

Generic Adaptation Framework: a Process-Oriented Perspective

Evgeny Knutov, Paul De Bra, Mykola Pechenizkiy

Department of Computer Science

Eindhoven University of Technology

P.O. Box 513, 5600 MB

Eindhoven, the Netherlands

e.knutov@tue.nl, debra@win.tue.nl, m.pechenizkiy@tue.nl

Abstract

Adaptive Hypermedia Systems (AHS) have long been mostly domain- or application-specific systems. Existing reference models provide domain-independent high-level descriptions of AHS. They focus on abstract data structures and only briefly describe the *adaptation process* in a generic way. In this paper we consider the process aspects of AHS from the very first classical *user modelling-adaptation* loop to a generic detailed flowchart of the adaptation in AHS. We introduce a Generic Adaptation Process (GAP) and by aligning it with a layered (data-oriented) AHS architecture we show that it can serve as the process part of a new reference model for AHS.

1 Introduction and Background

Hypermedia applications provide their users with the freedom to navigate through a large hyperspace whereas Adaptive Hypermedia (AH) offers personalized content, presentation, and navigation support. Adaptive systems are becoming more popular and are being used to perform automatic personalization, by selecting and presenting information. Adaptation of an information system or service to a user has been proven to be a powerful and useful concept (Brusilovsky 2001). It is especially helpful in reducing the information overload which is frequently experienced on the Internet or any other information system of a large scale.

Throughout the development of the Hypermedia and later AH research field people have been trying to create *reference* models of these categories of systems. Major reference models have been favouring a layered architecture, starting with the Dexter Hypertext Model (Halasz and Schwartz 1990; 1994), and later the Tower Model (De Bra et al. 1992) (introduced as the Extensible Data Model for Hyperdocuments). Later this was continued in the adaptive hypermedia field with the most referenced AHAM model (De Bra et al. 1999), followed by other systems/models, such as LAOS (Saksena and Cristea 2006), APeLS (Conlan and Wade 2004), the Munich model (Koch 2001). However, these developments were mostly concerned with the structure and/or the data model, but not as much with the process underlying the adaptation.

Newly developed systems have brought in new terms, concepts, approaches, models, methodologies, prototypes and use-cases. But in general they have not led to a clearer picture on the *ideal* adaptation process and haven't even introduced such a notion. Although there is a larger variety of specialized systems than ever before, there is still no consensus as to what is the *ideal* generic (or general purpose) AHS would be, and in particular what its adaptation process should be.

In this paper we examine the issue of aligning the adaptation process, based on an extensive list of AH methods and techniques (Knutov et al. 2009), with a layered structure of AHS. We show that to some extent the process influences and defines the composition and the ordering of such a layered structure in such a way that it partially arranges the order of the layers, defines couplings and determines the major transitions in the system. We show that the process driven approach gives more insight into AH development methods and the composition of the AH system.

The main goal of a new GAF (Generic Adaptation Framework) reference model in this area is to provide the architecture, to separate essential elements from optional ones and to define criteria to distinguishing between these elements, to provide a modular structure that can be used either separately or together, depending on the needs of the intended application and that can be developed over a generic framework to satisfy different needs.

1.1 Adaptation Process Modelling

Hereafter by *Generic Adaptation Process (GAP)* we mean the interaction in AHS which starts with the goal statement, exploits features of the user and domain models in different contexts and adapts various aspects of the information and presentation to the user. Figure 1 shows this user modelling / adaptation loop as originally presented in (Brusilovsky 1996).

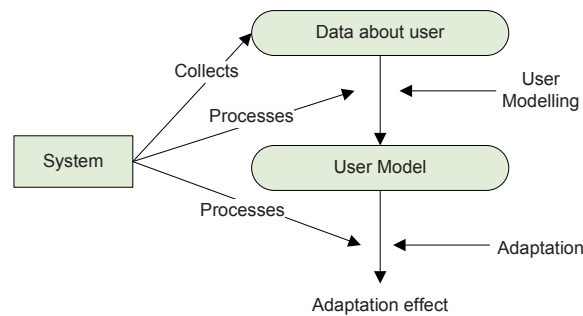


Figure 1: Classic loop user modelling - adaptation

Considering a generic adaptive system one may think not only about defining a framework or reference (data) model but also about what the adaptation process within the system looks like, beyond what Figure 1 shows. Next two figures show some extensions of the classical loop, taking into account that selection of user information or reasoning about the user model to obtain answers about the user is an essential part of the adaptation process (Figure 2) and that either the user or an administrator (or both) need the ability to scrutinize the user model (Figure 3).

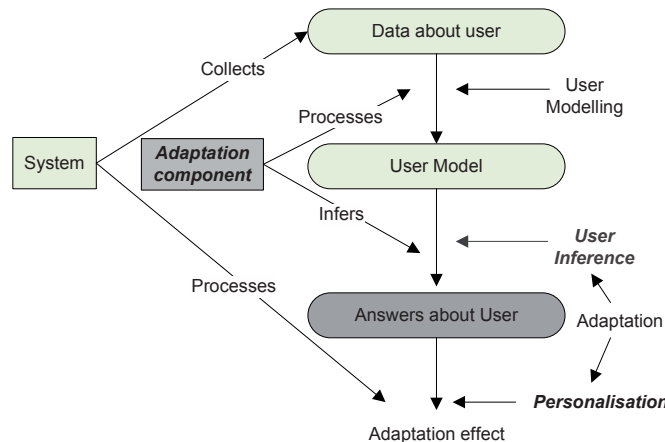


Figure 2: User model inference - adaptation loop

These updated *user-modelling - adaptation* loops give a more extensive overview of the adaptation. In (Knutov et al. 2009) we took a different approach: we extended the classification of initial AH Methods and Techniques with an adaptation process cycle to give the first insight of the GAP flow, shown in Figure 4.

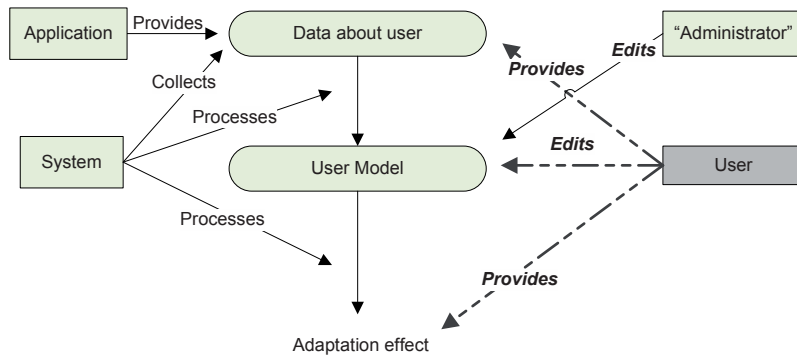


Figure 3: Collaborative user model editing (User and Administrator involved)

Although coupling the AH methods classification with the *adaptation process* had a different purpose from what is shown in the classical (and later) loops of *user modelling - adaptation* our goal is to show the diversity of the adaptation process representation and the possibility of aligning not only the *user-modelling - adaptation* loop into the adaptation process but the adaptation methods and techniques as well.

We consider that most of the AHS aspects can be aligned in the *adaptation process* which can serve as a reference process for AHS design, comparison and evaluation.

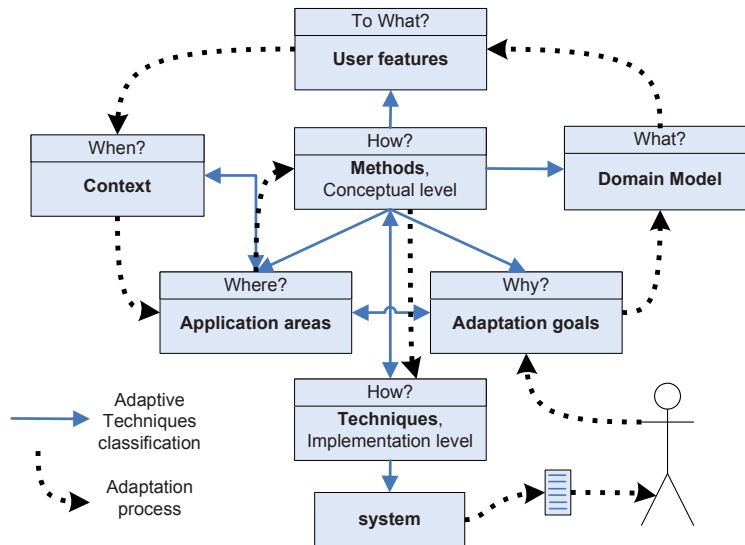


Figure 4: Classification of AH methods and techniques; adaptation process highlights (Knutov et al. 2009)

1.2 Goals

In this paper we describe the reference adaptation process, aligning it with the traditional *adaptation questions* (section 2) and formalizing it in a single generic manner. In particular, we:

- provide a flowchart diagram of a generic AHS;
- place the notion of the adaptation process in the context of a generic layered AHS;
- align the layers of AHS in a “sequence chart” and present the reference adaptation process.

1.3 Outline of the paper: contribution

In this paper we introduce the notion of *GAP*, which consolidates most of the developments in the AHS field. Our approach starts from a methods and techniques classification, which leads to a layered-based model. We then match the adaptation process with the layered structure. We design a suitable layered structure for AHS (Knutov et al. 2009) and get to the process representation by rotating these (normally horizontal) layers of the AH framework 90 degrees which eventually turns into an informal sequence chart (section 4). It shows inter-layer connections, information and control flows first of all in the AHS, and at the same time it easily matches the workflow representation. Thus we have a two-fold process presentation which can to some extent be derived from the AHS structure and which can also drive requirements for modular AHS composition.

