Adaptive Hypermedia Systems Analysis Approach by Means of the GAF Framework

Evgeny Knutov, Paul De Bra, and 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 concentrating on adaptive guidance of links between domain concepts with lots of custom developments and ad-hoc implementations. Here we consider a formalization approach to AHS composition and design by defining building blocks' interfaces and presenting corresponding dependencies by means of the GAF framework. This helps to identify system design guidelines and start building adaptive system from scratch as well as analyze adaptive system behaviour, architecture and risks involved.

1 Introduction

Since the most cited Adaptive Hypermedia (AH) model AHAM [1] new terms, definitions and models have been introduced and realized in prototypes. Most AH models focus on a layered architecture and concentrate on adaptation to the linking and navigation between concepts of a domain. With the exploding popularity of the Web searching rather than linking, or Recommender systems (RS) to rank relevant content and provide personalized information the area of AHS has gained a lot. The Generic Adaptation Framework (GAF)[1] research project aims to develop a new reference model for the adaptive hypermedia research field. The new model considers new developments, techniques and methodologies in the areas of adaptive hypermedia and adjacent fields. Besides GAF concerns the detailed system analysis in terms of AHS building blocks, connections and dependencies, approaches that can be used to implement such a system.

GAF conceptual scheme of the layered structure is presented in Figure 1. It aligns the order of the layers in the system according to the classification of AH methods and techniques [5]. 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 idea behind this (adaptive eLearning application, Recommender System, etc.). 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 scenario (partially considered as use-cases in [4]) that will partially determine the layer arrangement and to some extent will define the mandatory and optional elements and drive the system design.

[1] `http://www.win.tue.nl/~eknutov/gaf.html`

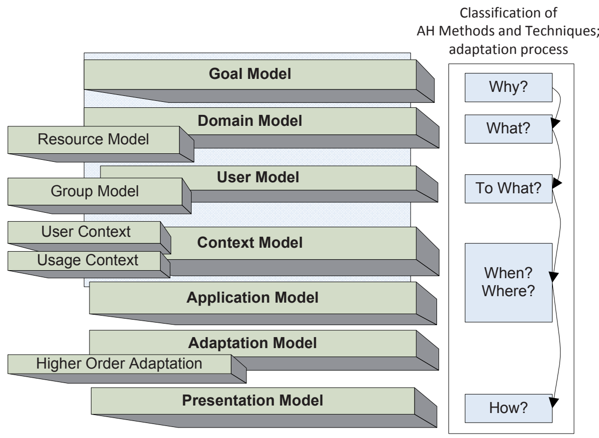

Fig. 1. Conceptual scheme of GAF layered structure

2 AHS Analysis Approach

As thoroughly investigated in [7] the evaluation of AH systems plays an important role. The described layered evaluation provides the description of the system functionality and helps to solve many related problems. In our work we consider a more formalized and specific system analysis approach by taking up systems' block composition scenarios, interfaces. Thus we define dependencies between models, methods they use to communicate with each other and particular implementations (based on usage scenarios). As a reference we took the approach from [3]. The main steps of such an analysis are presented in Figure 2. By scenarios here we mean framework use-cases (adaptive search, adaptive eLearning, recommender system, etc.), mostly covered in [4]. These scenarios are represented by 'sequence charts' and are constructed using GAF layers. We also consider system specific aspects and AHS building blocks composition which impacts the system architecture, such as event-driven system or service oriented or these two together.

As a result of this approach we would have elementary base concerns of AHS, which would explain mandatory and optional building blocks of the system, trade-off available, mostly concerning optional elements of AHS, and the dependencies involved presented as table. We will elaborate the approach further and explain it through the example of the Domain Model (DM).

3 AHS Models Analysis Approach: DM example

Hereafter we elaborate the analysis approach and consider the AHS DM. In Figure 3 we show an example of DM interface dependencies. Analyzing it down further we comprise the dependency table of building blocks' interfaces (such as Domain, Use, Resource, Context models), scenarios of how these models are used and which type

