

## Association for Information Systems AIS Electronic Library (AISeL)

---

ACIS 2004 Proceedings

Australasian (ACIS)

---

December 2004

# Towards Systematic Personalization of Information Spaces

Roland Kaschek  
*Massey University*

Klaus-Dieter Schewe  
*Massey University*

Catherine Wallace  
*Massey University*

Claire Matthews  
*Massey University*

Follow this and additional works at: <http://aisel.aisnet.org/acis2004>

---

### Recommended Citation

Kaschek, Roland; Schewe, Klaus-Dieter; Wallace, Catherine; and Matthews, Claire, "Towards Systematic Personalization of Information Spaces" (2004). *ACIS 2004 Proceedings*. 104.  
<http://aisel.aisnet.org/acis2004/104>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Towards Systematic Personalization of Information Spaces

Roland Kaschek<sup>1</sup>, Klaus-Dieter Schewe<sup>1</sup>

Catherine Wallace<sup>2</sup>

Claire Matthews<sup>3</sup>

Massey University, Palmerston North, New Zealand

<sup>1</sup>Department of Information Systems

[R.H.Kaschek@massey.ac.nz](mailto:R.H.Kaschek@massey.ac.nz), [K.D.Schewe@massey.ac.nz](mailto:K.D.Schewe@massey.ac.nz)

<sup>2</sup>Department of Communication and Journalism

[C.M.Wallace@massey.ac.nz](mailto:C.M.Wallace@massey.ac.nz)

<sup>3</sup>Department of Finance, Banking and Property

[C.D.Matthews@massey.ac.nz](mailto:C.D.Matthews@massey.ac.nz)

## Abstract

*Web information systems often have large and complex information spaces. These spaces serve as a conceptual framework for all user-system interactions. Their complexity is likely to cause users having trouble carrying out with these systems the business cases in their portfolio. Aiding users therefore requires a view of reduced complexity on the information space being defined and made accessible to the users. Such a view is called a story. Working out the stories that best help users beneficially use Web information systems is the task of information space personalization. This paper analyses a Web information system. A Kleene algebra is used as a mathematical model of a story of this system. We illustrate that formal manipulation of Kleene algebras can be used for personalizing information spaces. This paper thus supports the proposal of using Kleene algebras as mathematical model of Web information system usage.*

## Keywords

Web information system, story boarding, information space, SiteLang, codesign, Kleene algebra

## INTRODUCTION

From a functional perspective and for this paper we adopt the definition of information systems (that according to Hirschheim et al. (1995) is due to Langefors) as technically implemented media for recording, storing, disseminating (and processing) linguistic expressions. Web information systems (WIS) are information systems that are accessible via the Web. This connection to the Web has a number of implications that introduce specific characteristics to WIS. The latter ones have been distinguished from other information systems in Kaschek et al. (2004a) by their openness with respect to users (virtually everyone can use them), data (input can be easily provided and output easily post-processed with other WIS), technology (a variety of devices may be used for interaction), and access channels (a variety of transmission media and protocols is available). This openness comes with a price: increased complexity. It increases the role of systems analysis and design for non-trivial systems.

During systems analysis a business model is obtained that captures requirements and domain knowledge. A variety of semantic models and analysis methods exists and is used for systems analysis. Even languages are available with which executable system specifications can be obtained. These languages and respective approaches help identify functional requirements and implement respective prototype systems. These languages in general are more helpful regarding functional rather than non-functional requirements. This is the case because they are limited in the design choices offered to developers. Therefore, systems design is a task that still needs to be carried out and involves a lot of ingenuity and human work. Its purpose is creating a model of the implementation of a computerized system (in our case a WIS) that meets the functional and non-functional requirements in so far as this is reasonable under the constraints applying to the actual project.

