginning of systems development. Context-awareness appears as a promising idea for increasing usability of Web Information Systems. In the present paper we propose an approach to context-awareness of Web Information Systems that systematically distinguishes among the various important kinds of context. We show how parts of this context can be operationalized for increasing customers' usage comfort. Our approach permits designing Web information systems such that they meet high quality expectations concerning usability, performance and maintainability. We demonstrate the validity of our approach by discussing the part of a banking Web Information System dedicated to online home-loan application.

**Keywords:** Web Information Systems, Web services, context-aware information systems, media objects, SiteLang

## 1 Introduction

### 1.1 Generations of Web Services

Understanding Web Information Systems (WIS) as monolithic and presentation-oriented query-answer systems would be too simplistic. Implementing the individual services of a WIS only on the basis of XML or (D)HTML suites suffices for the interface accessible by a particular customer. The quality of service provided by a WIS both expected and implemented, however, evolved over the last decade and has evolved beyond mere completeness. Extending the classification in (Berger 2003, p.146) we distinguish between different generations of WIS.

First generation (1G): *"build it, and they will come"*
: First develop a WIS, then customers will come, because they believe that it is useful. Many of the 1G-WIS were informational, i.e., they weren't interactive.

Second generation (2G): *"advertise online sales, and they will come"*
: Develop a WIS and market it. Customers will come, because the advertisement convinced them about the WIS's usability. The WIS may be transactional, i.e., contain interactive interfaces to company products and services. A standard interface is provided but hard to learn. No particular customer usage aid is offered.

Third generation (3G): *"realize a pleasant use of high quality services, and they will come"*
: Customers will find using the WIS helpful. They will do the marketing. 3G-WIS's typical characteristics are:

- high value and up-to-date content,
- high performance,
- brand value of the provider, and
- pleasant and easy use for casual as well as for frequent customers.

Many WIS including several banking WIS are still 2G. However, impressive and well-developed WIS, e.g., the Amazon web-site, demonstrate the feasibility of 3G-WIS. The success of such WIS is based on deep understanding of the application area, the customers needs, abilities and habits. Adaptation to customers — if provided — is based on allocating the most suited subspace of the WIS application space to the customer.

WIS can be classified into e-business, e-learning, edutainment, community, information and personality WIS. In the e-business class the B2B systems have been more successful than B2C systems. This success results from well-understood usage scenarios built into the WIS. We observe that usage scenarios are better understood for B2B-WIS than for B2C-WIS.

Storyboarding is a design approach focusing on usage scenarios. However, so far it is mainly used employing pinboard approaches, see e.g. (Siegel 1998, Van Duyne et al. 2003). Pinboard approaches map a number of scenarios observed in the application onto tree-structured web sites. Storyboarding in the movie business is used to design much more complex scenarios. To overcome this limitation the storyboard specification language SiteLang has been introduced in (Thalheim and Düsterhöft 2001). Until now it has been applied in more that two dozen WIS projects of the Cottbus InfoTeam since 1999.

Our development experience implies that implementing 3G-WIS requires sophisticated database support, see (Thalheim 2000a). Our approach to guarantee for this support is based on the theory of media types, which generalize database views (see e.g. (Schewe and Thalheim 2001)). Another finding from our practical experiences is that customer behavior has changed. They are no more patiently waiting until their needs are met. They require personal interfaces. Customization of system interfaces to users is known for quite a while. However, WIS are targeting new and casual customers. These customers are not capable or willing to arrange for system adaptation. Internet service providers report customers frequently complaining about insufficient user-friendliness and unsophisticated WIS.

### 1.2 Problems of Complex Applications

Modern applications, in particular WIS often appear to be relatively simple, if only their interface is considered. Their point-and-click operating mode is deliberately set up in a way that causes the impression of simplicity. Internally, however, things may be quite different. A client-server multi-tier architecture with HTML-server, database server and application server might be used. This implies some non-trivial development tasks done such as database design and development of an application programmer interface or similar.

In addition, several customer types may be known to the application system. A WIS may appear very different to customers of different type. The functionality they access, however, is still the basic functionality as implemented by the servers mentioned before. Consequently as many function schemas and data schemas need to be developed as there are anticipated customer types. These schemas need to be integrated to develop a consistent view of the key application functionalities. Views that are based on these schemas need to be generated allowing the individual customers to operate with the application in the way that is most natural for them. The development of WIS thus can be a quite complex process.

Among others the complexity of this development process depends on the degree of use of an underlying database, from which dynamic web pages are created. Furthermore, the complexity of this process depends on the number of versions of usage processes of the WIS that need to be anticipated. Since different usage processes may lead to different data and functionality accessible to customers. Additional complexity comes in — e.g. in the case of modern retail banking — when a requirement is set in place that various access channels — e.g. channels needed for cellphone- or PDA-access — should be made available to customers. Apart from the purely technical problem arising from discretionary access channels the problem of layout for these channels has to be solved.

### 1.3 An Application Example

An example of a WIS in retail banking showing a relatively high diversity of the usage process is online loan application, if considered in full generality as we do it here. For an introduction to lending in general, of which the loan business is just a part, see e.g. (Valentine 1999). Not all banks offer online home-loan application facilities. Those that provide such facilities do not necessarily allow customers to deal with them completely online. Banks that offer effective online loan application are the Swiss UBS AG and the New Zealand and Australia based ASB Bank. The acceptance of a longer interruption of service at the ASB site indicated that at least for this bank online home-loan application is not yet considered a major part of their business.

Complexity in home-loan applications results from the fact that the applicant not necessarily is exactly one natural person. For each of the applicants properties and debts need to be identified and valuated. Often banks would accept only home-loan applications of at most two people, in general the couple that is going to live in the home financed with the loan. Complexity is further increased by the loan not necessarily being a fresh one but being already granted to someone who due to his or her financial conditions has chosen to move the loan to a different bank.

Furthermore, the properties offered for securing the loan may belong to a variety of types. Some of these types, e.g. real estate property may require physical inspection to determine the value they can cover. Other properties such as financial instruments, i.e. shares, options or accounts, may only need an inquiry to the respective depot or account. If cash is a security, then it might even be impossible to finish the process electronically, as the cash needs to be brought to the bank branch, counted and deposited.

It is similar with debts. On real estate properties there might be liabilities that require an assessment of the actual value. Of course there is quite a number of so-called loan structures (see (Valentine 1999, p.226f.)) distinguishing between loans. For instance, they may differ from each other in their term, frequency of repayments, borrower's authorization to increase the debt (e.g. overdraft facility), the minimum security ratio or the repayment structure. The latter addresses the schema of how capital and interests are paid for by the customer. Independently of the loan structure a customer might chose among several loan options that specify how the interests develop over time, i.e. they may be fixed for a particular time period or they may float like general interest rates in banking.

Apart from these principal choices in a home-loan there are a number of tools available for customer's use throughout online home loan applications such as a borrowing power calculator, a repayment schedule calculator, etc. Additionally, dictionaries of banking terms, act excerpts and comments as well as descriptions of the applying financial instruments need to be accessible to customers.

All the possible options in the financial instruments and the respective variations of the WIS usage will only be considered by a small number of customers. In more technical terms we have to deal with a generic process type. Most of its instances realize only a part of the possible variations. At present online home-loan application systems are not typical retail banking applications. Automated clearing house (ACH), i.e. direct deposit of payments, withdrawing monthly mortgage payments, etc. are more typical. According to (Berger 2003, p.150f.) its use in the US is steeply increasing and has after starting problems even increased productivity. However, ACH is a back-office activity, whereas online home-loan application is a customer-home or front-office activity. According to (Berger 2003, p.149) internet-only banks performed more poorly than conventional banks did. If this finding implies that online home-loan applications are less productive than conventional home-loan application processing then we believe that this is only a temporary phenomenon. We believe that cultural obstacles concerning internet-banking will disappear when 3G-WIS have become more popular.