We present a proof of concept of the *generic* process by showing 3 diverse use-cases (flowchart and sequence chart) (section 6). We show that our adaptation framework encloses both types of process representations in one go. The first example presents a classical Adaptive course (Carro et al. 2003; Bra et al. 2006) which can be easily aligned with the generic adaptation workflow. The second example scrutinizes a different field and shows web search and the way we are used to query information on the Web being aligned with the adaptation sequence. In the third example we consider Recommender Systems (RS) compliance. We briefly touch upon provenance issues and show that our way of presenting the adaptation process and aligning it with the classical classification questions of AH methods and techniques can be enhanced with provenance information by aligning the adaptation process model with process-centric provenance models (section 5). This will provide opportunities for the data lineage and supports more elaborate reasoning and thus adaptation.

2 Questions of Adaptation and Adaptation Sequence, Related Work

Adaptation can be defined by posing and answering six major questions:

- Why do we need adaptation? (*Why?*)
- What can we adapt? (*What?*)
- What can we adapt to? (*To What?*)
- When can we apply adaptation? (*When?*)
- Where can we apply adaptation? (*Where?*)
- How do we adapt? (*How?*)

This type of classification has been initially introduced in (Brusilovsky 1996). Here we not just revisit these questions, but address the issue of aligning them (also aligning the corresponding methods, techniques and respective modules (layers) of AHS) in *GAP* which can serve as a process guideline and framework for defining the way an AHS functions.

Figure 4 considers an order in which the adaptation questions could be asked (and answered), thus leading to the first informal definition of the adaptation process. The *classification* of AH methods and techniques is outlined by the solid lines representing the typical dimensions for the analysis of adaptive systems (Schneider-Hufschmidt et al. 1993); at the same time we join the same classification blocks considering the adaptation *process perspective* which is depicted by dotted lines. This process is usually initiated by the user stating the adaptation goal and thus answering the *Why?* question. Then in the process we consider the *What?* and *To What?* questions, which emphasize Domain Model (DM) and UM descriptions. *When?* and *Where?* go next providing context and application area definitions. Lastly, the *How?* question describes methods and techniques at the conceptual and implementation level.

Taking into account user needs and system components (anticipating both core and optionally available components) we would like to present a process which explains the transitions, states, sequences and flows in a generic AHS. First we revisit a few such systems. Then, based on the research and summarization done in (Knutov et al. 2009), we present the flowcharts of an adaptive system and finally come up with the conceptual sequence chart of a layer-structured *Generic Adaptation Framework (GAF)*.

Considering the adaptation process in other systems we mention a few examples of how the authors tried to capture an idea of defining the adaptation processes (both implicitly or explicitly) in their systems and matching processes with the layered structure of their systems.

In the GOMAWE system (Balík and Jelínek 2007) (Figure 5) the authors tried to fit the adaptation process in the general ontological model of the system they designed. Though there is still much to be considered in terms of the real inter-layer transitions, we can already observe a few basic transitions such as the Event Interface which either triggers the *Push Reasoning* or provides the data for the *Pull Reasoning* interfaces of the Reasoning layer. Here *Push* is responsible for transforming user events into UM updates which (events) happen when users interact with the system, and *Pull* retrieves the UM state. Moreover these connections tie different layers of the designed system together.

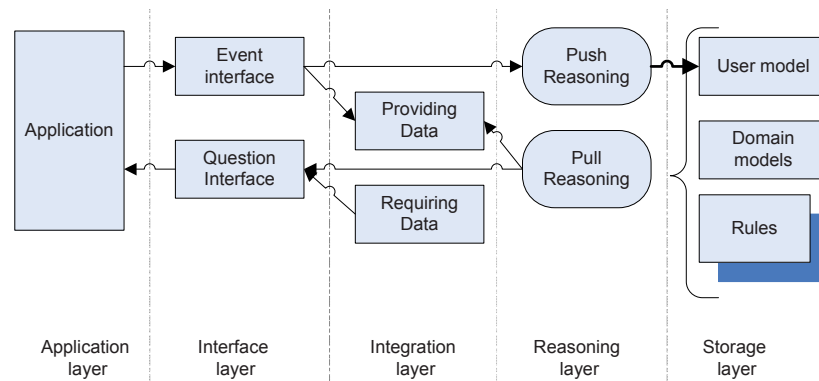


Figure 5: Overview of the General Ontological Model for Adaptive Web Environments (GOMAWE)

Another Generic Adaptivity Model was presented in (de Vrieze 2006) (Figure 6). It provides a cycle view of the adaptation process trying to match it with a layered structure. However, it doesn't represent any of the AHS adaptation functionality.

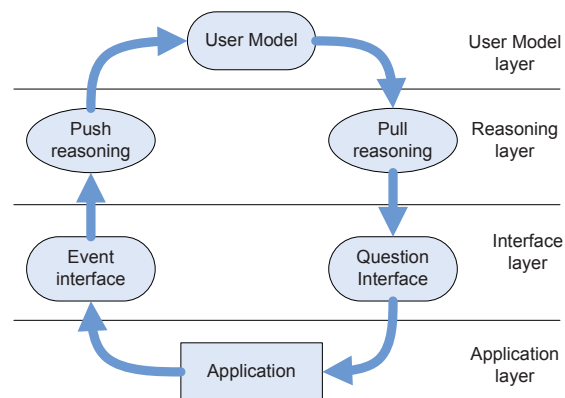


Figure 6: Overview of the Generic Adaptivity model

The Munich model defined by Koch in (Koch 2001; Koch and Wirsing 2002) presented the lifecycle model of adaptation in the UML formalized notion (Figure 7).

It defines the following 'layers' or components or states to be tied by these process loops: presentation, interaction, user observation, and adjustments of the systems (which include Adaptation itself and UM updates). These cycles start with an initial presentation and a default UM. Stereotypes are usually used to provide the information for the initial UM. Then the following steps of adaptation cycle follow (Koch 2001):

- **System Interaction** — which describes how to react to certain user action(s), resulting in the termination of this cycle and adaptive continuation;
- **User observation** — in which the evaluation of the information retrieved from UM is being done;
- **Adjustments** — comprising the two sub-states:
 - User model update — in which UM attributes are updated;
 - System adaptation — in which the adaptation is performed (adaptation of presentation, content or navigation) utilizing the state of UM.
- **Presentation** — when the system presents the adaptable elements taking into account what the information system knows about the user; the system remains in this state until the user starts interacting with system again.

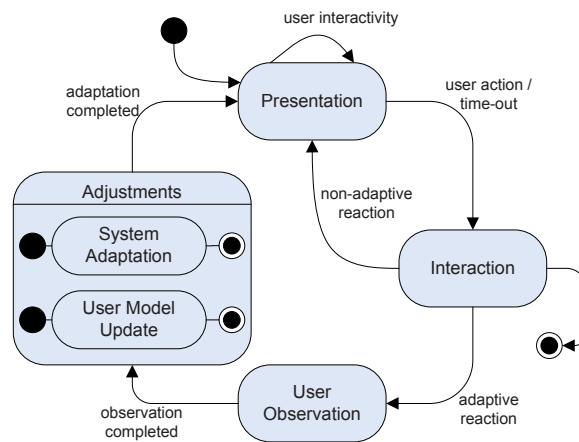


Figure 7: Lifecycle Model of Adaptation (Munich Model)

To some extent most of the adaptation loops fall under this classification. The interactions are continuous and recursive when the user continues using the system and explores the knowledge base in depth. We should also mention that here we don't consider any concurrent loops that may happen and influence each other in every aspect.

The General meta-Model for AHS, presented in (Seefelder de Assis and Schwabe 2004), captures input-output processes of an AH meta-model (Figure 8). Though these input-output parameters sometimes don't connect particular blocks in each model represented in the figure, they denote the process flow of the system, e.g. User Context is required as an input for the Adaptation Model to perform the adaptation, or the Adaptation functions update UM. Thus it might be useful to show the data flows of the system in terms of the input/output of each component as presented in Figure 8.

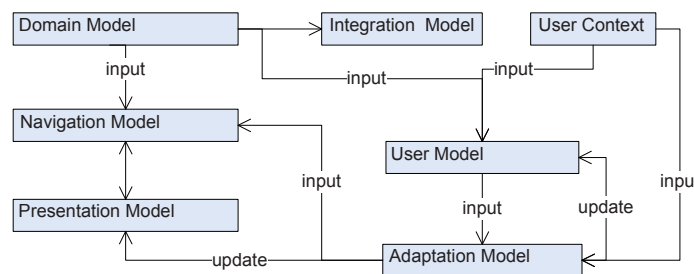


Figure 8: General meta-model for AHS

3 Adaptation Process Flowcharts

In the following section we summarise the procedural knowledge of the data/control and other flows in AHS and come up with a generic representation of AHS processes.

Hereafter we present the adaptation process flowcharts, which generalize the functionality of the AHS. In fact these flowcharts follow the system properties summarization presented in [(Knutov et al. 2009), (Tables 1, 2)]. Based on the summarised (and generalized) functionality we devise these adaptation process flowcharts.

We distinguish the following flowcharts:

- abstract adaptive process flowchart (Figure 9);
- goal acquisition and adaptation (Figure 10);
- adaptation functionality (Figure 11);
- test-feedback functionality (Figure 12).