According to Banker et al. (1993) Basili defined the complexity of a system S (as we may add here: with respect to a system T) as the amount of resources needed by another system T for its interaction with S. One strategy, we call it complexity reduction strategy, for reducing the complexity for WIS users, is providing access to only those system resources that are needed for carrying out the business cases belonging to the user's portfolio. It appears to be a reasonable assumption that for each business case in a user portfolio there is a minimum complexity required for satisfactorily dealing with this case. Another strategy, we call it complexity handling

strategy, that could be applied is making it easier to understand how to successfully carry out the cases in a portfolio with a WIS by providing an adequate language to use it. An approach to the latter is outlined in Tretiakov and Kaschek (2004). In this paper we restrict the discussion to the complexity reduction strategy. The complexity required by user portfolios may vary considerably over the users. Complexity reduction thus is a promising strategy as soon as a considerable number of user types of the WIS under development are identified.

We aim at the user complexity required for a WIS being minimized. We address this goal by the optimum aid that can be provided to users. It is, however, neither obvious what exactly optimum user aid is, nor is it clear how to tailor the information space such that optimum user aid can be provided. We summarize our respective working assumption:

- Optimum user aid is providing a WIS interface of minimum user complexity that enables the user carrying out comfortably the processes in his / her portfolio.
- Optimum user aid can be obtained by factoring out the parts of the information space that are not in a sensible way related the user.
- Factoring out the parts of the information space not sensibly related to a user is not mainly an implementation-time concern. Rather, it largely is a run-time concern. Capability to factor out is required on the fly with respect to newly introduced or modified user types or usage constraints.
- Capability to factor out at run-time information space parts not sensibly related to a user enforces formalization of system-, user-, and usage models.

## RELATED WORK

A few major groups are working on conceptual modelling of WIS. The ARANEUS framework by Atzeni et al. (1998) emphasises that conceptual modelling of WIS should approach a problem triplet consisting of content, navigation and presentation. This leads to modelling databases, hypertext structures and page layout. However, this approach does not focus on the anticipated usage processes and therefore does not fully support information space personalization.

Other authors refer to the ARANEUS framework. Baresi et al. (2000) address the integrated design of hypermedia and operations, thus picking up the functionality aspect, but remain on an informal level. Similarly, Bonifati et al. (2000) present a web modelling language WebML and start discussing personalisation and adaptivity of WIS, but again are very informal. The OOHDM framework in Rossi et al. (2000) emphasises an object layer, hypermedia components, i.e. links, and an interface layer. This is similar to ARANEUS except that OOHDM explicitly refers to an object oriented approach.

Ceri et al. (2002) emphasise a multi-level architecture for the data-driven generation of Web sites. They address adaptivity of Web sites by providing user-dependent site views. The authors (using a different terminology) emphasize views, navigation and presentation. Thus they address the problem triplet targeted by the ARANEUS framework.

The work on the Co-Design Methodology (CDM) for WIS started with the Cottbusnet project addressing the design and development of a regional information service, see Thalheim (1997, 1999) for a detailed description of the project, its approach and the achieved results). This resulted in a methodology oriented at abstraction layers and the co-design of data, views, operations and interfaces, Schewe and Thalheim, (2000). Storyboarding, Feyer et al. (2000), is central to CDM.

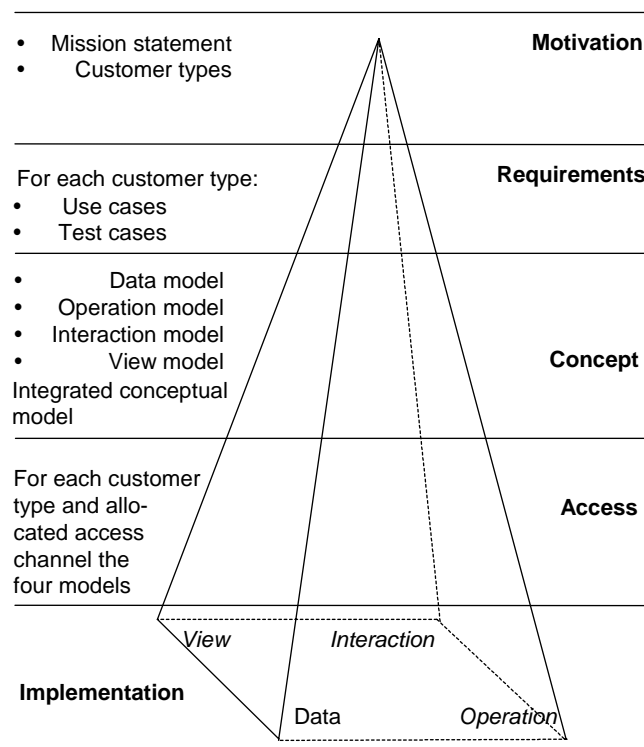
Schewe et al. (2002) and Kaschek et al. (2003a) apply storyboarding to electronic banking. Further issues regarding banking WIS are discussed in Kaschek et al. (2003b-d). In CACM (2004) there are several papers published regarding various aspects of architectures for financial services.