According to (Berger 2003) there is empirical evidence for an increasing market share of electronic payment. According to studies reported in (Berger 2003, p.162) there is even empirical evidence for increased productivity due to investment in IT labor, while there is no empirical evidence for IT investments increasing efficiency in general. This is consistent with the basic insight that not the mere use of IT but the kind and quality of this use can increase productivity. Our paper shall help making 3G-WIS more popular and thus contributes to internet banking more completely covering the business at a higher level of quality.

### 1.4 Related Work

A lot of related work has been done on the development of web information systems. The work in (Atzeni et al. 1998) emphasizes the design of *content* leading to databases, *navigation* leading to hypertext, and *presentation* leading to the pages layout. Other authors (see for example (Baresi et al. 2000), (Bonifati et al. 2000), (Gädtke and Turowski 1999) and (Rossi et al. 1999)) follow the same lines of thought or concentrate on the "add-on" to database design, emphasizing mainly the hypertext design dealing with navigation structures (see (Garzotto et al. 1993) and (Schwabe and Rossi 1998)). The work in (Feyer et al. 1998) presents the forerunner of the theory of *media types* (see (Schewe and Thalheim 2001)). Media types provide a theoretically sound way to integrate databases, external views, navigation structures, operations, and even support adaptivity to different users, environments and channels. The *adaptivity* feature distinguishes them from the *dialogue types* that are used to integrate database systems with their user interfaces (see (Schewe and Schewe 2000)).

The work in (Schewe and Thalheim 2001) already emphasizes that conceptual abstraction from content, functionality, and presentation of an intended site is not sufficient for the adequate conceptual modelling of web-based systems, even if complex media types are taken into consideration. Some of the approaches mentioned before (see (Atzeni et al. 1998), (Baresi et al. 2000), (Bonifati et al. 2000), (Gädtke and Turowski 1999), (Rossi et al. 1999), (Garzotto et al. 1993) and (Schwabe and Rossi 1998)) miss out on the important aspect of story boarding, which is needed to capture the business content of the system.

Story boarding in a process-oriented holistic manner focusses on user intentions. In more recent work some of the authors (Kaschek et al. 2003a) started to investigate this idea more thoroughly. Conceptual modelling traditionally considered more ontological aspects than epistemological ones. Since web information systems in two respects considerably differ from non-web information systems epistemological aspects, however, need to be taken more seriously: Web information systems are open in the sense that actual users virtually may be just anyone. In non-web system there was traditionally a much stricter access control preventing non-staff from using the system. The business idea, however, has changed and customers need to be attracted and pre-selected by a web information system. Furthermore, web information systems are open in the sense that it is very easy to use them for accessing other web systems. This introduces more competition among those who offer services on the web. Quality of web information systems in the sense of fitness for users' use thus tends to be more important than it was for non-web systems. Web information systems partly substitute staff-customer interaction by customer-computer interaction. Consequently, web information systems must focus on aiding customers in doing the business the system provider is engaged in. Clearly this only can be done on the basis of a customer model. User profiling together with story boarding is a holistic manner for this.

In (Schewe and Thalheim 2001) it is suggested that story boarding be supported through directed graphs called *scenarios*, in which the nodes represent the scenes and the edges correspond either to navigation or to actions issued by the user. This extends the work in (Feyer et al. 1998), where simply partially ordered sets have been used. In addition, user profiling is approached by using *user dimensions* capturing various aspects of how to characterise users. This has been extended in (Srinivasa 2001) to a formal description of interactive systems.

The work in (Düsterhöft and Thalheim 2001) presents a formalised language `SiteLang` to support the specification of story boards. the work also indicates ideas how to exploit word fields for designing dialogue steps in story boards. In (Schewe et al. 1995) and (Schewe 1996) refinement primitives for dialogues have been discussed. Due to the connection between dialogues and scenarios, this approach to refinement is also useful for story boarding. The work in (Schewe et al. 2002) applies story boarding and user profiling to the area of on-line loan systems.

### 1.5 Outline

In section 2 we discuss WIS specification, in particular story spaces and scenarios, we further discuss media objects, dialogue-step specification and context. In the following section 3 we discuss database design for WIS, utilization of context for WIS and a stepwise WIS generation approach called "onion generation". Finally, in section 4 we continue the discussion of our example and show how our approach can be applied to modelling of WIS. Due to space restrictions, however, we can only discuss the storyboarding part.

## 2 WIS Specification

### 2.1 Story Spaces and Scenario

Modelling usage processes right from the beginning of systems development requires using a sufficiently expressive high level semantic model as a respective conceptual framework. Storyboarding uses the metaphor "story" to conceptualize usage processes. We presuppose that a story (for the source of the interrogatives used here refer to (Zachman 1987, Sowa and Zachman 1992)) tells what happened, why and where, as well as who did it how and when. The *story* of customer-WIS interaction thus is the intrigue or plot of a narrative work or an account of events.

Within a story one can distinguish threads of activity, so-called *scenarios*, i.e., paths of scenes that are connected by transitions. See figure 2.1 for an example scenario. We do not intend to model branching stories. These require managing a number of activities at the same time, i.e., in parallel. A capability that -as we believe- many casual customers won't have. With the term *story space* we mean the integration of all scenarios in a story.

We define the story space $\Sigma_W$ of a WIS $W$ as the 7-tuple $(S_W, T_W, E_W, G_W, A_W, \lambda_W, \kappa_W)$ where $S_W, T_W, E_W, G_W$ and $A_W$ are the set of scenes created by $W$, the set of scene transitions and events that can occur, the set of guards and the set of actions that are relevant for $W$, respectively. Thus, $T_W$ is a subset of $S_W \times S_W$. Furthermore $\lambda_W : S_W \rightarrow \mathit{SceneSpec}$ is a function associating a scene specification with each scene in $S_W$, and $\kappa_W : T_W \rightarrow E_W \times G_W \times A_W$, $t \mapsto (e, g, a)$ is a function associating with each scene transition $t$ occurring in $W$ the event $e$ that triggers transition $t$, the guard $g$, i.e. a logical condition blocking the transition if it evaluates to false on occurrence of $e$, and the action $a$ that is performed while the transition takes place. The language SiteLang, see (Thalheim and Düsterhöft 2001), offers concepts and notation for specification of story spaces, scene and scenarios in them. Scenes and their specifications are discussed in subsection 2.2.

### 2.2 Scenes

We consider scenes as the conceptual locations at which the customer-WIS interaction, i.e., dialogue

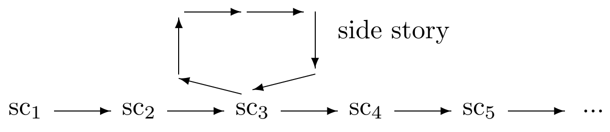


Figure 2.1: Scenario with a loop representing a side story

takes place. Dialogues can be specified using so-called dialogue-step expressions. Scenes can be distinguished from each other by means of their identifier: Scene-ID. With each scene there is associated a media object and the set of actors that are involved in it. Furthermore, with each scene a representation specification is associated as well as a context. Scenes therefore can be specified using the following frame:

Scene = ( Scene-ID
  DialogueStepExpression
  MediaObject
  Actors
    ActorID
    Right
    Tasks
      Assigned
      Roles
  Representation (styles, defaults, emphasis, ...)
  Context (equipment, channel, particular)

Dialogue-step expressions consist of dialogues and operators applied to them. Dialogue steps are discussed in subsection 2.4 below. The provided operators are based on the basic dialogue step algebra introduced in (Thalheim and Düsterhöft 2001):

