Hypermedia Systems Modelling Framework

Peter Dolog and Mária Bieliková
Department of Computer Science and Engineering
Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava, Slovakia
e-mail: dolog@dcs.elf.stuba.sk, bielik@elf.stuba.sk

This work was partially supported by Slovak Science Grant Agency, grant No. G1/7611/20.

Abstract

Modelling is an important activity in the development of complex systems. Hypermedia modelling is a relatively new direction of research. Based on achieved results in software systems development, the recognition of importance of hypermedia modelling took relatively short time from the time of widespread adoption of hypermedia systems. However, the hypermedia modelling is not that mature as its software counterpart.

The aim of this paper is to present a survey of hypermedia systems modelling along proposed five perspectives: hypermedia application perspective, development process perspective, aspect (static or dynamic) perspective, degree of formality, and notation perspective. First, a categorisation of hypermedia related models is proposed. According to this categorisation, the multidimensional modelling space is established. The space leads to a schema that is capable to support decisions about the reuse of methods and techniques for hypermedia systems modelling. Additionally, selected hypermedia modelling methods and techniques with their mutual dependencies are studied according to the proposed categorisation.

Keywords: hypermedia system, modelling perspective, application domain model, navigation model, presentation model.

1 Introduction

Hypermedia applications become very popular for providing, accessing, retrieving and manipulating information. Their popularity have been increased by growth of the Internet, which provided opportunities for deploying more complex hypermedia systems. The increasing complexity of hypermedia systems raised the need to employ modelling in hypermedia development process.

Models of a hypermedia and methods to its modelling are extensively studied only in past two decades. Several methods, modelling techniques and languages were developed, or adopted from especially software (or hardware) modelling. However, hypermedia applications differ from their software counterparts.

The significant feature of the hypermedia application is a possibility of nonsequential navigation between pieces of information or different documents, i.e. there is no single order that determines a sequence in which the text is read. Hypermedia applications provide rather exploratory user interfaces, which are often adapted to different audience or different devices according to anywhere, anytime and anybody paradigm. Hypermedia applications for the Web are developed for multiple platforms with several accessing approaches to information bases.

Information content constitutes not only regularly structured data but often unstructured (multimedia) items in the hypermedia application. This caused that new roles in the development process are needed: such as authors, layout designers, or multimedia experts.

Hypermedia development has to take into account also aesthetic and cognitive aspects as well, that traditional software engineering environments do not support [34].

Accordingly, the modelling methods and techniques known from software or hardware modelling have to be extended and adapted to support hypermedia modelling. The aim of this paper is to survey existing approaches to hypermedia modelling. We proposed a categorisation of hypermedia systems models. According to this categorisation, the multidimensional modelling space is established. The space leads to a schema that is capable to support decisions about the reuse of methods and techniques for hypermedia systems modelling.

The rest of the paper is as follows. In section 2 we propose perspectives, which constitute multidimensional modelling space. Established perspectives follow essential principles of systems modelling. We provide deeper insight into hypermedia application modelling perspective, which introduces hypermedia specific features to the systems modelling. Considering established perspectives, we survey application domain modelling (section 3.1), navigation modelling (section 3.2), and presentation modelling (section 3.3). In section 4 possible usage of taxonomies is discussed. Paper concludes with summary and possible directions in this research.

2 Modelling perspectives

Hypermedia system is usually understood as an authoring tool in the literature. On the other hand, hypermedia application is understood as an application of the hypermedia system in particular domain; i.e. the system provides information from that application domain and provides adequate user interface, and information exploring and retrieving functions for a user from that domain. However, both are systems – the former is *empty* and the later is filled with information related to particular domain.

It is widely accepted that models constitute important mechanism in the process of a system development. Models help us understand the system by omitting some details. The choice of *what* to model has a significant effect on understanding a problem and suggesting the solution. Significant influence on the solution has also the matter *when* particular model is created. According to answers on "what" and "when" questions, particular notation is chosen. Notation determines the level of formality of particular model. Notation is also determined by a decision about whether structural or behavioural model is developed.

According to mentioned questions we proposed a framework for systems modelling. The framework distinguishes five perspectives of the hypermedia system modelling. All perspectives are related one to another. Each represents rather different (orthogonal) point of view than different independent approach:

- *hypermedia application (tiers) perspective* is concerned with the question "what" to model. Tiers, which are commonly discussed in the hypermedia perspective are application domain, navigation and presentation;

- *development process perspective* follows the question "when". It reflects the fact that a modelling technique is applied in particular phase in development of a system (i.e., requirements, analysis, design, or implementation);
- *aspect perspective* is concerned with two commonly recognised aspects of models: structural and behavioural. Structural models describe structure of a system (e.g., subsystems, packages) and structural relationships between them (e.g., inheritance, dependency and associations). Behavioural models describe reactions of a system to external or internal events such as collaborations or message passing in the system or the system's reactions to user interactions.
- *degree of formality perspective* reflects how formal is a technique applied in modelling of a system (i.e., formal, semiformal, or informal);
- *technique and notation perspective* emphasises on modelling technique and corresponding notation used.

Mentioned perspectives or their categories can be refined. For example, behavioural techniques can be divided according to semantics of a technique. There are three types of semantics: *axiomatic*, *denotational*, and *operational* [26]. Axiomatic semantics map programs directly to properties, which characterise their behaviours. Denotational semantics map programs onto the functions, from which input-output behaviour can be derived. Operational semantics allow that behaviour can be derived from the sequence of transitions a program may perform. Formal models can be classified according to formal model or formal theory used as a base for modelling language. The basic categories are *set theory*, *logic*, *algebras*, *graphs* and *automatons*.