## STORYBOARDING

It is well known that a lot of system development projects do not lead to the desired results. In part this appears to be a consequence of the quality of actual usage processes not being satisfactorily taken into account during systems analysis and design. According to common prejudice software-based systems (such as WIS) are inherently hard to use and design. There is, however, evidence for non-software-based systems being not significantly more usable and better designed. Norman in "The design of everyday things" Norman (2002) presents numerous examples for non-software-based everyday things that continue to cause trouble to their users. Wodtke (2003) and Preece et al. (2002) have, among others, concluded, that the quality of usage processes deserves more attention if the difficulties with developing, using and maintaining software-based systems shall ever be overcome.

Roughly spoken is storyboarding the activity of creating storyboards. In the Web, see for example UN (2003), one finds definitions like: “A storyboard is a low fidelity prototype consisting of a series of screen sketches.” Regarding our more formalized but related understanding of storyboarding see, e.g. Feyer et al. (1998), Kaschek et al. (2004a). Storyboarding initially focuses on user-system interaction and aims at describing intended or typical usage scenarios. It derives from these scenarios the resources, i.e., data / views, operations and dialogues needed by a user for interacting with a WIS. Storyboarding proposes to integrate finally all scenarios into one model, the story space, and reconstruct from it the usage scenarios, i.e., stories started with. Using it we rely on:

- Thalheim’s abstraction layer model ALM, see **Figure 1** and refer e.g. to Kaschek et al. (2004a) for a more detailed discussion. The ALM identifies abstraction layers (motivation, requirements, concept, access, and implementation) and for each abstraction layer sub-models for WIS development
- The CDM Thalheim (1997, 1999), Schewe and Thalheim (2000) recommending that the system parameters data, view, interaction, and operation continuously and concurrently drive the development process.



**Figure 1: Abstraction layers and design driving parameters**

Story boarding presupposes a WIS reference model according to which users have access to an information space that contains information objects with which users can interact and several of which are related to each other by navigation links that the user can choose for traversing the information space. Usage processes of WIS, i.e., stories can be understood as journey in the information space. Such journeys may be modelled as a directed, labelled graph. The labels indicate the individual actions carried out at a graph node, which in story boarding is called a scene, and which can be understood as providing users access to an information object. The operations of the information object may refer to an underlying database and actually store, retrieve or process data. Often the clarity of such graphs is flawed by several of their edges crossing and the number of labels being that large that it is hard to see to which edge or vertex of the graph they are attached. Also the semantics provided by the language of graph theory is limited. Consider for example a situation occurring very frequently in process specifications expressed in graph-based process specification languages: a vertex in a process has an out-degree or an in-degree larger than one (i.e., it has more than one outgoing or more than one incoming edge respectively). Without further elaboration on this structure it is not clear what exactly it means! Temporal, causal, modal or other aspects of relations between vertices could be intended with such a specification. More expressive process specification languages have been investigated. One such language, SiteLang, was introduced in Düsterhöft and Thalheim (2001). See, e.g., Kaschek et al. (2004a) for a further discussion of SiteLang. Its semantics can be defined by the mathematical structure of Kleene algebra. Using the latter for story boarding was introduced in Schewe and Thalheim (2004). We are going to use Kleene algebras in this paper without introducing them formally, because we only want to illustrate their utility for WIS development.