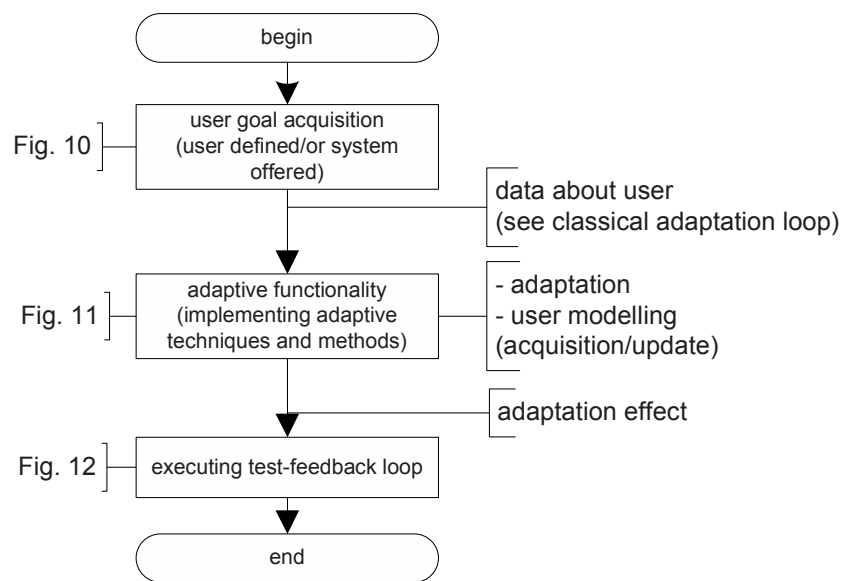


Figure 9: Generic Adaptation Flowchart (to be considered as the aggregation of Figures 10 - 12)

Each flowchart represents a certain aspect of the adaptation process, annotated to give more insight in the description of some blocks. On the right side of each chart we link parts of the process to the layers of the GAF model. The communication between the layers is illustrated in Figure 14. We also mark with numbers the exact correspondence of Figure 14 calls and transitions with the outlined blocks on the flowcharts (Figures 9-11) in order to show the conformity of the sequence and flowchart approaches.

As a result of this generalization we present the flowchart of an adaptive course use-case in Figure 15, which fully complies with the generic representation.

In *Goal acquisition and adaptation flowchart* (Figure 10) we start with the group analysis, thus assigning the user to a group or acquiring group properties in order to take them into account while choosing the adaptation goals. Here we also make assumptions that the user may belong to one group and may not change the group within a session. The user may have his/her own goal or be advised by the system, as well as proposed to use the group goal. In any case goal suitability is checked to determine whether the user can follow it. All suitable goals are elaborated in a sequence of concepts or the most appropriate *project* (defined set of concepts to study) is chosen. The *Adaptation functionality flowchart* (Figure 11) presents the main Adaptation Engine (AE) functionality in a sequence of concept-content adaptation steps for a particular user. In general we analyse conditions for a particular