To provide modelling framework for some of mentioned perspectives is appearent from the beginning of research in the field of hypermedia. The first general-purpose hypermedia models, which appeared in the late eighties and in the early nineties can be taken as a foundation for general modelling frameworks (e.g., Dexter model [19], Amsterdam Hypermedia Model [20] or Dortmund Family of Hypermedia Models [45]). Their levels form similar tiers or views as we listed according the hypermedia application perspective.

3 Hypermedia application modelling perspective

Proposed modelling framework is applicable for various systems. Models differ in the perspective concerned with characteristics of solution (in our case hypermedia) environment, i.e., "what to model" question is resolved (hypermedia application perspective for hypermedia). Because our primary intention is to support hypermedia systems modelling, we concentrate in this paper on the hypermedia application perspective.

Hypermedia application modelling perspective follows two categories recognised by Garzotto et al.: *application domain models* and *system models* [16]. The models of hypermedia integrate two features. They determine (i) language elements for structuring the hypermedia information space, and (ii) the architecture of hypermedia applications built upon it.

Nowadays, system models are understood in wider context. It seems to be better to look at a system model as a set of specific models, which are based on different views of an application. Modelling languages designated for description of software systems such as entity-relationship diagrams, Object Modelling Technique language or Unified Modelling Language (UML) were taken and customised for these purposes. Especially, the UML explicitly distinguishes between several views of a modelled system. Views, which are essential in hypermedia application modelling are [9, 14]:

- *application domain:* incorporates concepts with their mutual relationships in particular domain. Application domain model determines, which information will be provided by hypermedia application;
- *navigation:* defines a way, how application will navigate a user in the information net. Navigation model groups information chunks into contexts and interconnects them by links;
- *presentation:* is concerned with appearance of information chunks to a user. Presentation model defines spatial layout and content of information chunks related to the user interface. It also defines presentation classes or objects, spatial relationships between them and content associated with them.

3.1 Application domain model

Application domain model comprises abstract concepts, which are provided as information in a hypermedia system. Moreover, the application domain model could serve as a model for indexing a content. The content can be considered as an instance of particular concepts, or as a set of structural features of a concept. Application domain model can also involve typical models of tasks, which may serve as a base for later navigation and predefined interaction between a user and the target system. Application domain models serve as a base for data model of the software application.

Application domain modelling as a component of application domain engineering is discussed in [8]. Simplified view of the application domain model is that it is a consistent set of requirements for the system in the particular application domain[1], and the concepts and their relationships. This holds in hypermedia engineering too.

Concepts' static structure of an application domain is mostly modelled by two approaches: (i) application of semiformal techniques such as class diagrams or entity-relationships diagrams, or (ii) application of graph formalism for semantic nets. Semiformal techniques are based on defining concepts as classes or objects with their attributes (features). Concepts are interconnected by several types of relationships (association, subtyping or generalisation/specialisation and containment).

The Relationship Management Methodology (RMM) involves entity-relationship diagrams (ERD) without modification [24]. Hypermedia Design Method (HDM) [16] employs customised ERD. Hypermedia application often involves unstructured information. HDM introduces entities, which derive its content from its components. Components can be for example sections of documents. Entities represent objects in the real word (exemplar perspective of a concept). Entities can be grouped into entity types. The distinction between entities in classical ERD and entity types in HDM is that HDM entity types have more complex inner structure formed by entities, their components and perspectives. Web Modelling Language (WebML) [5] is also based on ERD. The generator from WebML to Microsoft Active Server Pages templates was implemented.

[1] Czarnecki refers to them as feature models.

Methods such as Object-Oriented HDM (OOHDM) [39], UML based Web Engineering (UWE) [21], W2000 [2] (successor of HDM)[2], or Domain Engineering Approach to Hypermedia Engineering (DEAHE) [13] employ class diagrams. DEAHE introduces feature modelling for hypermedia application domain engineering. The feature model is a set of models, which describes features of concepts from conceptual model and their depandencies. The features can be optional or mandatory. The features can stand as mutually exclusive variants (XOR features), mutually required features (AND features), and features whose optional number can be chosen (OR features). This modelling technique was considered for reuse of information modelled by conceptual and feature models in several hypermedia applications, which belongs to one application family.

WebComposition Markup Language (WCML) [17] is based on component oriented modelling. Main element in the WCML language is a component with properties and declarations of behavioural features. The components can inherit from other components or aggregate other components. Special property for the component content is introduced too. WCML model can be compiled into target environments [15]. The mappings from OOHDM designs to WCML were introduced [40].

Summary of notations and primitives for static structure modelling using semiformal techniques is listed in Table 1.