## PERSONALIZING INFORMATION SPACES

Conceptually information spaces can be described as sets of scenes and scene transitions. The latter connect scenes to other scenes and are known as navigation links. At each scene users interact with an information object. Tailoring information spaces systematically and grounded on an algebraic theory requires identification of concerns or viewpoints according to which this tailoring shall be performed. Among the possible angles for tailoring the information space we restrict ourselves to user-related angles since we want to aid users in the use they make of WIS. Applying the well-known Zachman framework, see, e.g. Zachman (1987) we propose consideration of:

- User action, i.e., what is the user currently doing and what did s/he do beforehand? This refers to data accessed and operations triggered as well as to usage context, discussed for example in Kaschek et al. (2003b) and Binemann-Zdanovicz et al. (2004) respectively.
- User intentions, i.e., why does the user interact with the WIS?
- User-system interaction time, i.e., when does the user interact with the WIS?
- User preferences, i.e., how does the user interact with the WIS and how would s/he like to do it?
- User location, i.e., where does the user interact with the WIS?
- User characteristics', i.e., who is the user and what characteristics does s/he have? This refers to user typing, discussed for example in Kaschek et al. (2004b) as well as to the devices and access channels used for system interaction.

Presupposing that the WIS is going to be used in the well-known “point-and-click” style, it is obvious that from a conceptual point of view the user complexity of a WIS is mainly determined by two things: firstly, the quality of decomposition of overall WIS-functionality into clickable items, and secondly, the total number of these clickable items per scene. For now we abstract from the quality of the decomposition chosen and focus on the number of clickable items accessible at scenes. However, we come back to this issue below when we analyse the example-WIS. The set of clickable items among which a user may choose during WIS interacting must be limited if providing optimum aid is intended. Story boarding, as was explained above, inherently leads to good results. Storyboards can be expressed in the language SiteLang. Schewe and Thalheim (2004) have shown that SiteLang’s modelling notions can be formally defined with help of Kleene algebra and that formal analysis of algebraic expressions can be used for identification of the clickable items that shall be accessible to a particular user. For the latter we introduce the state of an information space journey. This state can be used for deciding which clickable items to offer next. We are going to use predicates for this purpose that are associated with the interrogatives used in the above item list.

For identifying how these predicates could be associated with users, we model users as instances of tuple types and ensure that with the respective tuple type all data, that from a business perspective are considered as important, can be represented. This then gives rise to a number of dimensions for each of which scale values are available that enable discriminating between users in each of these dimensions. Users therefore are modelled as a point in a multi-dimensional space, i.e., the user space. Several strategies for maintaining the state of story space traversals can be identified:

- History-minded strategies
  - Aggregating, i.e., the new state of a story space journey is the conjunction of the current state and a number of predicates.
  - Dynamic, i.e., the new state of a story space journey is a function of the current state and the item clicked most recently.
  - Static, i.e., the state of a story space journey is defined at user-system interaction start and is never updated.
- Rationale-minded strategies
  - Oblique, i.e., the state of a story space journey is defined in terms of subsets of scales of a number of user characteristics.
  - Straight, i.e. the state of a story space journey is defined in terms of a set of user models, i.e., points in the user space.

Clearly, the static strategy does not yield minimum user complexity of the WIS if the needed functionality significantly differs over user-system interaction time. Furthermore, the aggregating strategy leads to increasingly limited sets of clickable items being accessible to users. Thus this strategy often will result in

simplistic stories. Therefore the only history-minded strategy that appears promising is the dynamic strategy. Among the rationale-minded strategies the oblique strategy seems to better fit the dynamic strategy. The advantage of the oblique strategy is that it focuses on a number of independent user characteristics and that combined with the dynamic strategy simplifies maintaining and analysing the story space traversal state.

Consider for example a banking WIS. The scale values for user dimensions *occupation* and *account balance* might be “student”, “worker”, “academic”, “unemployed”, and “retired” as well as “below 1,000”, “1,000 – 9,999”, “10,000 – 99,999”, “100,000 – 1,000,000”, and “above 1,000,000” respectively. While a bank might have a number of customers who are students with account balances above \$1,000,000 it is not reasonable to expect that the aid a bank would wish to provide to these individuals actually depends on both scale values, i.e., on an individual being wealthy and at the same time being a student. Rather the aid that a bank would want to provide would depend on only one of these scale values. This example seems to suggest that for practical WIS the oblique strategy combined with the dynamic strategy for maintaining the story space traversal state would be chosen.