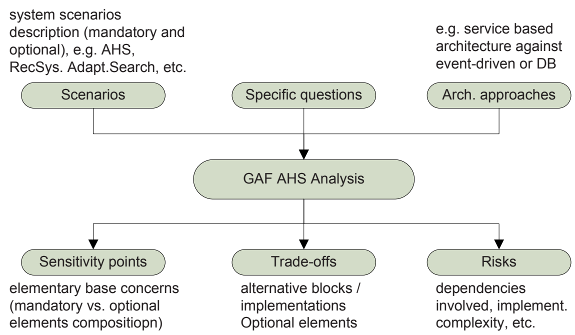

Fig. 2. AHS analysis approach

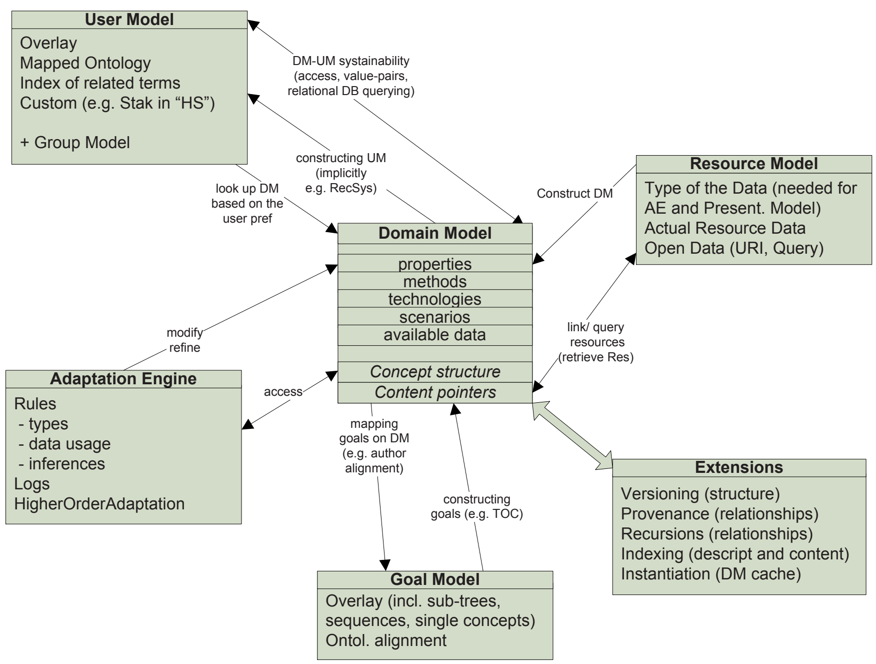

Fig. 3. Domain Model interface dependencies

of system is being described (AHS, Adaptive eLearning, Recommender System, etc.), possible technologies to implement it (Data Bases, OWL ontologies for semantic web enabled systems, TF-IDF index for search, etc.). As a result we'll have a detailed picture of the system components, evaluated against the reference model (GAF), which will help to identify all pros and cons.

Considering any arbitrary DM properties and interfaces we analyze them against the following properties and methods of the reference structure (see Figure 4 for de-

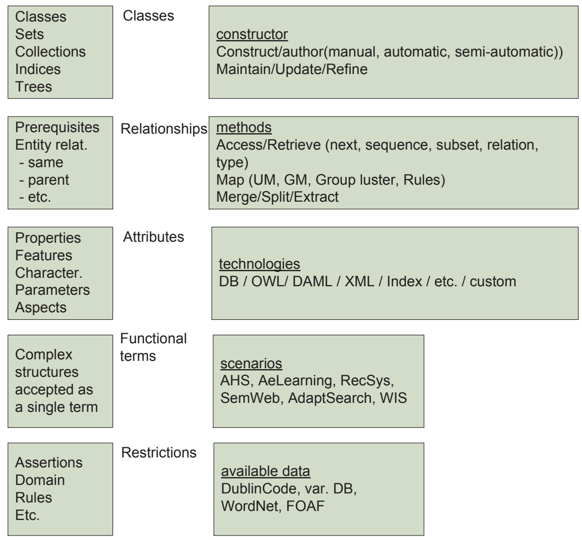

Fig. 4. Domain Model abstraction class

Table 1. partial GAF blocks high-level dependencies: DM example

DM properties and methods	Scenario	Resource Model	Adaptation Engine	User Modelling
concept tree	conventional AHS eLearning	content pages/frames	ECA reasoning, prerequisites relations	UM overlay
feature space	recommender system	datasets	promotions and ranking mechanisms	implicit user profiling
index	adaptive search	WWW	ranking	implicit user profiling

tails). The major division here concerns methods and properties of the abstract Domain Model class. Further we distinguish classes (like sets or collections of concepts or concept maps, indices, trees, etc.), relationships (which are conventionally constituted by the ontology relationships), attributes of the concepts (e.g feature space, properties, characteristics, etc.), then functional terms which are denoted by complex structures usually treated as a single term, and restrictions defined by assertions or some specific domain rules.