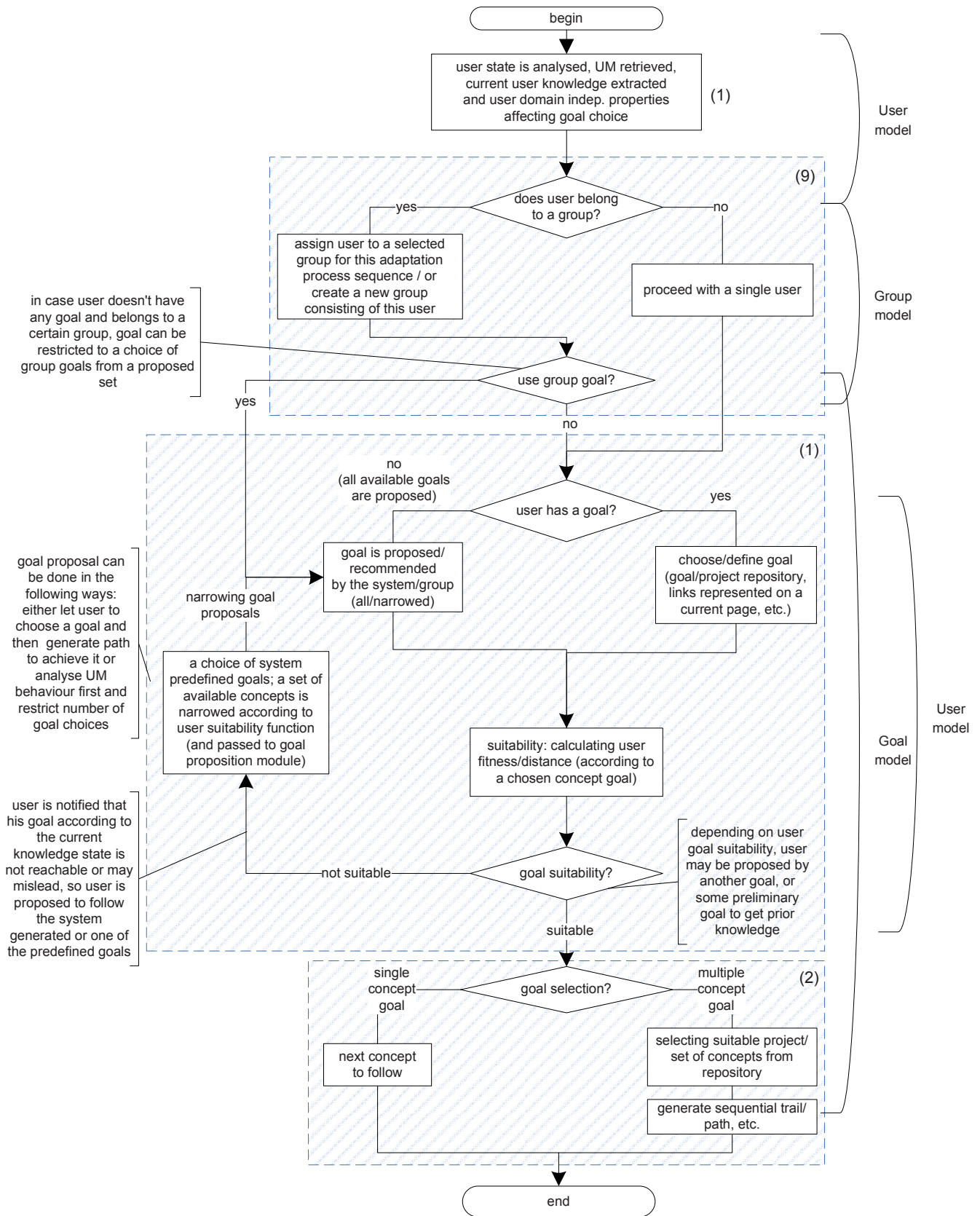


Figure 10: Goal acquisition and adaptation flowchart

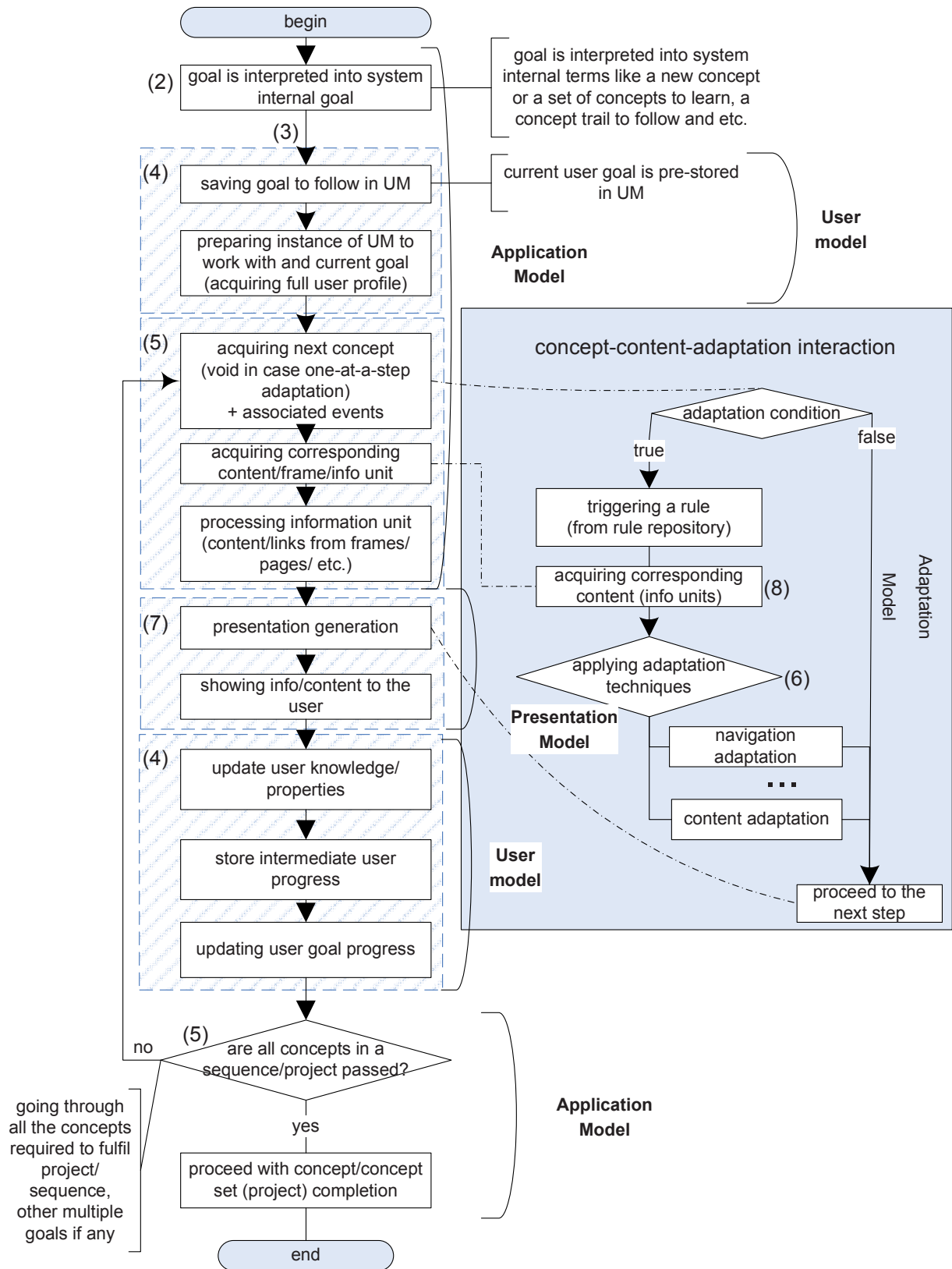


Figure 11: Adaptation functionality flowchart

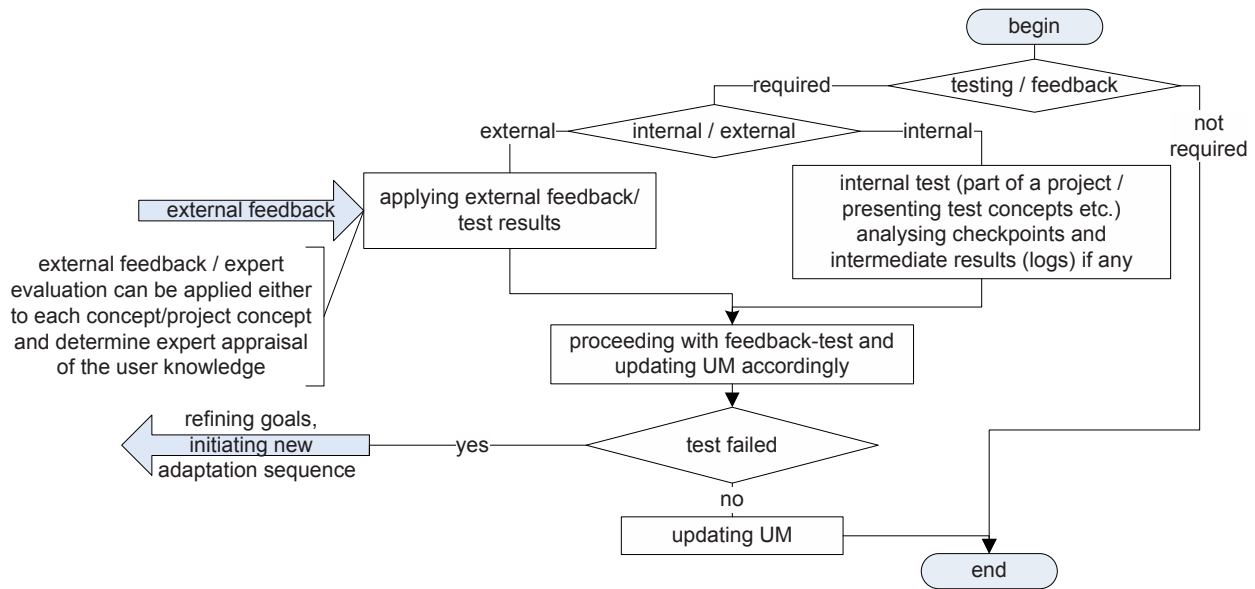


Figure 12: Test-feedback functionality flowchart

step and execute adaptation rules which apply adaptation techniques and perform presentation, content and navigation adaptation. After that UM attributes are updated accordingly and the user proceeds with the next concept either on a *one-per-click* or project-organized basis. This figure looks very similar to what was done in the IMMPS model (Bordegoni et al. 1997), presenting a reference architecture for intelligent multimedia presentation systems where the knowledge server was separated from the main flowchart in order to separate and retain the knowledge base from other system functionality. For the same reasons to separate AE functionality we have the distinguished *concept-content adaptation interaction* block. The flowchart presents the case where adaptation is done and then the user model is updated. However, an AHS may also decide to first update the UM and then perform adaptation. AHA! (De Bra and Calvi 1998; De Bra et al. 2006) and GALE (van der Sluijs and Höver 2009) do it this way. *Figure 12 represents Test-feedback functionality.* Here if such a feedback is required the user continues either with the external evaluation or internal assessment which could be part of a project or a separate questionnaire or test instance. If a user failed the test, user goals might be refined and he/she could be requested start all over again.

4 ‘Rotating’ the Layers of AHS: Adaptation Process and the Layered Model

Figure 13 presents the conceptual structure of GAF (Knutov 2009) in which we align the order of the layers in the system according to the classification of AH methods and techniques (Figure 4). Though this order represents the basic understanding of the adaptation questions every particular system may vary or even omit some of these, thus leading to a different composition of the system layers determined by the different adaptation process.

Now considering the generalized adaptation process flow-charts presented in (Figures 9-12) and the layered nature of AHS (Knutov et al. 2009) we present GAP within the layered GAF model. We believe that in order to couple, align, sort and arrange the layers of such a system (both the generic model or some particular domain focused implementation) one should keep in mind an adaptation process sequence that will partially determine the layer arrangement and to some extent will define the mandatory and optional elements and drive the system design.

Thus we decided to rotate the anticipated layered structure representation (which normally has horizontal layers) by 90 degrees counter-clockwise and match it with the adaptation process flowchart. Figure 14 shows such an abstraction of GAP in terms of the system layers. (It appears rotated back as it only fits on the page in landscape orientation.)

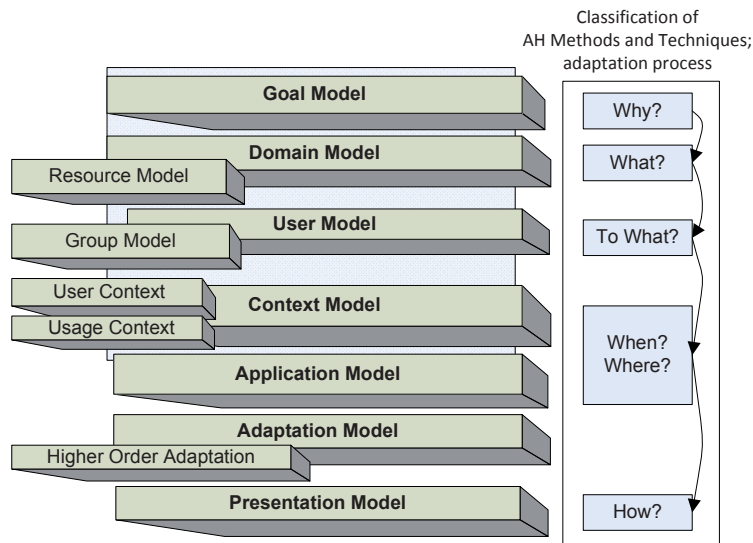


Figure 13: Conceptual scheme of a Generic Adaptation Framework (GAF)

We have marked the communication arrows with numbers to set up a correspondence with the flowcharts, where respective blocks are outlined and marked with the same numbers. This is done to show the coherence of the sequence and flowcharts. We should also note that not every connection in the adaptation process sequence exists in the above-mentioned flowcharts due to the more extensive description of the GAF process sequence chart. The marked connections are:

- (1) — User goals are defined. In case user doesn't define any goal it can be proposed by the system or a group goal can be used;
- (2) — User goals are aligned with DM, considering the conceptual structure of the domain. According to the selected goal a suitable set of concepts to follow is chosen;
- (3) — Adaptation is initiated and control is passed to the Application Model (AM);
- (4) — Operations of UM properties such as acquisition (which is performed before actual adaptation) and update (which may be done before as well as after the actual adaptation execution depending on the system implementation) (corresponds to a few places on the flowchart);
- (5) — Operations mainly concerned with working with the concepts from DM;
- (6) — Appropriate adaptation methods and techniques are invoked;
- (7) — Retrieved content is passed to the Presentation model to be rendered/generated and presented to the user;
- (8) — Corresponding content (for concerned concepts) is retrieved from the Resource model and handed over back to AM;
- (9) — Group related operations (assigning users, retrieving group properties, defining new groups, etc.).

5 Provenance Aspects of the AHS Process

Provenance is information about the origin, ownership, source, history, lineage and/or derivation of an information object or data. Provenance is important in any AHS as it is vital for providing a detailed explanation of user action and system usage, data origin and inference, ensuring analysis of dependencies in the system and repeatability of user and system interactions. There are several provenance modelling classifications including: *Process-centric* modelling: it refers to the description of the process with particular change steps within the system through which