Producing systems that are capable of adapting to system users must be expected to imply maintenance issues. It therefore appears that this capability should be based on the reflection pattern Buschmann et al. (1999) or similar software patterns. The reflection pattern recommends the introduction and exploitation of meta information for realising adaptability. The pattern essentially works such that a system meta repository is introduced. This repository in our case needs to contain a system meta model, a user meta model, an event or command meta model, and references to the functions that can carry out the transformations needed. A respective system would operate such that the event- or command stream is analysed and in case adaptation need is recognised according to the user meta model and the system meta model, the transforming procedure retrieved and executed. We do not discuss such architectural issues in more detail. We rather use in the following example the kind of data that could be used for detecting the need for transformation and for specifying the transformation functions meeting the need.

## EXAMPLE

The paper Binemann-Zdanovicz et al. (2004) discusses a relatively small storyboard of a banking WIS the complexity of which, however, is remarkable. In this section we focus on a part of such a WIS and show how it can be tailored to minimize user complexity. The WIS we are investigating is the UBS AG WIS. Parts of an earlier analysis of this WIS are published in Kaschek et al. (2003a). We expect the WIS of other major banks to be different but not to be of lower user complexity. One can access the WIS from <http://www.ubs.com/e/>. We are using a slightly modified terminology for sake of abbreviation. Note that already at the entry level of the information space a number of options is available for tailoring the system to its user: A multi-language facility is in place that allows users to choose the language of interaction. English, German, French and Italian are available. This capability clearly addresses user preferences. The identity of the user is addressed by a choice offered for client login. Potential user interests are addressed by a number of options. These are: the company, news about it, the business group, business with individuals, business with corporations and institutions, and market information. Furthermore, the choices are offered to interact with the investment-banking, funds, and UBS-quotes sub-WIS as well as location finder, service finder and contact sub-WIS.