- Basic control commands are sequence ; (execution of steps in sequence), parallel split $|\wedge|$ (execute steps in parallel), exclusive choice $|\oplus|$ (choose one execution path from many alternatives), synchronization $|^{sync}|$ (synchronize two parallel threads of execution by an synchronization condition $^{sync}$, and simple merge + (merge two alternative execution paths). The exclusive choice is considered to be the default parallel operation and is denoted by ||.
- Structural control commands are arbitrary cycles $^*$ (execute steps w/out any structural restriction on loops), arbitrary cycles $^+$ (execute steps w/out any structural restriction on loops but at least once), optional execution [ ] (execute the step zero times or once), implicit termination ↓ (terminate if there is nothing to be done), entry step in the scene ↗ and termination step in the scene ↘.
- Advanced branching and synchronization control commands are multiple choice $|^{(m,n)}|$ (choose between $m$ and $n$ execution paths from several alternatives), multiple merge (merge many execution paths without synchronizing), discriminator (merge many execution paths without synchronizing, execute the subsequent steps only once) n-out-of-m join (merge many execution paths, perform partial synchronization and execute subsequent step only once), and synchronizing join (merge many execution paths, synchronize if many paths are taken, simple merge if only one execution path is taken).
- We also may define control commands on multiple objects (CMO) such as CMO with a priori known design time knowledge (generate many instances of one step when a number of instances is known at the design time), CMO with a priori known runtime knowledge (generate many instances of one step when a number of instances can be determined at some point during the runtime (as in FOR loops)), CMO with no a priori runtime knowledge (generate many instances of one step when a number of instances cannot be determined (as in a while loop)), and CMO requiring synchronization (synchronization edges) (generate many instances of one activity and synchronize afterwards).
- State-based control commands are deferred choice (execute one of the two alternative threads, the choice which tread is to be executed should be implicit), interleaved parallel executing (execute two activities in random order, but not in parallel), and milestone (enable an activity until a milestone has been reached).
- Finally, cancellation control commands are used, e.g. cancel step (cancel (disable) an enabled step) and cancel case (cancel (disable) the step).

These control composition operators are generalizations of workflow patterns ( see, e.g. (Workflow Management Coalition 1999, Jablonski 1996)) and follow approaches developed for Petri net algebras.

A graphical representation of a login scene is given in figure 2.2. We are interested in well-formed dialogues and do not allow specifications which lead to and-split or or-split common in workflow specifications. This scene is specified by the dialogue step expression

*Enter_login* ;
  ( *Customer_login* ; [ *Change_profile* ; ]
    ( *Service_kind_selection* ; *Service_selection* ;
      *Service_customization)*
    || *Join_cooperating_group*
    || *Join_bank_club*
    || *Join_bank_programs*
    || *General_customer_information )*
$|\oplus|$ ( *Anonymous_Login* ; [*Extend_adding_identity* ; ]
  ( *Program_selection* ; *Module_selection* ;
    *Unit_selection) )*

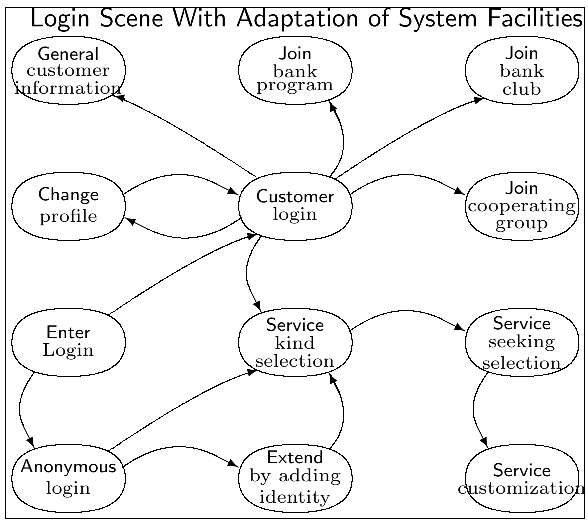


Figure 2.2: Scene for Login Into a Bank WIS

### 2.3 Media Objects

A scene is supported by *media objects* following the codesign approach. Media objects are instances of media types.

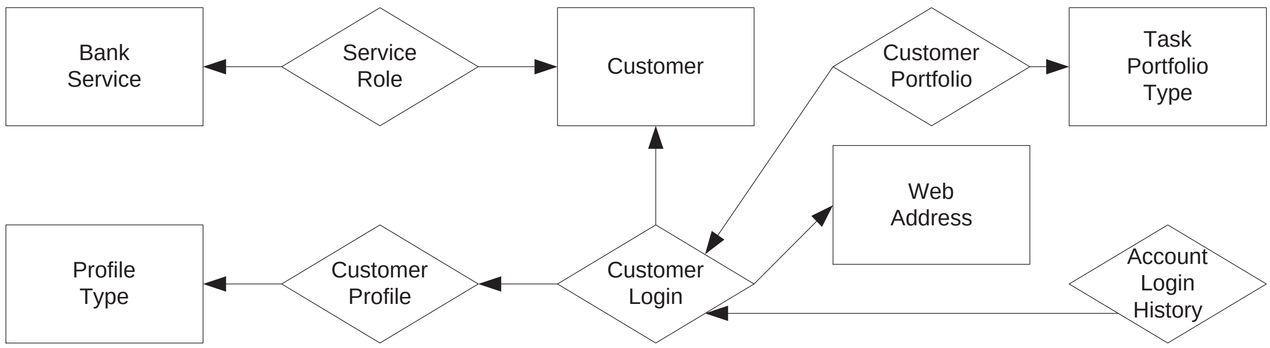


Figure 2.3: Cutout of the profiling schema

The core of a *media type* is defined by a view on some underlying database schema, i.e. it consists of a view schema and a defining query. However, this query must be able to create identifiers in order to create links between the various media objects. This core of a media type — called *raw media type* in (Schewe and Thalheim 2001) — is extended in three directions:

- As a first extension operations are added to the view in the same way as d-operations were added to dialogue objects in (Schewe and Schewe 2000). Basically, the use of operations just adds dynamics to the media objects. So, if a media object is associated with a scene, the operations of the media object define the available dynamic functionality.
- The second extension provides adaptivity and hierarchies. Adaptivity to the user deals with needs arising from different users. Adaptivity to the technical environment copes with technical restrictions of end-devices. Adaptivity to the communication channel deals with adaptation to needs arising from various communication channels. For all three forms of adaptivity media types provide mechanisms for a controlled form of information loss, which is coupled with algorithms for the splitting of information content. The hierarchies are adopted from dimension hierarchies in OLAP.
- The third extension simply covers ordering and other presentation options.

Thus, roughly speaking media objects consist of *abstract containers*, *supported DBMS processes* and *database manipulations requests*. Basic media objects (Schewe and Thalheim 2000) are characterized by syntactic expressions, have a semantical meaning and are used within a certain pragmatical framework. Media objects can be parameterized. Typical parameters are the representation style, the actor frame, and the context frame. Therefore we distinguish between media objects and runtime media objects in which all parameters are instantiated.

During runtime, the media object is extended by specific *escort information* (Thalheim 2000). This escort information is represented for user support. It allows the user to see the history of steps performed before being in the current state. Escort information is further generated from the story space. In this case a user is informed on alternative paths which could be used to reach the given scene and which might be used for backtracking from the current scene.

For the generation of media objects and their composition on the basis of information units we extend the classical SQL frame to the frame

`generate` MAPPING : VARS → STRUCTURE
  `from` VIEWS
  `where` SELECTION CONDITION
  `represent using` STYLE GUIDE
   & ABSTRACTION
  `browsing definition` CONDITION
   & NAVIGATION

The views and therefore the media object may have hidden parameters (for instance, EventID) which are not visible to the actor. They can be parameterized by variables (for instance, @Today). For media objects we reuse ideas developed for OLAP technology (Thalheim 2000):

- *views on ER schemata* (abstraction on schemata (aggregation, scoping, ...), versions),
- *variations of generation functions*,
- *display with canonical functionality* (drill-down, roll-up, rotate, pivoting, push, pull, dimension, aggregation),
- *using generic evaluation functions and models*,
- *implicit incorporation of hierarchies* and
- *implicit incorporation of time, space, ...*.