Method/Model	Notation	Primitives
OOHDM	Class Diagram	Class, Association, Aggregation, Generalisation/Specialisation
SOHDM	CRC, Scenario Diagram	CRC Card, Relationship, Task, Flow
RMM	ERD	Entity, Relationship
HDM	Customised ERD	Entity, Component, Perspective, Collection, Relationship
W2000	Class Diagram, State Diagram	Class, Association, Aggregation, Generalisation/Specialisation, State, Transition
HDM-lite	Customised ERD	Entity, Component, Perspective, Collection, Relationship
UWE	Class Diagram	Class, Association, Aggregation, Generalisation/Specialisation
WebML	Customised ERD	Entity, Relationship
WSDM	Class diagram	Class, Association, Aggregation, Generalisation/Specialisation
WCML	Component Diagram	Component, Relationship, Aggregation, Inheritance
DEAHE	Customized Class Diagram (Conceptual and Feature Model)	Concept (class), (OR, XOR, AND) Feature, Variation Point, Relationship, Dependency, Assosiation, Generalisation/Specialisation, Aggregation

Table 1: Summary of techniques for application domain modelling.

One of the first approaches to modelling a hypertext application – the NodeCards system – is graph based. NodeCards system developed in Xerox PARC [18] allows to model application domain by so called NodeCards. The notion of file boxes was introduced. The file box is intended to be used as a container for node cards. Each NodeCard in the system had to be

[2]In W2000 the hyperbase information design is counterpart of conceptual modelling.

associated to at least one file box. The file boxes can contain other file boxes. Information is embedded in the NodeCard. NodeCards together with file boxes form directed acyclic graph. Another graph based model with domain semantics is proposed in [46]. It is also discussed as general-purpose hypermedia model in [10] because it comprises also navigation aspect. The domain is modelled as a graph with associated domain labels for nodes and links. The grouping mechanism is provided through presents of grouping nodes. Nested Context Model (NCM) [41] also models structure of document as a tree.

Behavioural models of an application domain mostly express activities. They are represented by activity diagrams or flow charts. These models describe processes, activities or scenarios in particular domain. The navigation can be derived from these models, or they can serve as organizing framework for incorporation of hypermedia application into an organization.

Scenario-Based Object-Oriented Hypermedia Design Methodology (SOHDM) [29] employs scenario diagrams (similar to activity graphs in the UML) for task and activity specification. In [10] the UML activity graphs are used in domain modelling for expressing activities and processes. They serve as a base for navigation in information about tasks and processes from application domain in hypermedia application.

3.2 Navigation model

Navigation models are built upon application domain models. They describe possible navigations through information space determined by the application domain model. The navigation in hypermedia application is considered at two levels: *local* and *global*. This categorisation reflects the principles identified for example in [44, 28] where the importance of large local (currently presented information) and global (all information) coherence and low cognitive overhead with additional tasks performed at the user interface is stressed.

Two aspects are commonly employed in modelling of the navigation. The former is based on structuring information into contexts, which represent coherent information chunks interconnected by meaningful links — static sructure of navigation. The latter is based on operational semantics, i.e. we explore to which state the system will get when particular event will raise the transition — behavioural aspect of navigation (interactions).

Navigation can be viewed from two different user's points of view: browsing and querying [23]. Technically they are equivalent. The difference consists only in who specifies query over a hyperbase. Browsing semantics are mostly specified by a designer of hypermedia application. The query is specified mostly by a user or by a designer when he wants to verify a navigation model.

Structural techniques for specifying navigation are employed for example in OOHDM [39]. The navigation is in OOHDM specified by two different views: navigation class schema and navigation context schema. In the former, the object oriented view mechanism is employed [1]. The navigation classes are views over structural models of an application domain model. They also incorporate derived attributes and attributes, which do not appertain to the application domain model. Navigation context schema is intended to group navigation classes into contexts and access structures such as indexes, guided tours and so on. Similar principle for specifying navigation is employed in UWE [21] and WebML [5]. There is only little difference in denotation of elements and modelling language employed. WebML in addition introduces web specific modelling elements. Object-oriented views are also used in SOHDM [29]. The contexts are specified by access structure nodes, which are interconnected by links. RMM [24] based methodologies use slices for denotation of contexts where entities and their attributes are grouped into these slices. HDM [16] provides perspectives for its components and entities. It means that the entity can have different contents in different contexts (perspectives).

Behavioural specification of navigation is based especially on state diagrams and their equivalents, such as petri nets. The notion of path is employed. Stotts and Furuta [42] proposed a model (called Trellis model) for specification of *browsing semantics* where binary petri nets were used. This model was later extended with coloured petri nets to support collaboration specification [43] and context adaptation [33]. Hypermedia related extension of Harel's state machines was proposed in XHMBS (eXtended Hyperdocument Model Based on Statecharts) [36]. This approach integrates structural features of state diagrams (AND and OR composition of states) and behavioural features (events and transitions). The path can be reconstructed at several levels according to composition tree of state machine. Browsing semantics is specified similarly in [11] where interaction diagrams and UML state diagrams are used with a little extension to support event associated with request for content. The integration with user modelling by class diagrams to support adaptation modelling was proposed later in [12]. This approach was integrated into DEAHE [13], where mapping between feature model of application domain and state diagram was also introduced.

Querying is sometimes denoted as model checking. Queries specify properties, which a model have to satisfy. From a user point of view, the properties represent information or its types, which the user is looking for. From a designer point of view, queries are properties, which model have to satisfy (primarily according to requirements specification). In both cases a query language is needed, which is mostly determined by underlying state machine based model. The query represents the properties of a trail, which is searched in a model. Such approach is discussed for example in [43] (model verification) and in [30] (querying), later extended with methods for probability of transition raising [31].

Summary of mentioned techniques for navigation modelling is given in Table 2.