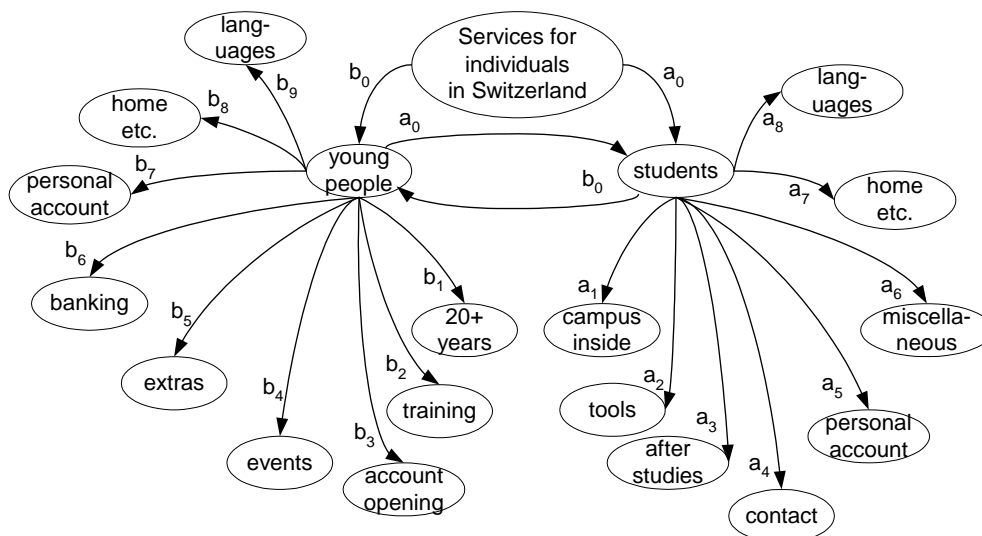


Figure 2: Simplified UBS "young people" and "students" process

Invoking the UBS service finder for individuals leads to choices being available to the user regarding where he wants to do business. For having an impression of the variety of banking services available Switzerland is chosen because in the UBS home market one can expect all services to be available. With respect to Switzerland UBS distinguishes between three types of clients: all individuals, students and young people, and wealth management clients. In Figure 2 we display in simplified and non-exhaustive form the process that is specified by the UBS WIS for students and young people. In this diagram the ovals represent scenes and the arrows between scenes represent scene transitions. The arrow labels  $a_i$  and  $b_j$  (for  $0 \leq i \leq 8$ , and  $0 \leq j \leq 9$ ) represent activities that trigger the transition indicated by the arrow to which the labels are attached. If activity  $a_i$ ,  $i > 0$  triggers a transition to scene  $S(a,i)$  then the activities possible at this scene are not represented in the diagram to keep it clear. However, it is presupposed that at this particular scene activities  $a_{i,j}$  are enabled with  $1 \leq j \leq m_i$ . We apply the same convention for the scenes  $S(b,i)$  at which one arrives after performing activity  $b_i$ , for  $i > 0$ .

Presupposing that processes if represented as Kleene algebra are composed out of activities and using the convention that “+” represents choice of activities, that “;” represents sequence of activities (and usually is simply omitted) and that “\*” represents iteration of activities the process in the figure can be represented as the algebraic expression:

$$b_0(a_0(\sum_{i=1, \dots, 8} a_i)^* + b_0) + \sum_{i=1, \dots, 9} b_i(\sum_{j=1, \dots, m_i} b_{i,j})^* + a_0(b_0(\sum_{i=1, \dots, 9} b_i)^* + a_0) + \sum_{i=1, \dots, 8} a_i(\sum_{j=1, \dots, n_i} a_{i,j})^* )^*$$

One can now project this process on the type “young people” and on the type “student”, i.e., remove all in it that is dedicated to students and young people respectively. Then one gets the processes:

- $b_0(\sum_{i=1, \dots, 9} b_i(\sum_{j=1, \dots, m_i} b_{i,j})^*)^*$ ,
- $a_0(\sum_{i=1, \dots, 8} a_i(\sum_{j=1, \dots, n_i} a_{i,j})^*)^*$ , respectively.

Obviously this is exactly what one gets if one decomposes **Figure 1** into the parts dedicated to young people and students respectively. Note that to each object in the information space there is a number of labels attached indicating the user types entitled for interaction with the information object associated to the scene. These labels can be used to automatically obtain the projections mentioned. First the scenes are determined that are supposed to make up a story. Secondly, based on a partial mapping  $\sigma$  associating activities to scenes and being part of a Kleene algebra, the activities can be added that shall be accessible to a user in this story. The mathematical concept used here (behind the scenes) is that of the generated sub algebra. This sub algebra is specified as the intersection of all sub algebras containing the given subset of the algebra support, in our case the information space. In this way minimum user complexity is guaranteed. Reuse of usage processes to some extent can be supported by using co-retracts of Kleene algebras. For more detail regarding co-retracts of algebras see, e.g. Kaschek (2004).

The example might look too simple. While its simplicity is one of the reasons for it being chosen one should note that the smart design used by the UBS developers has significantly reduced process complexity. The design strategy chosen can be summarized as:

- For each user type there is a “hall” scene defined from which several “chamber” scenes are accessible and the hall scene displays to the user always the index to the chamber scenes as well as detail regarding the chamber being selected by the user.
- There are no navigation links between chamber scenes or from chamber scenes to hall scenes.
- If more complex functionality is required in one of the chambers than can be handled within the design as explained so far then it leads to a new browser window being opened and the old one preserving the system state prior to opening the new window. We abstracted from this aspect of the design in our example.

Obviously the chosen design strategy implies a particularly easy integration of user type specific stories and to low complexity of the integrated story as well as the stories that need to be integrated.