Furthermore, *involved actors* are specified in dependence on their profiles, tasks assigned to them, their access and manipulation rights, and their roles to be taken while visiting the scene. This specification is based on (Altus 2000) and similar to profiles of actors in information systems.

It is our aim to specify generic scenes. Thus, we add the *representation styles* which can be applied to the media object of the scene. Representation depends on the equipment of the actor. In the city site projects, we have gained experience with different representation styles: internet display with high-speed channel, internet-display with medium speed display (default style), videotext and WAP display. For instance, for videotext any graphical information is cut out or replaced by textual information.

Finally, the context of access is specified. Access determines the display facilities. Channels can be of high or low speed. The particular usage of a scene by an actor depends on the scenario history.

The login scene in Figure 2.2 is based on the schema in Figure 2.3.

The corresponding media object specification has the following structure:

MEDIAOBJECT(@*Customer_ID*) =
`generate` *(ID, profile, portfolio, context)*
 `from` *Customer* ⋈ *Login_Account_History*
   ⋈ *Customer_Profile*
   ⋈ *Customer_Portfolio* ⋈ ...
 `where` *Customer.ID* = @*Customer_ID* ...
 `represent using`

*XSL_style.Ident* =
*Profile_Type.Preference.StyleIdent*
& *createVarsFor(profile, portfolio, context)*
`browsing definition` *Customer* $\succ$ *portfolio* $\succ$ *...*
& NAVIGATION *none*

The representation styles determine the order and the tailoring of the elements of the media object.

### 2.4 Dialogue Steps

We conceptualize the customer-WIS interaction as a dialogue between these two. Therefore the customer-WIS interaction unfolds in a sequences of dialogue steps, i.e., elementary communication acts. The basic WIS-state transformations triggered by actors can thus be understood as caused by dialogue steps. These may access the media object that is associated to the scene within which the dialogue step occurs. Comparable to (Goldin et al. 2000) we use the following frame for specifying the control of dialogue steps:

`on` *precond* `if` *event* `and` *guard*
`do` *action* `result in` *postcond*

Consequently dialogue steps may be specified by the following frame:

DIALOGUESTEP(Identification ) =
( SUB-UNIT = view on media object of the scene
ENABLED PROCESSES =
subset of supplied processes,
manipulation requests
ACTOR =
subset of enabled actors in a given context
CONTROL = ( precondition, enabling event,
guard, postcondition) )

Dialogue step specifications can be represented graphically as shown in figure 2.4. The figure for the scene 'anonymous login' represents the specification of dialogue step 'login'.

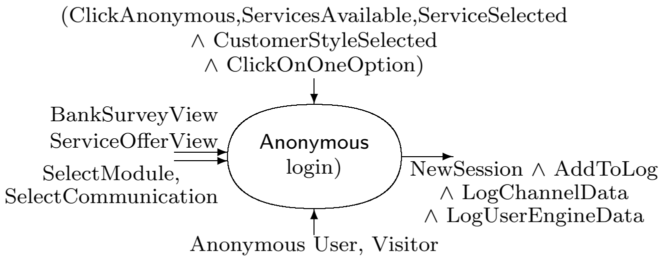


Figure 2.4: Dialogue Step for Anonymous Login

Based on the properties of the actions we conclude, for instance, that after withdrawal a previous member of a cooperating group cannot participate in the discussions in the community. A task property frame is defined by a task name, reasons for task involvement, an aim, a postcondition (enabled next activities), the information from the database, the information for the database, the resources (actor, resources, partner), and a starting situation (precondition, activity, priority, frequency, repetition rate).

We use graphical representations of scene specifications as indicated by figure 2.5. Scenes are represented by frameboxes and dialogue steps by ellipses. The transitions among dialogue steps are represented by arrows between these. We use the graphical notation developed for state charts, e.g., the default start step of a scene is denoted by a solid circle, the end state by a solid circle surrounded by an empty circle, the history entry into a scene is denoted by an 'H' surrounded by an empty circle. Furthermore, we can adopt refinement and clustering, concurrency, delays and time-outs, transient states, event priorities and parameterized states. For more detail on state charts see, e.g. (Harel and Naamad 1996) and for their application (Rumbaugh et al. 1991).

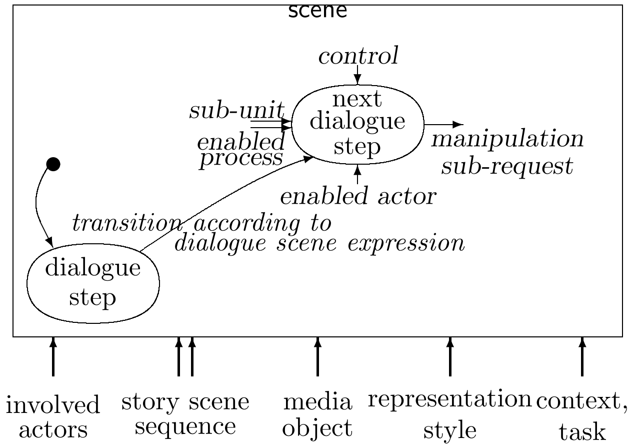


Figure 2.5: Representation of scene specifications

### 2.5 Context

Context has been usually defined within the object sets of the database (Bell 2001, Connolly 2001). There only very few trials to consider context of the scenarios or stories (Whitsey 2003). In (Thalheim 2000a) context has been defined for media types. For dealing more complete and justifiable with context we start with a dictionary definition of context of something as that what one needs to understand the something. This implies our understanding of context as a three place predicate $C(S, H, A)$ which if true says that actor $A$ needs helper $H$ to act reasonably on $S$. If the actor is an individual then we stay with the focus on understanding. For non-human actors, however, we focus on acting according to predefined quality aspects and behavior rules. The something we consider here as relevant are WIS-parts. The helpers we here take into account are the various data that are relevant for the WIS-parts in question. The actors we consider here are the WIS and the individuals occupying the roles: customer, vendor and developer with respect to the WIS at hand. We thus distinguish the following contexts:

**Customer's scenario context** , i.e., that what the customer needs to understand for efficiently and effectively solve his/her business problem.

**Vendor's WIS-context** , i.e., that what the vendor needs to understand how to run the WIS economically. Data that typically is part of this context are:

- the intention of the provider,
- the theme of the web site,
- the mission or corporate identity of the site, and
- the occasion and purpose of the visits of actors.

**Developer's WIS-context** , i.e., that what the developer needs to understand for being capable of implementing the WIS. Data that typically is part of this context are:

- the potential environment, e.g. hard- and software, channels,
- the information system, especially the associated databases,

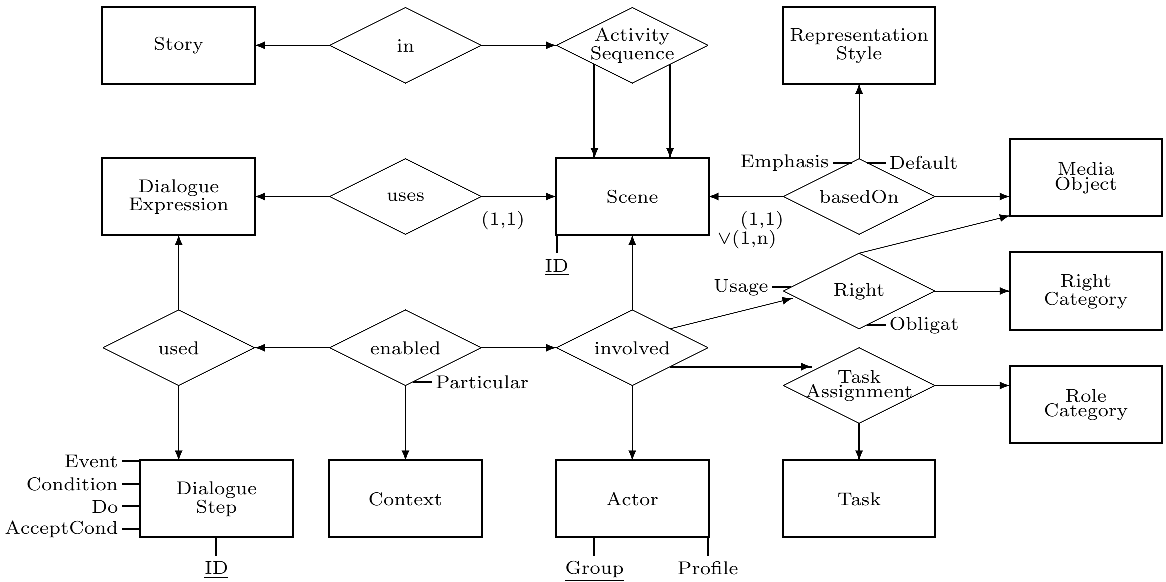


Figure 3.1: The Structure of the Web Site Database

- the story space, scenes, dialogue steps, roles, and rights,
- the tasks to be performed within the story, and
- the roles in the scenario.