Method/Model	Notation	Primitives
OOHDM	Class Diagram	View, Navigation Class, Navigation Context, Link, Index, Guided Tour
SOHDM	Class Diagram, Graph	Object Oriented View, Access Structure Node, Link
RMM	M-Slices	Slices, Links
HDM	Graph	Node, Structural Link, Application Link, Perspective Link, Collection, Component
W2000	Traversals, Collections	Collection, Traversal
HDM-lite	Traversals, Collections	Collection, Traversal
UWE	Class Diagram	Navigation Class, Index, Guided Tour, Link
WebML	Compositions, Link models	Unit, Page, Site View, Link, Selector
WSDM	Component Diagram	Component, Link, Navigation Track
Trellis	Petri Net	Place, Transition
XHMBS/M	Extended State Diagram, Class Diagram	State, Transition, Navigation Class, Navigation Context, Link, Index, Guided Tour
DEAHE	UML State Diagram, Class Diagram	State, Transition, Guard, Event, Action Navigation Class, Navigation Context, Link, Index, Guided Tour

Table 2: Summary of techniques for navigation modelling.

3.3 Presentation model

Presentation model incorporates visual characteristics of information being presented. This view comprises layout configuration of information and its appearance. Beside structural characteristics represented mostly by class diagrams or their equivalents, there are dynamical characteristics such as synchronisation of layout configuration after navigation.

OOHDM [39] uses Advanced Data Views (ADV) for presentation modelling. Similarly to navigation views, ADVs allow to derive attributes from several navigation and/or domain classes. For this purposes the configuration diagrams are used [6]. The attributes for expressing visual characteristics can extend the specification of such object. The behaviour is specified by a message relationship to external ADV. ADVCharts are used for expressing possible states of specific presentation objects (such as buttons) with definition of a message, which is sent to another object when the state is changed. However, state-based modelling is used in OOHDM only for ADV behaviour modelling. HDM-lite, as a successor HDM, brings similar idea — structuring the user interface. The presentation model is represented by three basic types: *component page-types*, *collection pages*, *traversal page-types* [14]. The UML class diagram is employed in [21] with stereotypes for presentation and navigation. The UML class is extended by frame sets as counterpart to HTML frame sets and presentation classes. The state charts are used only for specific objects similarly to OOHDM. WebML [5] incorporates presentation modelling too. The models of presentation are represented by two types of style sheets: *untyped style sheets—models* and *typed style sheets*. The former are generic and independent from the content. The latter are intended for specific concepts (information items).

The refinement of navigation model based on the UML state diagram is used in [12][3]. State diagrams of navigation are extended with special presentation states. Dynamics and adaptation are expressed by guard/action/event typing of transition. The guards, actions and events can reference user model features and according to them adapt presentation. The graphic appearance of information chunks is modelled by *presentation classes*. They are mapped to particular states.

Traditional Dexter-based reference models (Dexter hypertext model [19], Amsterdam hypermedia model, Adaptive hypermedia application model [4]) incorporate the runtime layer where presentation of a component is seen as an instance of the component. Two fundamental constructs are introduced—instance and session. Instance is used as a mechanism for instantiation and mapping components and links to anchors. Session is used to map the instance to corresponding component.

RMM [24], HDM [16], or W2000 [2] consider HTML as presentation modelling language. The abstraction of HTML can be found in visual modelling language — World Wide Web Design Technique (W3DT) [3]. The approach is based on top-down principle of large scale web site decomposition. The W3DT modelling language incorporates elements for modelling web site, which may contain one or more diagrams. It means that the site is modelled by several diagrams (directory structure, linkage of pages, etc.). The diagrams are structured into pages, which can have optionally one or more layouts. Pages (form, index or menu) may be linked to other pages. The authors distinguish static and dynamic elements. Jim Conallen provides stereotypes for modelling web application architectures using UML [7]. His approach is based on the architectural and programming elements supported by web environment. Architecture model depends on whether the site is dynamic or static. Several stereotypes for relationships are provided such as link, target, builds, etc.

[3] Later incorporated into DEAHE [13]

Method/Model	Notation	Primitives
OOHDM	Abstract Data Views, ADVCharts	View, Object, State, Transition
SOHDM	User Interface design	UI objects (button, image, list...)
HDM	HMTL	slot, frame, generated HTML
W2000	HTML	Page Types, HTML elements
HDM-lite	HMTL	Page Types, HTML elements
UWE	Class diagram	Presentation class, Framesets
WebML	Pages, Style Sheets	Entity, Relationship
W3DT	Simplified HTML Metamodel	Pages, Diagrams, Forms, Layouts
DEAHE	Refined State Diagram	State, Transition, Guard, Event, Action

Table 3: Summary of techniques for presentation modelling.

Summary of mentioned techniques for presentation modelling is given in Table 3.

3.4 Dependencies between models

The modelling techniques described in previous subsections are used in the models, which are common for any hypermedia application. We denote them as *core hypermedia applications models*. Relationships between mentioned models (sometimes denoted as levels [37]) are very important. Models' relationships and possibilities of transformations were not studied extensively in case of hypermedia modelling. High level dependencies between core models are depicted in Figure 1. We use the UML notation where dashed arrows represent dependencies and boxes represent packages. The arrows point from the dependent (source) package to another (target) package. Each package of one of the core model is split into two subpackages. These packages represent two aspects in each core model: structure and behaviour.