The example can be extended for illustration of slightly more advanced concepts. For explaining this extension we recall the concept of precondition and post-condition of activities. These are logical formulae that guard the activity with which they are associated. In case of the precondition this means that the activity after being triggered only is carried out if the precondition is true. The post-condition, however, is supposed to be true once the activity has been carried out. Suppose that with respect to the scenes in Figure 2 one wants to have a situation in place where the scenes “personal account” in the sub-diagram for young people and for students are derived from a more complete scene respectively. This kind of problem requires the source of the transition being deducible from the transition being carried out. This can be achieved by using post-conditions. Suppose that conditions are abbreviated by an integer indexed letter  $c$ , i.e.,  $c_1, c_2, \dots$ . Let  $c_1 \equiv$  “the user wants to access

account details” and  $c_2 \equiv$  “the source scene is the young people scene” as well as  $c_3 \equiv$  “the source scene is the student scene”. The transition from “young people” and from “student” to personal account can thus be described as  $c_1b_7c_2$  and  $c_1a_5c_3$  respectively. Note that in this transition representation the activity (i.e.,  $b_7$  and  $a_5$  respectively) is prefixed by the precondition and postfixed by the post-condition.

The accounts scene implemented by the WIS under study offers access to 12 different kinds of accounts. Denote the activity to invoke these with  $x_1, x_2, \dots, x_{12}$ . Furthermore there is a toolbox offered on the accounts scene with a finance calculator, a mortgage calculator, a price calculator, a budget calculator, and account tools. Denote the activities to invoke any of these  $t_1, t_2, t_3, t_4$ , and  $t_5$ . None of these are accessible to users coming from the “young people” scene. Similarly, for those users only the “generation personal account” scene is accessible at the “accounts” scene. The situation is a bit different with respect to students. Students may access the finance calculator and the budget calculator. Furthermore students may access the “campus personal account” scene.<sup>1</sup> If now each of the account activities  $x_1, x_2, \dots, x_{12}$  except the ones leading to the “generation personal account” scene or the “campus personal account” scene are guarded by the post condition  $\neg c_2 \vee \neg c_3$ , then this post condition is true if the user does not arrive via the “young people” or the “student” scene. If s/he, however, arrives from one of these scenes then the preconditions are false and the respective links are disabled, i.e., the scenes are not accessible. Clearly, in a similar way the tool links  $t_i$  can be enabled or disabled.

## CONCLUSION

We have discussed storyboards in current banking and some strategic aspects of information space personalization. We furthermore suggested by an example that a WIS usage processes can be specified with a Kleene algebra. We then illustrated that user type specific stories can be derived automatically from stories integrating usage processes for various user types. We also have argued that the levels of reuse and system appearance consistency can be increased if Kleene algebras are used. Together this suggests that Kleene algebra indeed is a suitable tool for specifying WIS usage processes and thus provide a suitable language for design and maintenance of WIS. Future work regarding Kleene algebras will aim at creating a tool for generating WIS out of storyboards. Furthermore algorithms will be worked out for the automatic translation of WIS (specified according to the meta models discussed in the related work section) into Kleene algebras and vice versa. The reusability of WIS would be significantly increased by the availability of such algorithms for the main approaches to WIS development since Kleene algebra would be established as a lingua franca of WIS.

## COPYRIGHT

Roland Kaschek, Klaus-Dieter Schewe, Catherine Wallace, and Claire Matthews © 2004. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.

## REFERENCES

- Atzeni, P., Gupta, A. and Sarawagi S. (1998). Design and maintenance of data-intensive Web sites. In *proceedings of EDBT'98*.
- Banker, R., Datar, S., Kemerer, C. and Zweig D. (1993). Software complexity and maintenance costs. *CACM Vol. 36, no. 8; 1993*; pp. 81 – 93.
- Baresi, L., Garzotto, F., and Paolini, P. (2000). From web sites to web applications: New issues for conceptual modeling. *ER Workshops 2000*.
- Binemann-Zdanovicz, A., Kaschek, R., Schewe K.-D., and Thalheim, B. (2004). Context-aware Web information systems. In *proceedings of APCCM 2004*.
- Bonifati, A., Ceri, S., Fraternali, P., and Maurino A. (2000). Building multi-device, content-centric applications using WebML and the W3I3 tool suite. *ER Workshops 2000*.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. (1999). *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons; 1999.

---

<sup>1</sup> We simplify again for illustration purposes: both “students” and “young people” additionally may access the “personal account scene”. Furthermore “young people” may access the “campus personal account” scene. However, this access is via an additional link-list.