WIS's scene context , i.e., that what the WIS needs to be capable of making solving certain business problems easy and pleasant for customers. Data that typically is part of this context are:

- History and current usage allow context adaptation to scenarios which are played at present by the current user.
- Adaptation to the current environment is defined as context adaptation to the current channel, to the client infrastructure and to the server load.
- Users are grouped to actors. Therefore, we can define the current user by instantiation of the actor.
- Goals and particular, policy (exceptions, social, organizational) define a specialization of the content, structuring and functionality of a web page.

A WIS is supported by media objects that belong to media types. The collection of all media types is called *suite*. Our framework offers four hooks for dealing with the context we need to consider:

1. Specialization of media type suite for usage and user adaptation:

   The database types may have subtypes specializing the database types. Media types are defined on the basis of views. Therefore, we can follow the approach discussed in (Thalheim 2000a) for specialization of types. Specialization is definable through specialization of types, instantiation of parameters. extension of types, and restriction and constraint application.

2. Application of rules towards generation of extended suites:

   Suites can be extended by providing view rules defining views on top of the media types. This approach supports portfolio extension and container extension.

3. Instantiation of explicit context parameters can be used for adaptation of web sites to the current profile, to the current environment or to the current workload.

4. Storage of utilization profile similar to login track supports to use history of previous utilization of the web site by the user. We extend the web site database by explicit utilization logs for used media objects, preferences of usage, users workspace or work rooms, and variations of users media objects.

## 3 Developing the Database Used to Generate the Web Site

### 3.1 Database Modelling for WIS

Web site management becomes a nightmare whenever a web site has been developed in a handicraft approach. For this reason, generation of web sites is currently based on web site content management. Content management systems currently support the representation of web pages on a take-and-place metaphor: Select or compile the content objects of a page, compile the navigation structure and place the content objects using page frames. Our web site team also used this approach. This approach is entirely satisfying the needs as long as the general structure of a web site is stable and no adaptation to the user is required.

In order to dynamically generate the web site we decided to store the web site stories in a database. The structure of this database is displayed in figure 3.1.

We specify the web site story space based on the our web SiteLang. This specification is inserted into the database by the SiteLang editor. We can now extract the page under consideration by an instantiated query from this database. Context may be infused directly depending on the query result.

Similar to the context infusion, users of a web site have their own profile, their own portfolio and their history. This information is used for adapting the content of the web site to the current usage.

### 3.2 Context Infusion in Scenarios

Typical business processes have a very large number of variants. Classically, workflow approaches have been used for specification of such varieties. Since the complexity of variants might be much higher the workflow approach did not succeed in providing a sound basis for the specification of all variants. We observe, however, that in practice these varieties are internally structured. They may be composed, extended or filtered by smaller scenarios. e-banking challenges storyboarding by its orthogonality and variety.

Instead of specifying all possible variants we prefer to model the generation mechanism of the very large variety of scenarios. This generation supports runtime adaptation to the current scenario, the context and other parameters. At the same time, banking sites are threatened to expose customers to the "lost in hyperspace syndrome". Therefore, customers should be supported in tracking back onto the right path.

Our solution to this challenge is based on generic parameters that are instantiated depending on the customer, the history, the context etc. Each set of media objects is specified by a context-free expression with a set of parameters. These parameters are instantiated depending on

- the customer profile,
- the customer task portfolio,
- the customer computational environment,
- the presentation environment, and
- the available and accessible media objects.

Instead of providing a full generation rule set we illustrate our approach on the basis of an example. A customer of a bank provides his/her identity $e_1$, inserts some data $e_{2,1}$ and $e_{2,2}$ in any order or signs that the bank may request these data from somewhere else $e_{2,3}$. Then the customer seeks a loan and fills the corresponding forms $e_3$. The customer gives bail data in different variants $(e_{5,1} \,||\, (e_{5,2} \,;\, e_{5,3}))$. The scenario is supported by the eight media objects. Now we can inject the context into the media object expression of the scenario. For instance, we may have the following stepwise refinements:

- Media objects of a scenario: $e_1 \,;\, ((e_{2,1} || e_{2,2}) \ltimes | \, e_{2,3}) \,;\, e_3 \,;\, (e_{5,1} \,||\, (e_{5,2} \,;\, e_{5,3}))$
- Extending by objects syntactic verbal context and meta-information:
  $\underline{\underline{e_{16} \,;\, [\, e_{21} \,;\, ]}}\, e_1 \,;\, ((e_{2,1} || e_{2,2}) \ltimes | \, e_{2,3}) \,;\, \underline{e_9} \,;\, e_3 \,;\, \overline{(e_{10} || e_{11})} \,;\, (e_{5,1} \,||\, (e_{5,2} \,;\, e_{5,3}))$
- Extending by story space associations, e.g., side paths, , filtering against availability and compiling against the customer profile
  $e_{16} \,;\, [\, e_{21} \,;\, ]\, e_1 \,;\, ((e_{2,1} || \underline{\underline{( {}_{SB} e_{2,2} || {}_{CB} e_{2,2} )}}) \,|\, \times \,|\, e_{2,3}) \,;\, \underline{[(\nearrow e_{17}; e_{18} \searrow ;)]}\, e_9 \,;\, \underline{\underline{\underline{{}^{Gr} e_{3,1} \,;\, {}^{An} e_{3,2} \,;\, {}^{Inf} e_{3,3} \,;\, {}^{Form} e_3}}} \,;\, (e_{10} || e_{11}) \,;\, (e_{5,1} \,||\, (e_{5,2} \,;\, e_{5,3}))$
- Filtering with or extending by the web site context: $e_{16} \,;\, [\, e_{21} \,;\, ]\, e_1 \,;\, (\underline{\otimes}\ {}_{SB} e_{2,2}\ \underline{\otimes} \ltimes | \, e_{2,3}) \,;\, [(\nearrow e_{17}; e_{18} \searrow ;)]\, e_9 \,;\, {}^{Gr} e_{3,1} \,;\, {}^{An} e_{3,2} \,;\, {}^{Inf} e_{3,3} \,;\, {}^{Form} e_3 \,;\, (e_{10} || e_{11}) \,;\, (\underline{\otimes}\, e_{5,2} \,;\, e_{5,3})$
- Coping customer's history - already finished dialogue steps and repeating dialogue steps: $\underline{e_1^{Repe}} \,;\, [(\nearrow e_{17}; e_{18} \searrow ;)]\, \underline{e_9^{Repe}} \,;\, {}^{Gr} e_{3,1} \,;\, {}^{An} e_{3,2} \,;\, {}^{Inf} e_{3,3} \,;\, {}^{Form} e_3 \,;\, (e_{10} || e_{11}) \,;\, (e_{5,2} \,;\, e_{5,3})$
- Coping with customers history - negotiation steps and pragmatical elements:
  $e_1^{Repe} \,;\, \underline{e_{25}} \,;\, [(\nearrow e_{17}; e_{18} \searrow ;)]\, e_9^{Repe} \,;\, {}^{Gr} e_{3,1} \,;\, {}^{An} e_{3,2} \,;\, {}^{Inf} e_{3,3} \,;\, {}^{Form} e_3 \,;\, (e_{10} || e_{11}) \,;\, (e_{5,2} \,;\, \underline{\underline{e_{5,2}^{Prak}}} \,;\, e_{5,3})$

### 3.3 The Onion Generation

XML documents provide a universal structuring mechanism. XSL rules allow to generate XML documents from XML suites. This opportunity supports a multi-layer generation of web information systems. Thus we use the multi-layer *onion generation* presented in Figure 3.2.