Methods can be defined by constructors used to author DM as well as refine, maintain or update it. Major DM methods describe the access and retrieve procedures mainly called by User Model (UM), Resource model (RM) and Adaptation Engine (AE) to access the conceptual structure and query corresponding content. We also define mapping methods which are used to maintain structure sustainability especially in overlay type of models or ontology mapping for instance. These mappings (or alignments) can be done between DM and User, Goals, Groups models and Rules sets. Additionally we have methods to merge, split and extract sub-models of DM, which can be used in distributed domain modelling or open corpus adaptation.

DM scenarios describe the system behaviour in terms of functional flow and user interaction. We have described most prominent use-cases of such a framework compliance with different types of systems in [4]. Thus the DM usage in different cases could be analyzed against these reference scenarios.

Finally we have a number of particular technologies to work with DM and associated or cross-technology data available to start modelling (e.g Dublin Core to devise adaptive eLearning application or a dataset feature list to devise recommender system or adaptive search portal). This may remind us of the UML notion used in [6] to formalize the AHS modelling, however we define more strict dependencies in the GAF formalization through defining interfaces, methods and scenarios, besides we use it to analyze system, identify alternatives and be able to compare and assess other systems in terms of the GAF framework. Table 1 presents high-level dependencies between DM properties and methods, scenarios and other AHS' models. This is just to give an idea of our approach, ideally these dependencies would be described in meticulous details, parametrizing abstract DM interfaces and to some extent show concrete technology or implementation approach for each of these models' interfaces.

4 Summarizing Implications of the Analysis Approach

Here we would like to summarize the major implications of our approach and anticipated benefits.

- Reference structures — being a reference model GAF and detailed dependencies of its layers will serve as an ideal starting point for AH system designers and researches in the field.
- Complexity and Performance — defining a number of dependencies and known technologies would give an impression of the system complexity.
- Compatibility and Compliance — compliance description ([4]) provides the description of use-cases and application scenarios of the GAF framework.
- Modifiability — trade-off between blocks or modules' alternatives will show the modification possibilities, or further system extensions.

5 Conclusions and Future Work

The coming years will bring more use-cases of how AHS can provide adaptation and personalization, what techniques will be introduced, and what research areas will introduce new technologies in its evolution. So far a study of existing adaptation and personalization approaches was done to comply with the layered structure of adaptive information systems, which raised the problem of system composition and design analysis. We try to solve this problem using a classical software architecture analysis approach extending it with adaptation framework specific questions and interface dependencies in order to meticulously analyze any adaptive system in terms of the GAF framework.

At the same time evaluating the proposed general-purpose AHS architecture (GAF framework) against recommender systems [2] has shown that the GAF architecture is sufficiently generic to accommodate the description of different personalization approaches including recommenders, as well as provide the flexibility of both AH and RS in one go by building a custom system with the GAF building blocks. The real though not very meticulous case study has proven our points. It has given us new challenges to investigate the applicability of new approaches, as well as new developments in adaptive information systems which will allow to decide on the system composition at the implementation level and this is where one would need the AHS analysis.

6 Acknowledgements

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

References

1. P. De Bra, G.-J. Houben, and H. Wu. AHAM: a Dexter-Based Reference Model for Adaptive Hypermedia. In *Hypertext*, pages 147–156. ACM, 1999.
2. J. Hannon, E. Knutov, P. D. Bra, M. Pechenizkiy, K. McCarthy, and B. Smyth. Bridging Recommendation and Adaptation: Generic Adaptation Framework - Twittomender compliance case-study. In *to appaear in Proc. of the 2nd International Workshop on Dynamic and Adaptive Hypertext (DAH'11)*, 2011.
3. R. Kazman, M. Klein, and P. Clements. ATAM: Method for architecture evaluation. Technical Report ESC-TR-2000-004, Carnegie Mellon, Software Engineering Institute, 2000.
4. E. Knutov, P. D. Bra, and M. Pechenizkiy. Generic adaptation framework: a process-oriented perspective. *J. Digit. Inf.*, 12(1), 2011.
5. E. Knutov, P. De Bra, and M. Pechenizkiy. AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Rev. Hypermedia Multimedia*, 15(1):5–38, 2009.
6. N. Koch. *"Software engineering for adaptive hypermedia systems"*. PhD thesis, Ludwig-Maximilians University of Munich, Munich, Germany, 2001.
7. A. Paramythis, S. Weibelzahl, and J. Masthoff. Layered evaluation of interactive adaptive systems: framework and formative methods. *User Modeling and User-Adapted Interaction*, 20:383–453, December 2010.