- CACM 2004 *Communications of the ACM*, Vol. 47, no 5, May 2004.
- Ceri, S., Fraternali, P., and Matera, M. (2002). Conceptual modeling of data-intensive web applications. *IEEE Internet Computing*, 6(4):20-30.
- Düsterhöft, A., Thalheim, B. (2001). SiteLang: conceptual modeling of Internet sites. In *proceedings of ER 2001*, pp. 179 – 192.
- Feyer, T., Kao, O., Schewe, K.-D., and Thalheim, B. (2000). Design of data-intensive Web-based information systems. In *proceedings of WISE 2000*, pp. 462 – 467.
- Hirschheim, R., Klein, H., and Lyytinen, K. (1995). *Information systems development and data modelling: conceptual and philosophical foundations*. Cambridge University Press. 1995.
- Kaschek R. A characterization of co-retracts of functional structures. To be published in *Post-Conference Proceedings of NWECS 2003*, World Scientific Publishers, 2004.
- Kaschek, R., Matthews, C., Schewe, K.-D., and Wallace, C. (2003a). Analysing Web information systems with the Abstraction Layer Model and SiteLang. In *proceedings of ACIS 2003*.
- Kaschek R., Schewe K.-D., Thalheim, B., and Zhang, L. (2003b). Integrating Context in Modelling for Web Information Systems. In *proceedings of WES 2003*.
- Kaschek, R., Matthews, C., Wallace, C., and Schewe K.-D. (2300c). Increasing Automation in Lending. In *proceedings of iiWAS 2003*.
- Kaschek, R., Wallace, C., Matthews, C., Künsch, P. (2003d). e-Mortgages: NZ State of the Art and Perspectives. *Invited paper at CSI 2003*.
- Kaschek, R., Schewe, K.-D., Wallace, C., and Matthews, C. (2004a). Story boarding for Web-based information systems. Chapter 1 of Taniar D. and Rahayu W. (Eds.). *Web Information Systems*. Idea Group Publishing; 2004.
- Kaschek, R., Schewe, K.-D., Thalheim, B., Kuss ,Th., Tschiedel, B. (2004b). Learner Typing for Electronic Learning Systems. In *proceedings of ICALT 2004*.
- Norman, D. (2002). *The design of everyday things*. Basic Books; 2<sup>nd</sup>. ed. 2002.
- Preece, J., Rogers, Y., and Sharp, H. (2002). *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2002.
- Rossi, G., Garrido, A., and Schwabe, D. (2000). Navigating between objects: Lessons from an object-oriented framework perspective. *ACM Computing Surveys*, 32(1).
- Schewe, K.-D., and Thalheim, B. (2000). Conceptual modelling of internet sites. <http://infosys.massey.ac.nz/~kdschewe/pub/slides/ER00tuti.ps> ( $i = 0, \dots, 6$ ). *Tutorial Notes at the 19th International Conference on Conceptual Modelling (ER 2000)*.
- Schewe, K.-D., Kaschek, R., Matthews, C., and Wallace, C. (2002). Modelling web-based banking systems: story boarding and user profiling. In *proceedings of eCoMo 2002*.
- Schewe, K.-D., and Thalheim, B., (2004). Reasoning about Web information systems using story algebras. In *proceedings of ADBIS 2004*.
- Thalheim, B. (1997). Development of database-backed information services for CottbusNet. *Technical Report CS-20-97, BTU Cottbus*.
- Thalheim B. (1999). Codesign of structure, behaviour and interaction for database applications with the abstraction layer model. In *proceedings of EMISA 1999*; pp. 229 – 249.
- Tretiakov, A., and Kaschek, R. (2004). An adaptive metaphor layer for e-learning: conceptual model and architecture. In *proceedings of ICCE 2004*.
- UN (2003). Usability Net 2003. Accessible via: <http://www.usabilitynet.org/tools/storyboarding.htm>. Accessed at 18 October 2004.
- Wodtke, C. (2003). *Information architecture: blueprints for the Web*. New Riders, 2003.
- Zachman, J. (1987). A framework for information systems architecture. *IBM Systems Journal*, Vol. 26, no. 3; 1987, pp. 276 – 292.