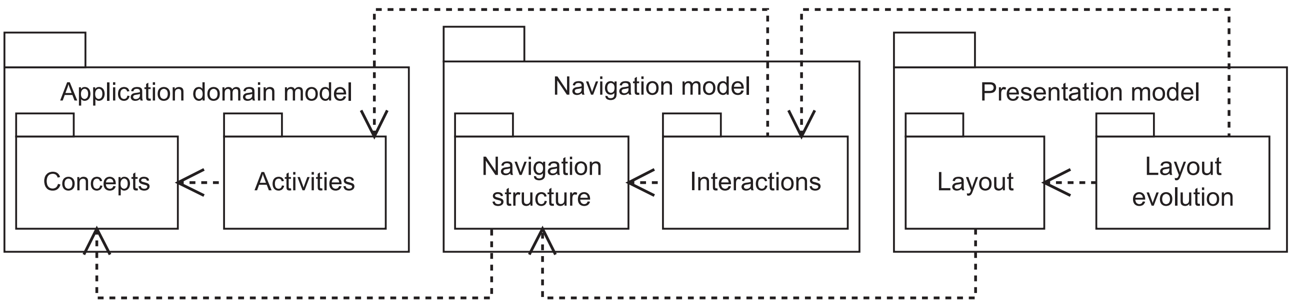

Figure 1: High-level dependencies between core hypermedia application models.

The structural *model of concepts* expresses the dependencies between concepts. The concepts from application domain form the base of a hypermedia application. Concepts index the content of the hypermedia application. They describe the information in the hypermedia application. The structural model of concepts serves besides modelling concepts per se also as an interface for accessing the information in the hypermedia application.

The dependency between the *activities model* and the structural model of concepts in an application domain means that the activities model references concepts, which describe the activities. The activities are linked usualy by transitions. This allows to look at the application domain from different perspectives (the relationships in the model have different semantics).

The *navigation structure model* references and merges concepts from modelled in the conceptual model. The concepts are merged into navigation views, packages or higher level nodes. The navigation model also instantiates the relationships between concepts from the conceptual model. Higher level navigation nodes participate in different contexts according to specific conditions. The nodes and contexts are interconnected by relationships from conceptual model or by specific navigation shortcuts.

The *interactions model* enriches the navigation structure model by events, which raise particular link (transition) between contexts and nodes. The nodes and contexts are transformed into superstates and substates. The interactions can be derived from higher level activites models. The navigation in the information space is performed according to the activities in the domain.

The navigation objects from the navigation structure model are mapped to several `layout` objects (*layout model*), which describe also the appearence of navigation objects. The symbolic alternatives of the presentation can be represented by several alternative substates. The information (concept) presented in particular context in particular navigation view can change its layout and appearence. To control this change the *layout evolution model* is introduced. Layout evolution is performed according to the interactions during browsing (navigation).

Hypermedia application perspective is open to new useful views, which ease comprehension of the hypermedia application and its development. For example, to support a personalisation a *user model* is incorporated. User model represents the relationship between the user and the information from application domain model. User modelling is very active research area especially in the field of adaptive hypermedia. However, the guidelines for modelling adaptive hypermedia applications are not so common in engineering methods for building hypermedia applications.

User models are very closed to navigation models because the user model influences interactions within a system and the user profile should also be dynamically updated according to the user interaction. In such cases the same techniques can be used for user modelling as for navigation structure modelling (such as class diagrams). In addition, they should incorporate meta information about a user, such as his preferences, goals, interests and so on. The user profile should also be dynamically updated according to the user interaction. The techniques are closely related to navigation dynamics modelling.

The classes and associations are used for user modelling in [27]. User modelling is based on the user model part of Adaptive Hypermedia Application Model (AHAM) [4]. Each user has a separate table in this model where a concept, the level of knowledge about the concept, whether it is read or whether the user is ready to read it are referenced. WebML [5] introduces more specific modelling elements: a user and a group, which are directly applied in user modelling.

Bayesian networks are employed for user modelling in [22]. The bayesian network is used as an index to the conceptual network. Probability determines to what extent the user will read next concept presented in information chunk. The probability is updated in accordance to what concept the user read. User model evolution is modelled in [27] by a state chart. The state of a user model can be changed according to an event generated from user interface.

Still different specific aspects appear that are useful to embrace into a hypermedia system model. For example, nowadays some adaptive hypermedia systems are able to adapt to something else than user characteristics. They consider environment characteristics, which are represented in an *environment model*. Environment model comprises also all aspects of the user environment that are not related to the users themselves.

The possibility for extension of proposed modelling framework can be seen in Figure 2.

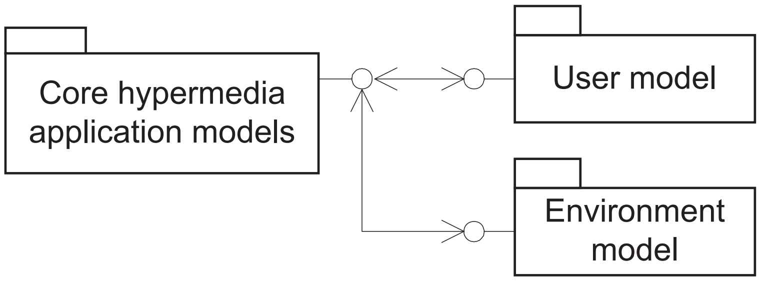

Figure 2: Extending core hypermedia application models.