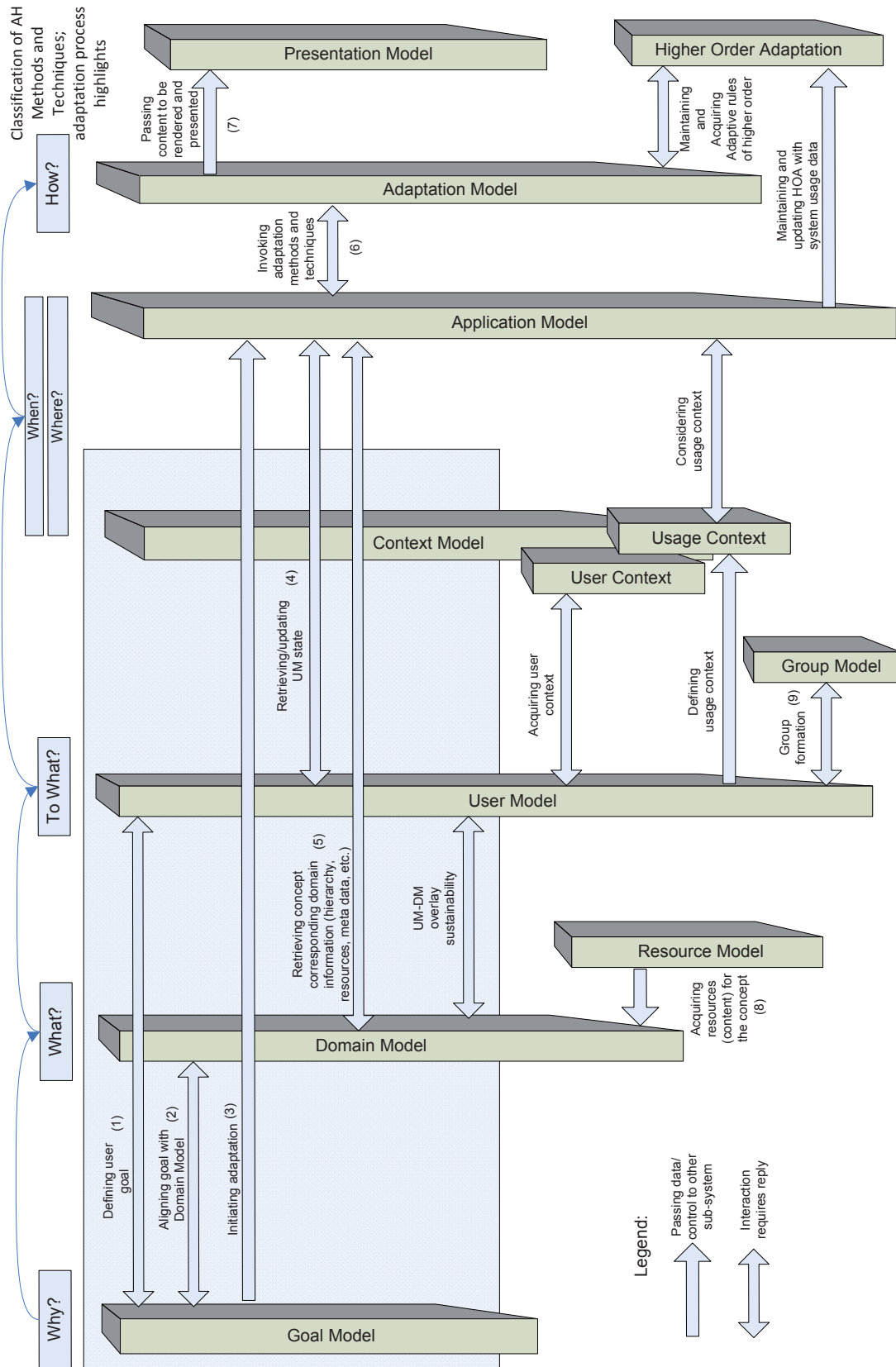


Figure 14: Conceptual Generic Adaptation Process sequence chart

Table 1: Aligning Adaptation and Provenance questions

Questions	AHS	Provenance Model	Comments
What?	Domain Model	denotes the sequence of events that affect the data object	answers describe the sequence of events when the user gets access to the domain information and acquires domain knowledge
Who? (To Whom?) Which?	describes the user profile selection (or/and device usage) (e.g. can be used to select a group or target users)	the set of all agents and/or devices involved in the process	
To What?	UM attributes (selecting particular attributes that are accessed and updated within the concerned adaptation process)	no actual representation in terms of provenance question, however historical information on accessing and updating UM represents provenance information	
Why?	stating the adaptation goal(s) (might be a domain concept, representing either a new goal to follow or a sequence of concepts)	the set of reasons for triggering a particular event (evidence of what has happened)	reasons and goals are complementary, indicating the premises of the adaptation process
When? Where?	Application Model (which serves as the core of the system: coupling other layers and dispatching information in AHS) and Context information keeps track and interprets the context information	the set of event times and locations	contextual information in general
How?	describing AH methods and techniques on a conceptual and implementation level (Adaptive Engine (AE) functionality); explains the sequence of event-actions; <i>describes the semantics</i> of cause-effect relations	the set of all actions leading up to the events (keeping track of the events, and corresponding action in the system); <i>describes the syntax</i> of events and actions recorded	in pair provenance and AH describe the syntax and semantics of AE functionality (record events and actions and show cause-effect relationship)

provenance data is obtained. It collects not only the data about a particular step, but about the application processes as well; and *Pipeline-centric* (Groth et al. 2009) which was introduced for workflow-based applications, and helps to determine the provenance of the application output based on the provenance graph of the application.

Considering the question-centric, extensive definition of the W7 Provenance Model (Ram and Liu 2006; Ram 2006) and the AH methods and techniques classification questions (Brusilovsky 1996; Knutov et al. 2009) we can essentially align the questions and corresponding answers with the adaptation process/sequence. This alignment provides us with the adaptation provenance. Due to this we see the commonalities in the semantics of these answers and questions. The idea of provenance information in AHS is emphasized in (Knutov et al. 2010) and not

elaborated in detail here.

One of the important points is that provenance data can be directly used in the adaptation process. It can be interpreted by AE and used to determine the result of the next adaptation step; it extends the capabilities of the adaptive reasoning from a conventional pre-authored type to more context and provenance/lineage dependent by taking the process of obtaining concerned data (value or property) into account. Considering the aforementioned process and pipeline centric types of provenance we may map the provenance on the adaptation processes shown in Figure 14, where each step of the adaptation sequence will represent a single data transformation and will answer both questions of Provenance and AH. These questions and corresponding answers can be found in Table 1 (Knutov et al. 2010).

6 Use-Cases

In this section we present three different examples of GAP applicability in three different contexts and with different representations. In first use-case we deal with an adaptive course for a university environment and we present it using the generic process flowchart. The second use-case shows how the generic process sequence chart complies with the web search process. The third use-case considers recommender system (RS) compliance with the GAP (also using the generic process sequence chart).

6.1 Course Adaptation Use-Case

As a proof of concept in Figure 15 we present an adaptive course use-case and describe the steps covered by the scheme.

The scheme in Figure 15 covers the following steps:

- Goal selection starts with the link selection, so that the possible goals correspond to the links the learner sees in the presentation; so a user selects the goal by following the first link and this goal may need to be elaborated in order to proceed with adaptation:
 - In case when no goal elaboration is needed the system will proceed with content adaptation according to user knowledge and goal (next iteration).
 - In case of a composite goal, elaboration will be done according to the goal repository structure, projects structure and so on until the most suitable goal sequence is found to fulfil user requirements (each iteration going one level deeper within the hierarchical structure if required).
- Having decided upon the goal (or a sequence of goals) it will be mapped on the DM concept structure to start with content adaptation.
- The current UM state will be acquired to proceed with the next step of rule parsing, acquiring the corresponding content and performing presentation generation until the system will go through all concepts designated to be learned by the user.

6.2 Adaptation and Search Use-Case

Figure 17 presents a picture of compliance of a search process with an overlaying GAP sequence chart. Here we assign search process steps from Figure 16 to a single layer or a transition in the system. Though we are facing certain issues discriminating Recommendation Engine functionality, in particular Search Engine and Ranking mechanisms (in this respect Application Model (AM) and Adaptation Model/Engine (AE) can be treated accordingly) we could align the search process and describe its functionality (in terms of aforementioned models) with GAF. On the one hand this proves the generic nature of GAF, and on the other hand it opens new horizons to facilitate search aspects in the AH field.

The search process complies with the reference structure of AHS and generic process sequence as follows:

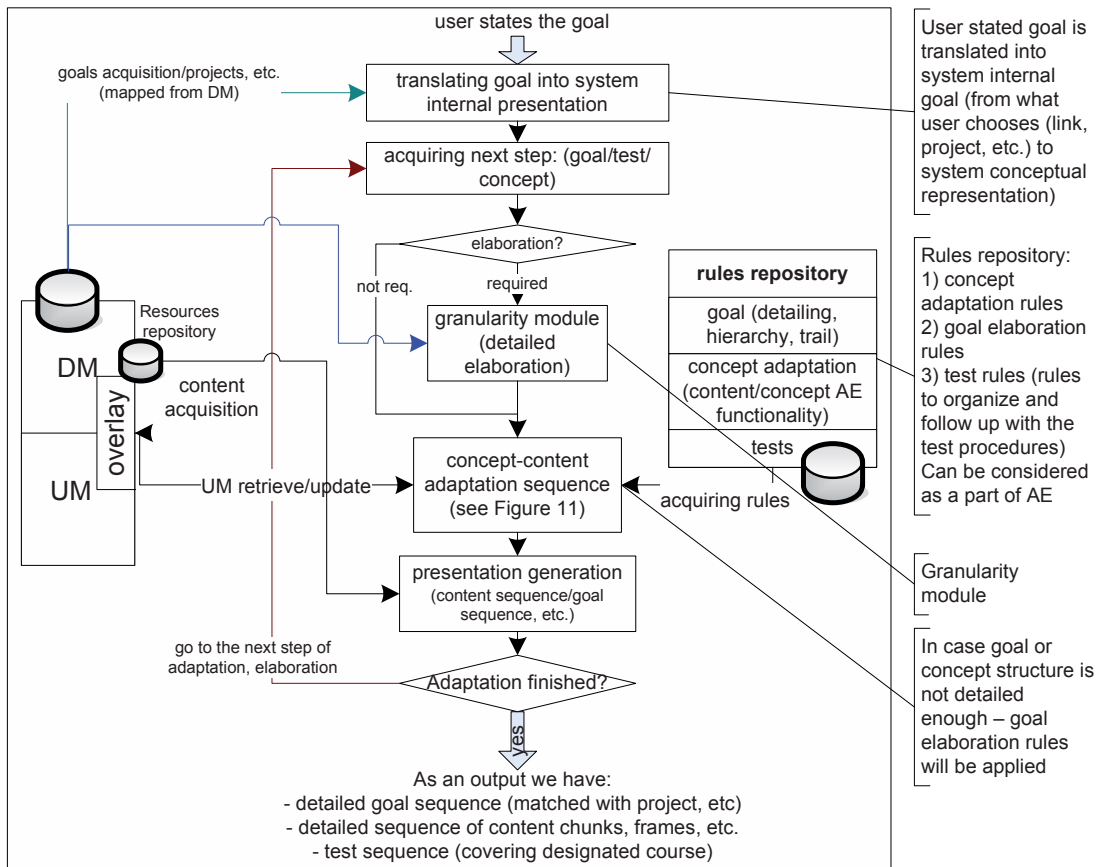


Figure 15: Use-Case: Adaptive Course flowchart

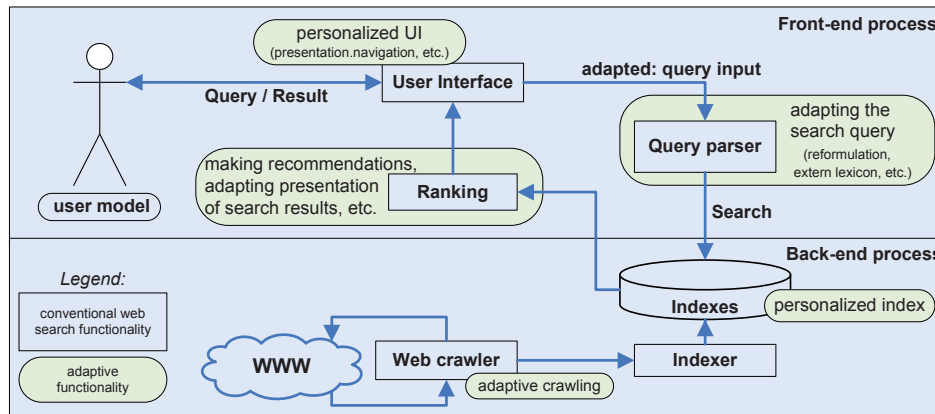


Figure 16: Web Search Schema

- The user states the goal thus formulating a new search query, which can be considered as stating or choosing a particular concept (set of concepts) to follow in AHS. It can be interpreted and aligned with DM (availability of concepts, concept structures and sequences, etc.) and UM (considering user competencies, preferences, experience, etc.) thus re-formulating and refining the search query (matching it with the common lexicon or using semantically related terms).
- The Domain Model is defined by the search index, representing keywords used to facilitate fast and reliable

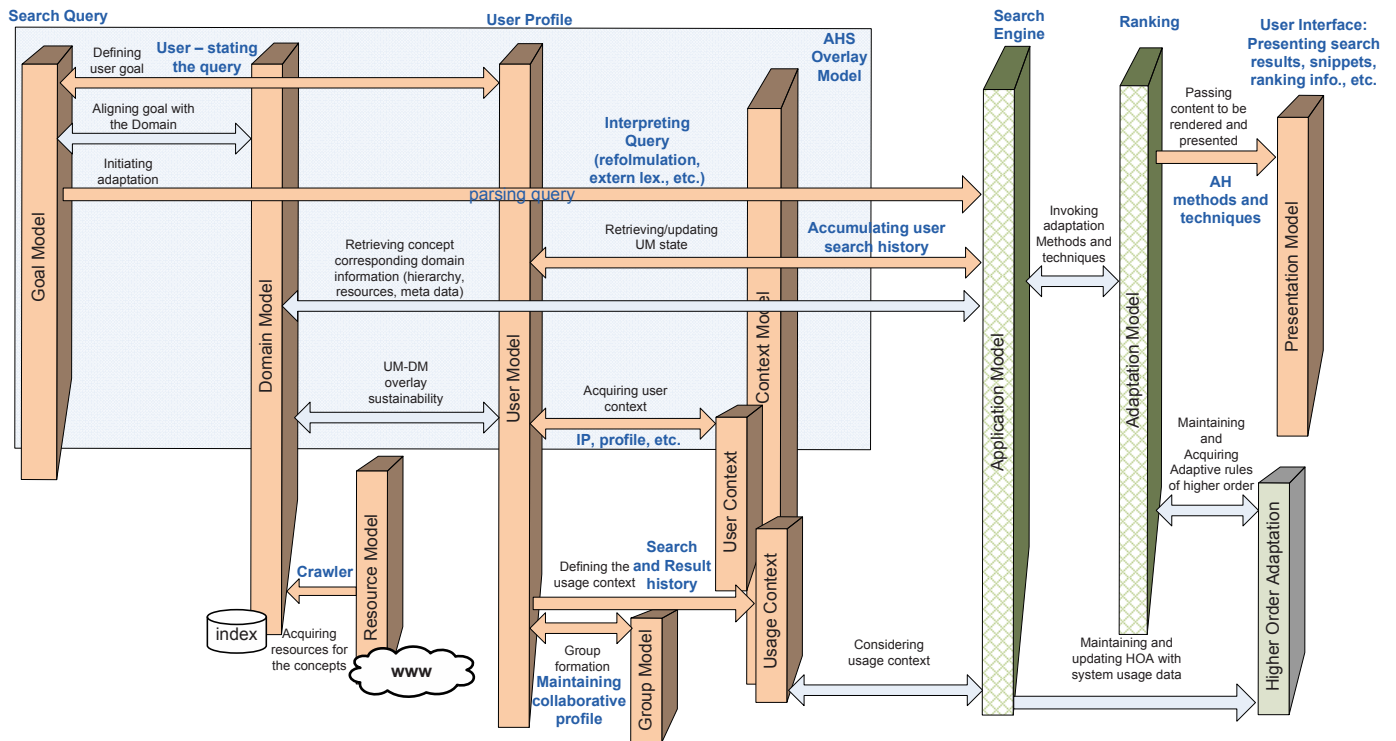


Figure 17: Search Compliance With Generic Adaptation Process

information retrieval, which is acquired from the *Resource Model* (and essentially WWW). The index information is obtained from WWW by means of crawling which is similar to the process of resolving content information of a concept in AHS.

- The *Context Model* defines user and usage context properties such as IP address, user profile/stereotype, or search and result histories accordingly.
- The *Group Model* refers to maintaining a collaborative profile of the user or stereotyping search results by location or user age group and gender, which later can be used to rank and recommend results.
- Retrieving and updating UM refers to *storing and accumulating UM search history* which can be used to reformulate queries or retrieve personalized results.
- *Application and Adaptation Models* may refer to the Search Engine and Ranking mechanisms, however it may not be entirely clear how to distinguish some particular parts of those. Here we would refer to the Adaptation Model for Ranking, since they both to some extent perform adaptation of the results. The Application Model then serves as the core of the system: coupling other layers and dispatching information in AHS or performing a search as the Search Engine.
- *The Presentation Model* renders search results and presents a ranked result list, snippets, additional rank information, groups result (e.g. pictures, videos, text, etc.).

To make this use-case more convincing and outline the further development of the framework compliance schema we matched the building blocks and described above adaptation process we continue with more elaborate Recommender Systems example.

6.3 Recommender System Use-Case

Figure 18 presents the picture of compliance of a recommendation (Ricci et al. 2011) and an overlaying GAP sequence chart. Here the generic process chart is constructed by coupling the layers of a general purpose AHS as in the case before, however the purpose is different. We consider recommendation steps to a single layer or a transition in the system and discuss this compliance presented in the figure further in this section. Though we are facing certain issues distinguishing Recommendation Engine functionality, in particular filtering and ranking mechanisms (in this respect Application Model (AM) and Adaptation Model/Engine (AE) can be treated accordingly) we could align recommendation and describe its functionality (in terms of aforementioned models) with GAF terms.

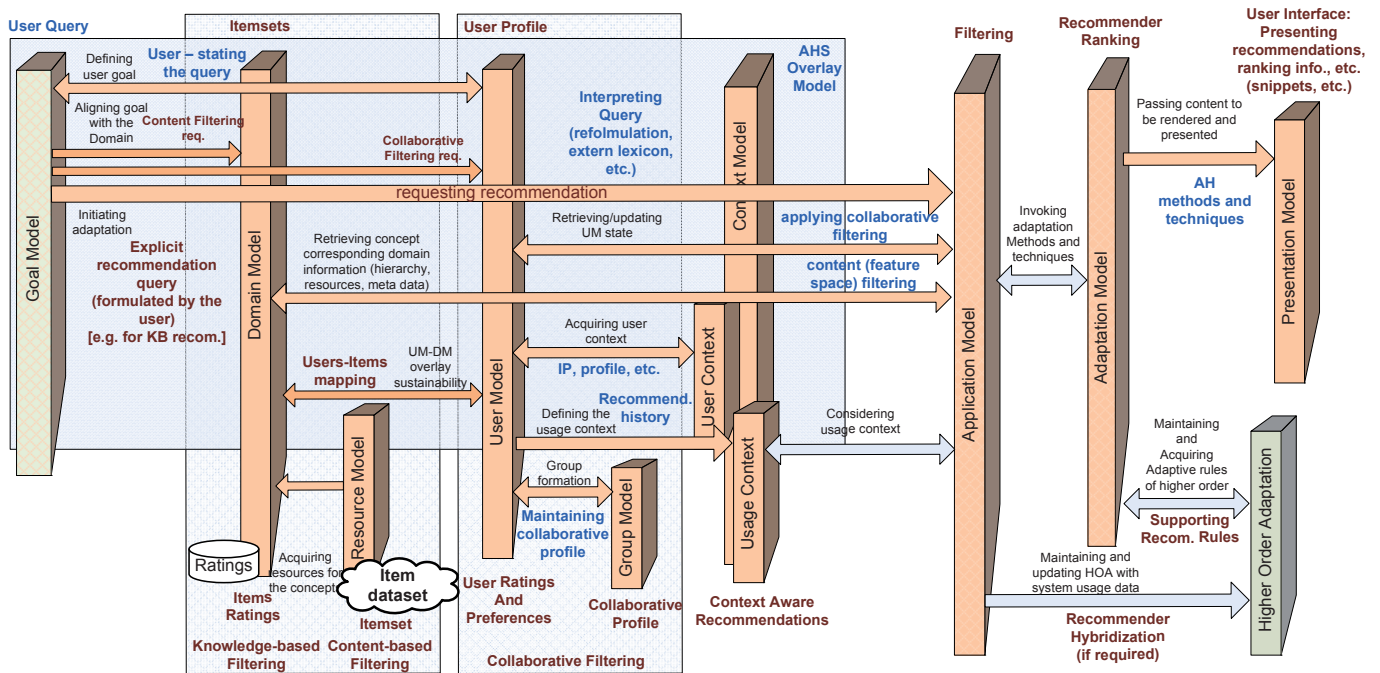


Figure 18: Recommender System Compliance With Generic Adaptation Process

Hereafter we are going to summarize the compliance of the recommendation process with the reference structure of AHS and explain the building blocks and interactions presented in figure 18:

- The User states the goal thus formulating a new recommendation query for which inferences over the user preferences are made (which is particularly interesting in knowledge-based recommender systems). This step can be considered as stating a goal or choosing a particular concept (set of concepts) to study/visit in an AHS. The goal can be interpreted and aligned with DM (availability of concepts, concept structures and hierarchies, etc.) and UM (considering user competencies, preferences, experience, interests, etc.). The same way the recommendation query can be reformulated, refined or aligned to match with the related user preferences using semantically related terms to get better recommendations by inferring closely related items (e.g. name of a favorite film director relates to a certain movie genre); UM is significantly important in collaborative filtering and corresponding recommender systems.
- The Domain Model (DM) is defined by the knowledge structure of the domain, representing keywords and terms together with the relationships which can be used to facilitate fast and reliable information retrieval and filtering of the concerned itemspace, where the resources are defined by RM. On the other hand DM may represent the feature space in case of content-based recommendations where content is again retrieved from RM. It happens more often in the recommender systems that the domain is represented by an ontology (Cantador et al. 2008) in order to facilitate more elaborate reasoning over the items relationships and

present more accurate recommendations.

- The Context Model defines user and usage context properties such as IP address, time, other activities of the user, etc. Modelling Context gives a possibility to consider context-aware recommendations and adaptation, both from the usage and the user point of view.
- The Group Model (GM) refers to maintaining a collaborative profile of the user(s) or stereotyping filtered results by location or user age group and gender, which later can be used to rank and recommend results for a particular user or mediate user models associated with different groups (Berkovsky et al. 2008). The Group Model in general may represent and serve heterogeneous user groups by looking up commonalities in profiles to form groups or similarities in the group system usage (usage patterns) to recommend next best items both in the context of recommender and adaptive system/application for the users within the same group.
- Retrieving and updating UM refers to storing and accumulating users' rankings and recommendation history which can be used to reformulate system queries or retrieve personalized recommendations by finding similar patterns in the users' system usage.
- Application and Adaptation Models may refer to the Recommender Engine involving Filtering and Ranking mechanisms, however it may not be entirely clear how to distinguish some particular parts of those between layers of AHS. Here we would refer to the Adaptation Model for ranking and recommendation rules, since they both to some extent perform adaptation of the results. The Application Model then serves as the core of the system: coupling other layers and dispatching information in AHS and Recommender respectively and performing a corresponding filtering method retrieving information from UM and DM for collaborative and content-based filtering respectively. Usually search and recommender engines are more robust and flexible for introducing or discovering new rules compared to the Adaptation Engines. However the rule systems which are conventionally used in AHS can easily facilitate reasoning in recommender systems (e.g. ECA type of rules to determine static recommendation filters such as gender or location aware, or at the same time serve as the basis for semantic reasoning to look up for a related concepts that the user might be interested in). AHS will also provide the so-called *higher-order adaptation* capabilities. They will monitor the user's behaviour also to adapt the adaptation behaviour by discovering new or refining old rules.
- The Presentation Model renders recommendation and adaptation results in such a way that a recommendation list is presented in the form of a ranked list, snippets, additional ranking information, result groups, etc. By applying AH presentation and navigation techniques (Knutov et al. 2009) we may present not only ranked or sorted lists, but the whole new spectrum of interaction becomes available to enhance user experience in recommender systems (e.g. (de)emphasizing results in the list, summarizing results, navigating through the list, presenting contextual and non-contextual links, annotating, etc.)

7 AHS Evolution, Layers and Structure

In Figure 19 we show the evolution of the Hypertext reference models, from Hypertext to Adaptive Hypermedia to the new Generic Adaptation Framework (GAF) which encapsulates most recent developments in AH and adjacent fields. A Dexter model description can be found in (Halasz and Schwartz 1994), so here we would like to concentrate on the adaptation features evolution and outline major differences of the emerging GAF system and its compliance with the GAP.

The AHAM (De Bra et al. 1999) reference model could be considered as an adaptive extension to the Dexter model. Here are the major points of AHAM:

- Any AHAM application must be based on a *Domain Model (DM)*, describing how the information content of the application or *hyper-document* is structured (using a conceptual representation of knowledge);
- A *User Model (UM)* must be devised and its sustainability should be maintained representing preferences, knowledge, goals, navigation and other relevant user aspects;

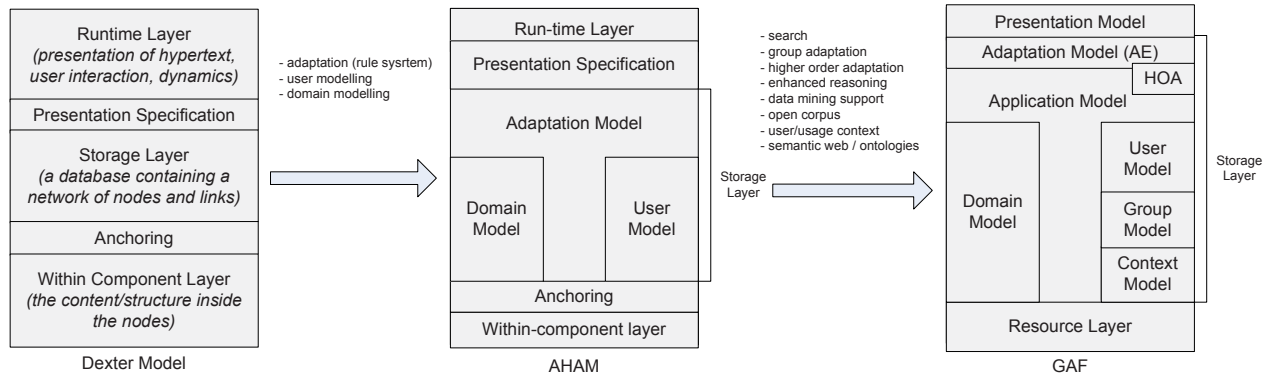


Figure 19: From Dexter, through AHAM, to GAF

- The presentation of content and link structure must be adapted to the user's behaviour as well as to the user's knowledge and interest. Thus an *Adaptation Model (AM)* should be defined consisting of adaptation rules. The rules define both the process of generating the adaptive presentation and that of updating UM.

In AHAM the Dexter Storage layer was split to support Domain and User modelling in order to facilitate adaptation to user attributes based on the conceptual structure of the domain, represented by the concept-link structure. The Adaptation Model (AM) encapsulates the Adaptation Engine (AE) functionality, the rule system performing adaptation based on the value of UM attributes.

Moving towards GAF, we enhance adaptation capabilities and include new methodologies and techniques, facilitating more elaborate adaptation. In figure 19 (to the right) we presented an extended draft architecture of GAF and briefly outline the enhancements (compared to AHAM). In Figure 20 we decouple GAF blocks in a way we used to describe the adaptation process and put them in a hierarchical structure to show what these major enhancements are and where they belong in the GAF block structure.

- *Ontologies* representing the conceptual structure of the application domain can be used to provide interoperability within and between adaptive applications. In general representing the conceptual structure of the domain, ontologies are the core part of the overlay model, from which the conceptual structure is inherited in multiple layers such as DM, UM and Goal Model. To some extent we can also include the Context model in case we consider each concept context while accessing it. If we want to start combining adaptation from different applications taking advantage of what one AHS has learnt about the user in another AHS the meaning of the concepts must be agreed upon, therefore, instead of arbitrary conceptual structures adaptive applications should be based on ontologies. A Domain Model based on an ontology makes interoperability feasible (Aroyo et al. 2004).
- *Open corpus adaptation* which is increasingly considered in adaptive applications (Brusilovsky and Henze 2007) is scrutinized by introducing Resource-Domain models interaction. This is where resources come from search results in dynamic object repositories or from a Web search engine (see section 6.2).
- *Context Awareness* allows system and application to be decoupled from the existing environment, and makes them more sensitive to adapt in many other ways rather than through a set of predefined rules. We consider usage and user context for GAF: both capturing the context of user behaviour and system usage, allowing to adapt to user and concept contexts (e.g. environment settings).
- *Data Mining* techniques applied in Information Retrieval settings can be used to cluster users into groups based on their navigational patterns or capturing long term effects of adaptation rules and their automatic update. It helps with information search (including web search) in open corpus domain and assists in identifying usage context and correlation between user context and adaptation.
- *Group-based adaptation* will extend the adaptation by taking group models into account. It determines partitioning of the users into groups and adapting to the group model.