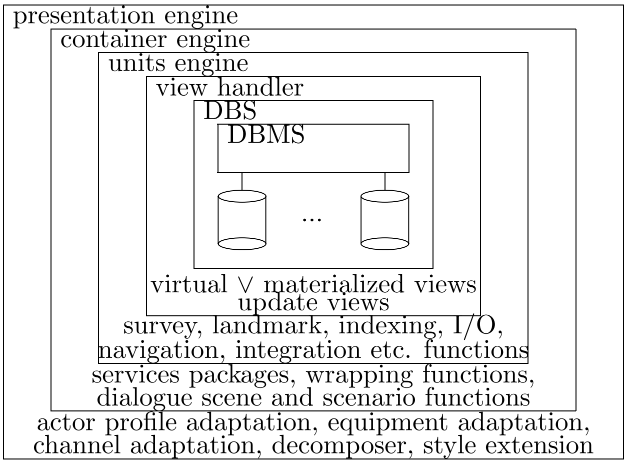


Figure 3.2: The Onion Approach to Stepwise WIS-Generation

The onion generation approach is based on the layered structure of the WIS arising from the use of SiteLang and media objects. On the outermost shell the presentation facilities are introduced. This shell deals with style presentation functions. Containers used in the next inner shell are used to ship information from the web-server to the user. Thus, this shell deals with the adaptation to the user and his/her environment. The next inner shell handles the information units, i.e. the core media objects. Inside this shell we find further shells dealing with views on the underlying database, and innermost we find the database itself.

The onion approach fits nicely into a translational approach, which generates consistent sets of XML documents. In our projects we used the XML extender of the database system DB2 to generate XML documents. Thus, the layering approach to the generation of XML displayed in Figure 3.2 allows to use another strategy to generate XML documents. This facility is displayed in Figure 3.3.

This transformation approach has been successfully used in two of our e-learning projects and our community services projects. These project require sophisticated context adaptation. The approach implements an XML suite on top of the relational DBMS DB2. The extended ER model (Thalheim 2000) provides a better approach to XML suite generation than relational models or the classical ER model for a number of reasons:

- Structures can be defined already in complex nested formats.
- Types of higher order are supported.
- The model uses cardinality constraints with participation semantics.

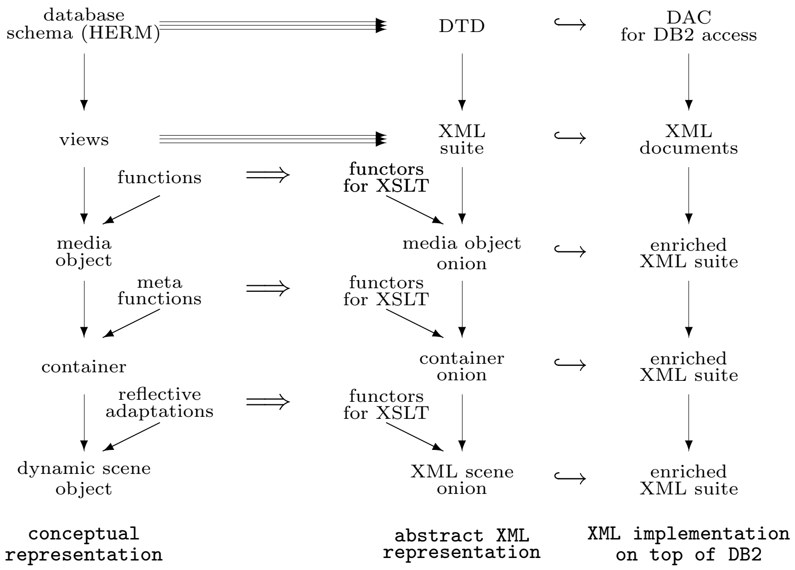


Figure 3.3: The General Procedure for Translation from SiteLang to XML

## 4 An Advanced e-Banking Application

### 4.1 Banking and Mortgages

According to (Wierichs and Smets 2001) a bank is an "... institution that as part of an economy offers financial services. The economical function of banks is to create a liquidity equalization in the cash flow that is reverse to the product and service flow. The focal points of the bank operational activity are conducting payments, acceptance of money for investment, and granting credits." Furthermore the particular liquidity equalization that is chosen out of the set of possible such equalizations is a preferable one. The respective preference structure is worked out by banks on base of an assessment involving financing cost and interests, see (Matthews et al. 2003).

A loan according to (Wierichs and Smets 2001) is the "relinquishment of money or other fungible (...) properties connected with the obligation of the debtor to give back the relinquished in equal kind, quality and quantity." An enhanced version of the model of the loan process is represented in figure 4.1 as a UML sequence diagram. It shows the roles involved in the loan process as the labels inside the rectangular boxes on the top of the diagram. It further indicates the concurrency that may be utilized in this process. It achieves this by means of showing the communication between the roles. This communication is represented by the arrows starting at the dashed lines that represent logical time, i.e., life lines of the roles. The labels attached to the arrows indicate the content of the message associated with the respective arrow. The bottom level rectangle containing the messages 'Payback()' and 'CheckPayback()' signifies that these messages are to be repeatedly sent until the stop condition signified by the asterisk and displayed below the rectangle 'debit position balanced' becomes true.

### 4.2 Mortgages and variants

The figure 4.1 from a bank technical point of view schematizes the process. This process clearly is not fully suited as the only base of application development. For aiding development more information is needed about how customers are anticipated to interact with the system under construction. We use here the function $\kappa_W$ of the story space $\Sigma_W$ of a WIS $W$ to show how the customer interaction with the WIS changes the appearance of it for the customer Story boarding is a useful technique to obtain the required information.

Our respective starting point is the investigation of the Web site of the Australian and New Zealand based ASB Bank. From earlier work, see (Kaschek et al. 2003) we knew that it offered an online loan application facility. We investigated this Web site more closely and found that this site at each of its pages essentially offered customers data that can be typed as follows:

- **advertisement**, i.e., information about ASB Bank including a welcome and a logo.
- **disclaimer**, i.e., a statement limiting the legal responsibility of ASB Bank with respect to the data displayed and the implications customers might draw from it.
- **search**, i.e., a facility taking an unlimited customer input and returning those ASB Bank pages that best met this search expression.
- **highlights**, i.e., the main contents that ASB Bank wants to be displayed at each particular of its Web pages.
- **path**, i.e., a redundancy eliminated sequence of ASB Bank Web pages visited so far by the customer interacting with the site and supposed to be used as a navigation aid.
- **reference**, i.e., a couple of links the target of which offer more information about the page actually visited by the customer.
- **business branch selector**, i.e., a navigation bar that breaks down the information space of the site into subspaces according to the business branches of ASB Bank.
- **subspace selector**, i.e., a navigation bar that for each subspace that corresponds to a business branch breaks down the subspace into 2nd. level subspaces.
- **subspace navigator**, i.e., for each 2nd. level subspace a navigation bar breaking down the subspace in a number of information space locations.