The core hypermedia application models provide interface for example by means of elements in these models (concrete classes, their operations, events, transitions, etc.). The similar interface is provided by models, which extend the core models (e.g., user model, and environment model). The elements provided by the interface can be referenced directly in the core models or extending models. The interfaces are represented by circle ended line. The UML notation for an interface). The bidirectional arrow between interfaces represents their usage by packages at both sides.

We can take as an example the integration between the user model and the navigation model. Interactions cause update of the user model. The user profile structure is updated according to the concepts beeing saw during browsing in the hypermedia application. An update of the user profile structure is performed by methods and operations from the user profile structure model. The user profile can stand in several states. The states and transtions between them form the profile evolution model (dependency between the user profile structure model and user profile evolution model).

4 Discussion

Proposed categorisation can be used for decision support in hypermedia development projects. Each of described views can serve as a slot for particular technique or model created using this technique. One view represents a family of techniques (or models). Views form multidimensional space, where mentioned perspectives of taxonomy represent dimensions. The dimensions are depicted in Figure 3.

The mapping of dimensions can be for example as follows: application domain model is considered as a structural model; this structural model of the application domain is represented by formal description logics; the application domain model is created during requirements definition phase by requirements engineer. Other models can be characterised similarly in the proposed space. A project is seen as a sequence of decisions in time represented as points in multidimensional space. These decisions can be documented as guidelines or methods for hypermedia development in specific situations, which are similar to those which were made in specific project. Described approach can be denoted as *knowledge reuse* in modelling.

Several authors contribute with their categorisations especially in the general field of human computer interaction. For example Ben Schneiderman divided design guidelines for user interface design into four categories [38]: high level theories or models, middle-level principles, specific and practical guidelines, and strategies for testing. Kemp and Buckner [25] extend taxonomy based on Schneiderman's view. They consider methodology as special subcategory. They also add several subcategories. Nanards' [34] or later [35] and Lowe's [32] design space model is very similar to our approach.

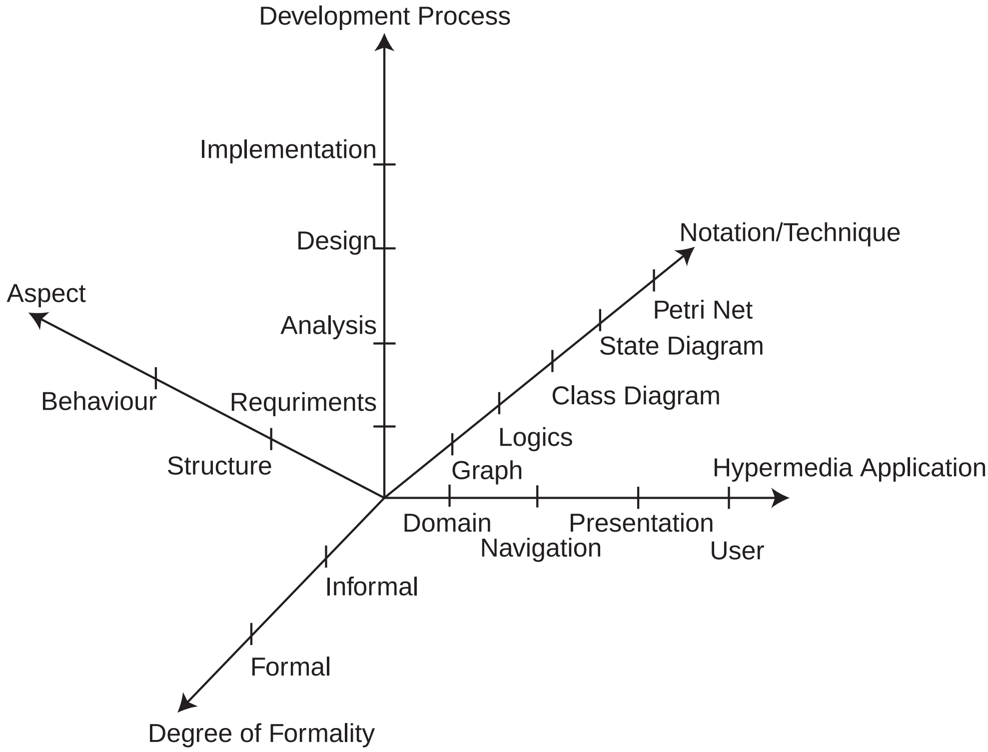

Figure 3: Multidimensional space for hypermedia modelling.

Mentioned approaches deal with design as a process. We restrict our discussion to modelling. Our approach differs from [35, 32] in the process model dimension. We separate this dimension from hypermedia application models. A process model is represented by points in the space while in mentioned approaches the process model represents dimension in a pattern space.

Our approach provides clear separation of concerns when developing hypermedia application. The main shortcoming in many current methods is that mentioned perspectives or their views are not clearly separated. For example, unlike [34] we separate language dimension from system aspect dimension.

5 Conclusions and directions for further research

In this paper we provided discussion and review of selected methods and techniques to hypermedia modelling. We discussed several points of view to modelling. According to these points of view (perspectives) we proposed a taxonomy of hypermedia related modelling techniques. This taxonomy is compared with several previously developed taxonomies. We also showed how this taxonomy can be used as a framework for modelling (modelling space). Essential views in the hypermedia application perspective with proposed dependencies between them were also discussed.

There are still several open issues in hypermedia modelling. Important finding is that current approaches to modelling hypermedia underestimate behavioural techniques.

First open issue is hypermedia development process based on reuse. We provided only high level schema for dimensions of this process, which is moreover oriented only to modelling. This issue should be evaluated, refined and supported by several guidelines.