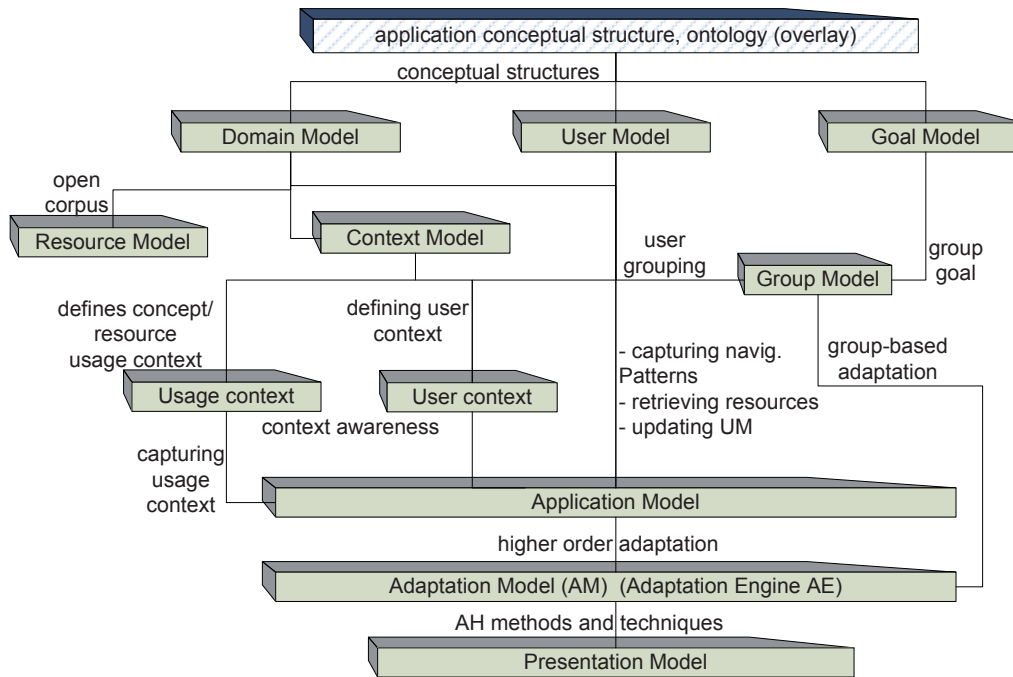


Figure 20: GAF layers hierarchy

- *Higher order adaptation* monitors the user's behaviour to adapt the adaptation behaviour (in particular to refine existing or create new rules, which is only possible in combination with machine learning approaches).

8 Conclusions and Future Work

In this paper we defined and elaborated various aspects of the *GAP*, introduced its model: flowchart and (informal) sequence chart. To comply with the layered model we anticipate that the aforementioned process structure will influence the layered composition of the AHS in such a way that the process defined by the system engineer will partially drive the order of the layers of such AHS and define important inter-layer transitions. At the same time we anticipate that the defined processes together with the reference model (e.g. the well known AHAM model or the new GAF model) may serve as a foundation for the system design, defining not only the system components but the system *functionality flow* as well, or even deviate into a separate branch of so-called *process-driven* architectures in the AH field. Moreover the formalized process driven approach gives more insight into AH development methods and unifies this development approach and system organization as it was first mentioned in (Koch 2001).

We have started elaboration of the generic layered structure of AHS in (Knutov et al. 2009) and then gave a first look at this kind of layered generic structure (Knutov 2009). This led to a process-oriented view of the generic layered AHS which was presented in this paper. Based on the research done in (Knutov et al. 2009) we proposed the generic flowchart compliant with the most popular AHS. The course adaptation flowchart in this case serves as a proof of an adaptation process concept (Figure 15), web search process (Figure 17) shows the compliance of the generic adaptation sequence and a conventional web search, opening a new possibilities to Open Corpus adaptation and confirms the generic nature of the GAF framework. Recommender systems use-case (Figure 18) extends the coverage of GAF and brings recommendations into place. And finally considering the layered (de)composition of an adaptive system we present a conceptual view of a *adaptation process* (Figure 14).

In the future we plan to extend the adaptation process sequence: elaborate the process description, particularly inter-layer interaction, sustaining the generic approach, emphasize the interoperability of a new AH developments (Ontologies, Open Corpus, Higher-Order Adaptation etc.) and the conventional AH approaches. This may lead

to describing interoperable and alternative interaction in the system thus representing a generic view of an AH framework which includes all possible variations of adaptation functionality and techniques.

9 Acknowledgements

This work is supported by the **NWO project GAF: Generic Adaptation Framework**.

References

- Aroyo, L., De Bra, P., and Chepegin, V. (2004). “Semantic Web-based Adaptive Hypermedia”. In *WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, volume 105 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Balík, M. and Jelínek, I. (2007). “Towards semantic web-based adaptive hypermedia model”.
- Berkovsky, S., Kuflik, T., and Ricci, F. (2008). Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286.
- Bordegoni, M., Faconti, G., Feiner, S., Maybury, M. T., Rist, T., Ruggieri, S., Trahanias, P., and Wilson, M. (1997). “A standard reference model for intelligent multimedia presentation systems”. *Comput. Stand. Interfaces*, 18(6-7):477–496.
- Bra, P. D., Smits, D., and Stash, N. (2006). Creating and delivering adaptive courses with aha! In *EC-TEL*, volume 4227 of *Lecture Notes in Computer Science*, pages 21–33. Springer.
- Brusilovsky, P. (1996). “Methods and Techniques of Adaptive Hypermedia”. *User Modeling and User Adapted Interaction*, 6(1-2):87–129.
- Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110.
- Brusilovsky, P. and Henze, N. (2007). “Open corpus adaptive educational hypermedia”. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 671–696. Springer.
- Cantador, I., Bellogín, A., and Castells, P. (2008). A multilayer ontology-based hybrid recommendation model. *AI Commun.*, 21(2-3):203–210.
- Carro, R. M., Ortigosa, A., and Schlichter, J. (2003). Adaptive collaborative web-based courses. In *Proceedings of the 2003 international conference on Web engineering, ICWE’03*, pages 130–133, Berlin, Heidelberg. Springer-Verlag.
- Conlan, O. and Wade, V. P. (2004). “Evaluation of APeLS - an adaptive elearning service based on the multi-model, metadata-driven approach”. In *AH*, volume 3137 of *Lecture Notes in Computer Science*, pages 291–295. Springer.
- De Bra, P. and Calvi, L. (1998). “AHA! an open adaptive hypermedia architecture”. *The New Review of Hypermedia and Multimedia*, 4:115–140.
- De Bra, P., Houben, G.-J., and Kornatzky, Y. (1992). “An Extensible Data Model for Hyperdocuments”. In *ECHT 92, Proc. of the 4th ACM Conference on Hypertext*, pages 222–231. ACM.
- De Bra, P., Houben, G.-J., and Wu, H. (1999). “AHAM: a Dexter-based reference model for adaptive hypermedia”. In *Hypertext ’99: Proceedings of the tenth ACM Conference on Hypertext and Hypermedia : returning to our diverse roots*, pages 147–156, New York, NY, USA. ACM.

- De Bra, P., Smits, D., and Stash, N. (2006). “the design of aha!”. In *Hypertext’06: Proceedings of the 17th ACM Conference on Hypertext and Hypermedia*, pages 133–134, New York, NY, USA. ACM.
- de Vrieze, P. T. (2006). “*Fundamentals of Adaptive Personalization*”. PhD thesis, Radboud University Nijmegen, Nijmegen, The Netherlands.
- Groth, P., Deelman, E., Juve, G., Mehta, G., and Berriman, B. (2009). “Pipeline-centric Provenance Model”. In *Proc. of WORKS ’09: the 4th Workshop on Workflows in Support of Large-Scale Science*, pages 1–8, New York, NY, USA. ACM.
- Halasz, F. and Schwartz, M. (1990). “The Dexter Hypertext Reference Model”. In *Proc. of the Hypertext Standardization Workshop by (NIST)*. NIST.
- Halasz, F. and Schwartz, M. (1994). “The Dexter Hypertext Reference Model”. *Commun. ACM*, 37(2):30–39.
- Knutov, E. (2009). “GAF: Generic Adaptation Framework”. Hypertext 2009 SRC poster.
- Knutov, E., De Bra, P., and Pechenizkiy, M. (2009). “AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques”. *New Rev. Hypermedia Multimedia*, 15(1):5–38.
- Knutov, E., De Bra, P., and Pechenizkiy, M. (2010). “Provenance Meets Adaptive Hypermedia”. In *Hypertext ’10: Proceedings of the 21th ACM Conference on Hypertext and Hypermedia*, pages 93–98, New York, NY, USA. ACM.
- Koch, N. (2001). “*Software engineering for adaptive hypermedia systems*”. PhD thesis, Ludwig-Maximilians University of Munich, Munich, Germany.
- Koch, N. and Wirsing, M. (2002). “The Munich Reference Model for Adaptive Hypermedia Applications”. In *Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference, AH 2002, Proceeding*, volume 2347 of *Lecture Notes in Computer Science*, pages 213–222. Springer.
- Ram, S. (2006). “Data Provenance”. Project accomplishments, The University of Arizona, Advanced Database Research Group, The University of Arizona.
- Ram, S. and Liu, J. (2006). “Understanding the Semantics of Data Provenance to Support Active Conceptual Modeling”. In *Proc. of ACM-L: 1st Int. Workshop on Active Conceptual Modeling of Learning*, pages 17–29, Berlin-Heidelberg. Springer.
- Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer.
- Saksena, M. and Cristea, A. (2006). “Towards more efficient generic semantic authoring for adaptive hypermedia”. In *APS ’06: Proceedings of the joint international workshop on Adaptivity, personalization & the semantic web*, pages 11–20, New York, NY, USA. ACM.
- Schneider-Hufschmidt, M., Ktihme, T., and Malinowski, U., editors (1993). “*State of the Art in Adaptive User Interfaces*”, pages 13–48. North Holland, Amsterdam, the Netherlands.
- Seefeldter de Assis, P. and Schwabe, D. (2004). “A general Meta-Model for Adaptive Hypermedia Systems”. In Bra, P. D. and Nejdil, W., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 433–436. Springer Berlin / Heidelberg.
- van der Sluijs, K. and Höver, K. M. (2009). “Integrating Adaptive Functionality in a LMS”. *International Journal Emerging Technologies in Learning*, 4(4):46–50.