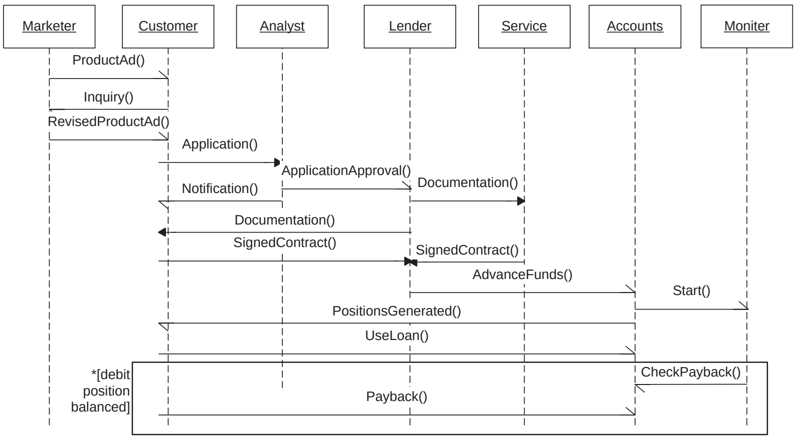


Figure 4.1: UML diagram representing the loan process

We have then represented the navigation structure offered by the ASB Web site as a state chart the states of which represent scenes. The state transitions are presupposed to be triggered by customers clicking links, i.e., navigation events. The labels attached to the state transitions are a string representing the navigation event and an action carried out throughout the transition. This action is prefixed by a slash, i.e., by "/". The semantics of the action is specified in form of a programming language like assignment and assigns new values to variables holding the data listed above. In this way one can specify what data and functionality is accessible to a customer at a particular scene. Explanations and tables or the like are presupposed to be just text. All other transition labels used as values in assignments are presupposed to be links. If a variable is supposed to hold several links then these are connected by a plus sign, i.e., by "+". If more than one action has to take place at a transition then all these actions are connected by a &-sign.

The variables used in figure 4.2 are D, S, H, A, R and P respectively representing values of type disclaimer, search facility, highlight, advertisement, references and path. Those of them being displayed at a particular page are represented as non delimited string, i.e., if all of them occur the string DSHARP is attached as label to the state representing the page. Furthermore the variables BS, SS and SN are used to respectively represent values of type business branch selector, subspace selector and subspace navigator. The initial state in the figure is reached after moving onto the home page of ASB Bank and clicking BS.Personal which signifies the business branch of retail banking. The other 1st. level subspaces of the application's information space are "All", "Business", "Institutional" and "Rural" in the obvious meaning. The subspace selector of "BS.Personal" allows to chose from 18 different 2nd. level subspaces. One of them is Home loans. Clicking it, i.e., "BS.Personal-SS.Home loans" leads to the initial state of figure 4.2.

If required we could add further navigation detail including the impact of navigation on the variables occurring in the figure. Furthermore if it would be required we could add further variables to represent data of types here not dealt with.

### 4.3 Adaptation to customers, context and specific case

Adaptation to customers is a must if optimal customer support is aimed at. ASB Bank realizes a limited customer adaptation in that it offers in the subspace selector of BS.Personal second level subspaces both for kids and for young folks. ASB Bank concerning the home loan subspace of its information space does not offer much adaptation to customers. It only offers a bank technical terms dictionary and specifically addresses first home buyers. Neither are all New Zealand official language versions of the Web site available nor can it be tuned to meet any kind of disabilities such as weak eyesight or color blindness.

The approach to adaptation to customers taken by sites like the one under investigation consists in identifying the subspace of the information space they create that most likely will fit best the needs of a particular customer. The match between customer and subspace is then done such that the customer is asked to give some characteristics of him or her into the system and based on that the respective subspace is chosen. ASB Bank does so concerning kids and young folks. Other banks have additionally the customer type student or wealthy individual. This strategy is suggested by the fact that the site vendor in general does not know much about the individuals accessing its site. A technique to consider knowledge about customers to the design process is the creation and use of personas, i.e., archetypical customers and design the navigation structure as well as the page layout such that it fits optimally to the personas used. Concerning more detail about personas in particular their construction see, e.g. (Wodtke 2003, pp. 159)

Adaptation of the business case at hand of course can only be achieved in response to the customer-site interaction. In the navigation structure diagram in figure 4.2 we have used variable of data types that were chosen with respect to the site at hand, i.e., ASB Bank's Web site. We expect that this adaptation can always be achieved the way we have proposed here. Once the analysis has shown what data and functionality shall be accessible to customers data and functionality can be typed and variables of the respective type can be used to describe how the site adapts to the actual use. A type level adaptation that can be carried out while customers are interacting with a site is semi automatic reconsideration of the type of customer: Customers in this respect are presupposed to be characterized by a value for each of a number of

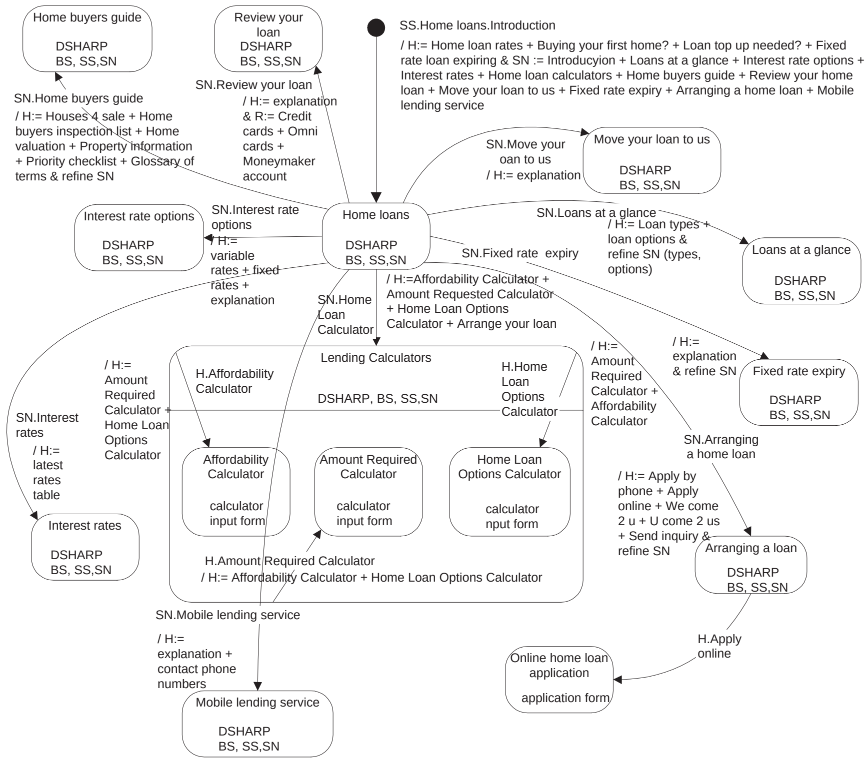


Figure 4.2: Navigation structure of a part of ASB Bank's Web site

dimensions. The customer type according to (Schewe and Thalheim 2001) can be defined as a convex region in the multi dimensional space create as cartesian product of the scales associated to the customer dimensions. Based on an automatic customer assessment that in response to his or her site-interaction updates the scores in each of the dimension throughout customer-site interaction one can then track how a customer's trace moves through this space and detect when a modified type would better fit the customer's behavior than the actual type does. Clearly such update should only be done with customer permission.

## 5 Conclusion

Banking services such as online home-loan application require a very sophisticated and well-adapted internet interface. Customers want to focus on solving their business problem, i.e., the goal they want to achieve by means of interacting with the WIS. They consider WIS as tools that shall be easy to handle, completely cover the business and do not add technical complexities to it. Customers want to have a pleasant usage experience, in particular they do not want to be treated like everybody else. They want WIS remember and exploit their usage peculiarities in authenticated and where adequate in anonymous sessions. This paper shows how this can be achieved. As a guiding principle we introduce considering context and using it to simplify WIS-handling. We have shown how WIS's scene context can be injected into the WIS and how XML suites can be generated using the story board of the site and available customer data.