In spite of a lot of work done in the area of hypermedia modelling, still different specific views or aspects appear such as user model or environment model. There are probably other views, which can improve understanding of hypermedia system features and can be possibly separated from those discussed in this paper. One question could be whether modelling of information or programming functional interfaces of hypermedia systems is special view of the hypermedia application or whether the interfaces could be derived from other models. Methods for existing and possibly for new introduced views should be studied and improved.

The unified language for modelling hypermedia, which is able to cover as much features of hypermedia systems as possible, is still missing. Possible way, how to overcome this problem is to integrate existing modelling languages elements into a new modelling language. The UML as de facto standard for software systems modelling can be used for this purposes.

The dependencies between models can be covered in modelling language. However, some of them should be introduced as transformation procedures or mappings. We discussed some dependencies in this paper but this issue requires further research.

Further research should be fulfilled by devising algorithms for hypermedia applications, especially for navigation and adaptation. Existing algorithms are not very effective. Algorithms for processing queries over semistructured databases, processing of mapping between models, algorithms for re-engineering information from legacy formats, etc. still constitute open issues.

References

[1] Serge Abiteboul and Anthony Bonner. Objects and views. In J. Clifford and R. King, editors, *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 238–247, San Francisco, California, March 1991. ACM Press.

[2] Luciano Baresi, Franca Garzotto, and Paolo Paolini. Extending UML for modeling web applications. In *Proc. of 34th Anual Hawaii International Conference on System Sciences (HICSS'34)*, Maui, Hawai, January 2001. IEEE Press.

[3] Martin Bichler and Stefan Nusser. Modular design of complex web-application with W3DT. In *Proc. of IEEE 5th Workshops on Enabling Technology: Infrastructure for Collaborative Enterprises (WET ICE 96)*, Standford, California, USA, June 1996.

[4] Paul De Bra, Geert-Jan Houben, and Hongjing Wu. AHAM: A dexter-based reference model for adaptive hypermedia. In K. Tochtermann, J. Westbomke, U.K. Wiil, and J. Leggett, editors, *Proc. of ACM Conference on Hypertext and Hypermedia*, pages 147–156, Darmstadt, Germany, February 1999.

[5] Stefano Ceri, Piero Fraternali, and Aldo Bongio. Web Modeling Language (WebML): a modeling language for designing web sites. *Computer Networks and ISDN Systems*, 33(1–6):137–157, June 2000.

[6] Derek Coleman, Fiona Hayes, and Stephen Bear. Introducing objectcharts or how to use statecharts in object-oriented design. *IEEE Transactions on Software Engineering*, 18(1):9–18, January 1992.

[7] Jim Conallen. UML extension for web applications, March 1999. White paper.

[8] Krysztof Czarnecki and Ulrich Eisenecker. *Generative Programing: Principles, Techniques, and Tools*. Addison Wesley, 2000.

[9] Peter Dolog. The aspects of hypermedia modelling. In *Proc. of 3rd Conference on Electrical Engineering and Information Technology for PhD Students*, Bratislava, Slovak Republic, September 2000. Slovak University of Technology, Bratislava.

[10] Peter Dolog. Modelling in hypermedia development, August 2001. Technical Report (A Written Part of PhD Examination). Department of Computer Science and Engineering, Slovak University of Technology.

[11] Peter Dolog and Mária Bieliková. Modelling browsing semantics in hypertexts using UML. In J. Zendulka, editor, *Proc. of ISM'2001 - Information Systems Modelling*, pages 181–188, Hradec nad Moravicí, Czech Republic, May 2001.

[12] Peter Dolog and Mária Bieliková. Navigation modelling in adaptive hypermedia. In Paul De Bra, Peter Brusilovsky, and Ricardo Conejo, editors, *Proc. of 2nd Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, pages 586–591, Malaga, Spain, May 2002. Springer LNCS 2347.

[13] Peter Dolog and Mária Bieliková. Towards variability modelling for reuse in hypermedia engineering. In Yannis Manolopoulos and Pavol Návrat, editors, *Proc. of 6th East-European Conference on Advances in Databases and Information Systems — ADBIS'2002*, pages 388–400, Bratislava, Slovakia, 2002. Springer LNCS 2435.

[14] Piero Fraternali and Paolo Paoliny. Model-driven development of web applications: The autoweb system. *ACM Transactions on Information Systems*, 28(4):323–382, October 2000.

[15] Martin Gaedke, Chrstian Segor, and Hans-Werner Gellersen. WCML: Paving the way for reuse in object-oriented web engineering. In *Proc. of 2000 ACM Symposium on Applied Computing (SAC 2000)*, Villa Olmo, Como, Italy, March 2000.

[16] Franca Garzotto and Paolo Paolini. HDM — a model-based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1):1–26, January 1993.

[17] Hans-Werner Gellersen and Martin Gaedke. Object-oriented web application development. *IEEE Internet Computing*, 3(1):60–68, 1999.

[18] Frank Halasz, T. Moran, and R. Trigg. Notecards in a nutshell. In *Proc. of CHI and GI Conference*, pages 45–52, Toronto, Canada, April 1987.

[19] Frank G. Halasz and Meyer Schwartz. The Dexter Hypertext Reference Model. *Comunications of the ACM*, 37(2):30–39, February 1994.