### Acknowledgements

We would like to thank Hans-Jürgen Engelbrecht from the Department of Applied & International Economics at Massey University for pointing to us work on the the economic effects of technical progress in the banking industry.

### References

Altus, M. *Decision support for conceptual database design based on the evidence theory - An intelligent dialogue interface for conceptual database design.* PhD thesis, Faculty of Mathematics, Natural Sciences and Computer Science of BTU Cottbus, Cottbus, 2000.

Atzeni, P., Gupta, A., and Sarawagi, S. Design and maintenance of data-intensive web-sites. In *Proceeding EDBT'98*, vol. 1377 of *LNCS*. Springer-Verlag, Berlin, 1998, pp. 436–450.

Baresi, L., Garzotto, F., and Paolini, P. From web sites to web applications: New issues for conceptual modeling. In *ER Workshops 2000*, vol. 1921 of *LNCS*. Springer-Verlag, Berlin, 2000, pp. 89–100.

Bell, J. Pragmatic reasoning: Inferring contexts. In *Proc. Context'1999* (1999), LNAI 1688, Springer, pp. 42–53.

Berger, A. N. The Economic Effects of Technological Progress: Evidence from the Banking Industry. *Journal of Money, Credit, and Banking 35*, 2 (2003), 141 – 176.

Bonifati, A., Ceri, S., Fraternali, P., and Maurino, A. Building multi-device, content-centric applications using WebML and the W3I3 tool suite. In *ER Workshops 2000*, vol. 1921 of *LNCS*. Springer-Verlag, Berlin, 2000, pp. 64–75.

Workflow Management Coalition, Ed. *The Workflow Management Coalition specification: Workflow Management Coalition terminology & glossary.* Workflow Management Coalition, Winchester, United Kingdom, 1999. Document Status Issue 3.0.

Connolly, J. H. Context in the study of human languages and computer programming languages: A comparison. In *Proc. Context'2001* (2001), LNAI 2116, Springer, pp. 116–128.

Düsterhöft, A., and Thalheim, B. SiteLang: Conceptual modeling of internet sites. In *Conceptual Modeling – ER 2001*, H. S. K. et al., Ed., vol. 2224 of *LNCS*. Springer-Verlag, Berlin, 2001, pp. 179–192.

Feyer, T., Schewe, K.-D., and Thalheim, B. Conceptual modelling and development of information services. In *Conceptual Modeling – ER'98*, T. Ling and S. Ram, Eds., vol. 1507 of *LNCS*. Springer-Verlag, Berlin, 1998, pp. 7–20.

Gädke, M., and Turowski, K. Generic web-based federation of business application systems for e-commerce applications. In *EFIS 1999*. 1999, pp. 25–42.

Garzotto, F., Paolini, P., and Schwabe, D. HDM - a model-based approach to hypertext application design. *ACM ToIS 11*, 1 (1993), 1–26.

Goldin, D., Srinivasa, S., and Thalheim, B. Is = dbs + interaction - towards principles of information systems. In *Proc. ER'2000* (2000), LNCS 1920, Springer, pp. 140–153.

Harel, D., and Naamad, A. The STATEMATE Semantics of Statecharts. *ACM Transactions on Software Engineering and Methodology 5*, 4 (Oktober 1996), 293–333.

Jablonski, S. *Workflow-Management-Systeme: Modellierung und Architektur*. Thomson's Aktuelle Tutorien. International Thomson Publishing, Bonn, Germay et al., 1996.

Kaschek, R., Matthews, C., and Wallace, C. e-Mortgages: NZ State of the Art and Perspectives. In *Proceedings of SCI 2003* (2003).

Kaschek, R., Schewe, K.-D., Thalheim, B., Zhang, L. Modelling contexts in web information systems. Proc. WES 2003.

Matthews, C. D., Kaschek, R. H., Wallace, C. M., and Schewe, K. D. IST in Lending: Unlimited potential but limited practice. Available from: http://cbs.dk/staff/lars.heide/ISTOS/program.htm, 2003. Paper presented at ISTOS Workshop in Barcelona, Spain, 28-30 March 2003.

Rossi, G., Schwabe, D., and Lyardet, F. Web application models are more than conceptual models. In *Advances in Conceptual Modeling*, P. C. et al., Ed., vol. 1727 of *LNCS*. Springer-Verlag, Berlin, 1999, pp. 239–252.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. *Object-Oriented Modeling and Design*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1991.

Schewe, B. *Kooperative Softwareentwicklung*. Deutscher Universitätsverlag, Wiesbaden, Germany, 1996.

Schewe, B., Schewe, K.-D., and Thalheim, B. Objektorientierter Datenbankentwurf in der Entwicklung betrieblicher Informationssysteme. *Informatik – Forschung und Entwicklung 10* (1995), 115–127.

Schewe, K.-D., Kaschek, R., Matthews, C., and Wallace, C. Modelling web-based banking systems: Story boarding and user profiling. In *Proceedings of the Workshop on Conceptual Modelling Approaches to E-Commerce*, H. Mayr and W.-J. Van den Heuvel, Eds. Springer-Verlag, 2002.

Schewe, K.-D., and Schewe, B. Integrating database and dialogue design. *Knowledge and Information Systems 2*, 1 (2000), 1–32.

Schewe, K.-D., and Thalheim, B. Modeling interaction and media objects. In *Proc. NLDB' 2000* (2000), LNCS 1959, Springer, pp. 313–324.

Schewe, K.-D., and Thalheim, B. Modeling interaction and media objects. In *Advances in Conceptual Modeling*, E. Métais, Ed., vol. 1959 of *LNCS*. Springer-Verlag, Berlin, 2001, pp. 313–324.

Schwabe, D., and Rossi, G. An object oriented approach to web-based application design. *TAPOS 4*, 4 (1998), 207–225.

Siegel, D. *The secrets of successful web sites*. Markt und Technik, München, 1998.

Sowa, J. F., and Zachman, J. A. Extending and formalizing the framework for information systems architecture. *IBM Systems Journal 31*, 3 (1992), 590 – 616.

Srinivasa, S. *A Calculus of Fixed-Points for Characterising Interactive Behaviour of Information Systems*. PhD thesis, BTU Cottbus, Fachbereich Informatik, Cottbus, 2001.

Thalheim, B. *Entity-relationship modeling – Foundations of database technology*. Springer, 2000. See also http://www.informatik.tu-cottbus.de/∼thalheim/HERM.htm.

Thalheim, B. Readings in fundamentals of interaction in information systems. Reprint BTU Cottbus, 2000. *Collection of papers by C. Binder, W. Clauß, A. Düsterhöft, T. Feyer, T. Gutacker, B. Heinze, J. Lewerenz, M. Roll, B. Schewe, K.-D. Schewe, K. Seelig, S. Srinivasa, B. Thalheim.* Accessible through http://www.informatik.tu-cottbus.de/∼thalheim.

Thalheim, B., and Düsterhöft, A. Sitelang: Conceptual modeling of internet sites. In *Proc. ER'2001* (2001), LNCS 2224, Springer, pp. 179–192.

Valentine, T. Lending: a survey. In *Readings in financial institution management. modern techniques for a global industry*, T. Valentine and G. Ford, Eds. Allen & Unwin, Crows Nest, NSW, Australia, 1999, pp. 223 – 253.

van Duyne, D. K., Landay, J. A., and Hong, J. I. *The design of sites*. Addison-Wesley, Boston, 2003.

Whitsey, M. Discourse context and indexicality. In *Proc. Context'2003* (2003), LNAI 2680, Springer, pp. 356–386.

Wierichs, G., and Smets, S. *Gabler Kompakt-Lexikon Bank und Börse*. Gabler Verlag, Wiesbaden, 2001.

Wodtke, C. *Information Architecture: Blueprints for the Web*. New Riders, Indianapolis, Indiana, 2003.

Zachman, J. A. A framework for information systems architecture. *IBM Systems Journal 26*, 3 (1987), 276 – 292.