[20] Lynda Hardman, Dick C.A. Bulterman, and Guido Van Rossum. The Amsterdam Hypermedia Model: Adding time and context to the dexter model. *Comunications of the ACM*, 37(2):50–64, February 1994.

[21] Rolf Hennicker and Nora Koch. A UML-based methodology for hypermedia design. In S. Stuart A. Evans and B. Selic, editors, *Proc. of UML 2000 Conference*, York, England, October 2000. Springer LNCS 1939.

[22] Nicola Henze and Wolfgang Nejdl. Bayesian modeling for adaptive hypermedia systems. In *Proc. of LWA'99: Lernen, Wissensentdeckung und Adaptivität — ABIS'99: 7ter Workshop Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen*, Magdeburg, Germany, September 1999.

[23] Jeroen Hoppenbrouwers. Advanced conceptual network usage in library database queries, September 1998. Technical Report. Infolab, Tilburg University, The Netherlands.

[24] Tomás Isakowitz, A. Kamis, and M. Koufaris. Extending the capabilities of RMM: Russians dolls and hypertext. In *Proceedings of the 30th Annual Hawaii International Conference on System Sciences*, January 1997.

[25] B. Kemp and K. Buckner. A taxonomy of design guidance for hypermedia design. *Interacting with Computers*, 12(2):143–160, November 1999.

[26] K.G.W. Goossens. Structure and behaviour in hardware verification. In J.J. Joyce and C.-J.H. Seger, editors, *International Workshop on Higher Order Logic Theorem Proving and its Applications*, volume 780, pages 73–87, Vancouver, Canada, 1994. Springer-Verlag.

[27] Nora Koch. Towards a methodology for adaptive hypermedia systems development. In Timm and Marc Rössel, editors, *Proc. of ABIS-98: Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen*, October 1998.

[28] Miloš Kravčík. Internet based collaboration: the presence and perspectives. In *Proc. of WebNet 97*, Toronto, Canada, November 1997.

[29] Heeseok Lee, Choongseok Lee, and Cheonsoo Yoo. A scenario-based object-oriented hypermedia design methodology. *Information and Management*, 36(3):121–138, September 1999.

[30] Mark Levene and George Loizou. Navigation in hypertext is easy only sometimes. *SIAM Journal on Computing*, 29(3):728–760, 1999.

[31] Mark Levene and George Loizou. A probabilistic approach to navigation in hypertext. *Information Sciences*, 1–4(114):165–186, 1999.

[32] David B. Lowe, Andrew J. Bucknell, and Richard G. Webby. Improving hypermedia development: a reference model-based process assessment method. In *Proceedings of the ACM International Conference on Hypertext and Hypermedia(Hypertext '99)*, pages 139–146, Darmstadt, Germany, February 1999.

[33] Jin-Cheon Na and Richard Furuta. Context-aware hypermedia in dynamically-changing environment supported by a high-level petri net. In *Proc. of 11th ACM Conference on Hypertext and Hypermedia*, pages 222–223, San Antonio, Texas, June 2000.

[34] Jocelyne Nanard and Marc Nanard. Hypertext design environments and the hypertext design process. *Communications of the ACM*, 38(8):49–56, August 1995.

[35] Jocelyne Nanard and Marc Nanard. Toward an hypermedia design patterns space. In *Proc. of 2nd Workshop in Hypermedia Development: Design Patterns in Hypermedia*, 1999.

[36] Fabiano Borges Paulo, Paulo Cesar Masiero, and Maria Cristina Ferreira de Oliviera. Hypercharts: Extended statecharts to support hypermedia specification. *IEEE Transactions on Software Engineering*, 25(1):33–49, January–February 1999.

[37] W. Retschitzegger and W. Schwinger. Towards modeling of dataweb applications - a requirements' perspective. In *Proc. of the Americas Conference on Information Systems (AMCIS)*, Long Beach, California, August 2000.

[38] Ben Schneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, 1998.

[39] Daniel Schwabe and Gustavo Rossi. An object-oriented approach to web-based application design. *Theory and Practise of Object Systems (TAPOS), Special Issue on the Internet*, 4(4):207–225, October 1998.

[40] Christian Segor and Martin Gaedke. Crossing the gap - from design to implementation in web-application development. In *Proc. of Information Resources Management Association International Conference 2000*, Anchorage, USA, May 2000.

[41] Luiz Fernando G. Soares, Rogério F. Rodrigues, and Débora C. Muchaluat Saade. Modeling, authoring and formatting hypermedia documents in the HyperProp system. *ACM Multimedia System Journal*, 8(2):118–134, March 2000.

[42] P. David Stotts and Richard Furuta. Petri-net-based hypertext: Document structure with browsing semantics. *ACM Transactions on Information Systems*, 7(1):3–29, January 1989.

[43] P. David Stotts and Richard Furuta. Hyperdocuments as automata: Verification of trace-based browsing properties by model checking. *ACM Transactions on Information Systems*, 16(1):1–30, January 1998.

[44] Manfred Thüring, Jörg Hannermann, and Jörg M. Haake. Hypermedia and cognition: Design for comprehension. *Communications of the ACM*, 38(8):57–66, August 1995.

[45] Klaus Tochtermann and Gisbert Dittrich. The dortmund faminly of hypermedia models—concepts and their applications. *Journal of Universal Computer Science*, 2(1):34–56, 1996.

[46] Weigang Wang and Roy Rada. Structured hypertext with domain semantics. *ACM Transactions on Information Systems*, 16(4):372–412, October 1998.