Flexible Adaptivity in AEHS Using Policies

Arne W. Koesling (1,2), Daniel Krause (1), and Eelco Herder (1)

(1) L3S Research Center,
Appelstr. 9a, 30167 Hannover, Germany
(2) Peter L. Reichertz Institute for Medical Informatics
University of Braunschweig-Institute of Technology and Hannover Medical School,
Muehlenpfordtstrasse 23, 38106 Braunschweig, Germany
{koesling,krause,herder}@L3S.de

Abstract. In this paper, we show how existing adaptive educational hypermedia systems can be enhanced by policies. In traditional systems, the adaptation is based on predefined user and domain models and fairly restricted adaptation rules. Policies allow for sophisticated and flexible adaptation rules, provided by multiple stakeholders. We present the benefits and feasibility of the approach with AHA! as a hands-on example.

Key words: AHA, adaptive, hypermedia, trust management, policy

1 Introduction and Problem Statement

Lifelong learning has become an essential element of our everyday working life. As lifelong learning is associated with a large diversity in interests, knowledge and backgrounds, the one-size-fits-all approach of conventional learning management systems may not cater all individual user needs. The field of Adaptive Educational Hypermedia Systems (AEHS) is an active research area and the community has become aware of the benefits of adaptivity. Nevertheless, in contrast to commercial (non-adaptive) systems, AEHS are often (prototypic) systems that are used by a small audience [10]. One of the reasons for this is the use of hand-tailored, application-specific models of the user and the domain, and a limited, predefined set of adaptation rules. Interoperability and flexible rules that allow for conflicting statements would allow adaptive systems to benefit from user profile information from other systems. Furthermore, authoring courses in adaptive systems can be a complex task for course designers. Policy languages, together with engines that interpret the policies, offer an easy-to-integrate solution for this problem. Depending on the language used, policies can be used for negotiations and for access control and explanations. This does not only allow for resolving conflicting statements, but also provides means for making a system scrutable. In this paper we describe our approach and discuss the benefits and feasibility. Section 2 shows a motivation scenario. In Section 3 we give a short overview of the features of policy languages. Section 4 shows the detailed benefits for AEHS and presents a hands-on example for AHA!. The paper ends with a section on related work and some concluding remarks.

2 Motivation Scenario

In the following scenario we demonstrate how an adaptive system may benefit from policies. John is a student in economics, with a minor in German. John performed several courses at different universities, making use of several AEHS. John wants to subscribe to a course in 'Advanced Business German' at a different university than where he studies. The course designer has defined as a prerequisite that the learner has to have at least 'Fundamentals of German Grammar'. John's ePortfolio does not contain the required credential, but has instead the credential 'Intermediate German 2' from his home university, which is actually better than the required credential.

When John applies for the course, a negotiation process is started in the background. As John considers his learning knowledge sensitive, the AEHS uses policies for negotiating a certain level of trust between the learner and the system. John's computer sends credentials to proof that he is a student of the university and the server shows the university credentials and the university's privacy policy. Additionally, the server asks for the credential 'Fundamentals of German Grammar'. John's computer offers the credential 'Intermediate German 2'. The policy engine at the server requests an external competence map service that confirms that John's competence fullfils the minimum requirement. John is therefore able to join the course. As a certain level of trust has been established, John grants access to his ePortfolio, so the server is able to fetch John's learner preferences. A third-party server is used for translating John's university's proprietary ePortfolio format into an exchangeable format. These preference policies are used to negotiate the actual structure of the group course by matching it with his fellow students' profiles and the teacher's preferred mentoring style.

Policies can also contribute to the ongoing work on scrutability. *Explanations* allow for queries to the policy engine asking about details for certain decisions. At some point in the course, John notices that his peer students receive more contextual information. The system may explain that the text was omitted because of his high level knowledge in that field, as stated by his certificate. As a group discussion was planned on Friday afternoon, his second choice, he wants to know why the discussion has not been planned on Thursday morning. The system shows a trace of the reasoning on his own and his fellow learners' agendas.

3 Policies

A policy is generally understood as a statement that defines the behaviour of a system. Policies are intended to guide decisions and actions. In today's software systems, policies are primarily used for solving security and privacy concerns – such as controlling access to sensitive user data – and to model business rules. As an example, new customers of an online shop have to pay in advance, while regular costumers may be allowed to pay after delivery. In the scope of eLearning, similar policies would be possible, formulated in a logic-based format, that depends on the policy language used. Assuming that the course designer creates a 'beginner business german course' and wants to ensure, that each participant has sufficient knowledge, she may define a policy stating a prerequisite for the course that requires a certificate of 'german fundamental grammar' to be possessed by the learner requesting access. In the Protune policy language this may be written as:

(1) $isAllowedToSubscribe(LearnerName) \leftarrow$
(2) $credential(C),$
(3) $C.type : competence,$
(4) $C.owner : LearnerName,$
(5) $C.issuer :' UK\ National\ Language\ Institute',$
(6) $C.attribute : LanguageCertificate,$
(7) $LanguageCertificate.name :' Certificate\ Name',$
(8) $LanguageCertificate.value :' German\ Fundamental\ Grammar'.$

(9) $credential(\_) \rightarrow type : provisional.$
(10) $credential(\_) \rightarrow actor : peer.$
(11) $credential(C) \rightarrow explanation : "Credential"\ \&\ C\ \&\ "is\ sent".$

Similar to logic programs, the predicate 'isAllowedToSubscribe' in line 1 holds, if each statement in the lines 2-8 hold. Lines 9-11 represent Metarules defining additional statements about the predicates used. Line 9 states the type of the predicate. In this case, we assume that 'credential' is defined further outside the policy and associated with the action to send a credential to the communicating party. Line 10 tells that the party that needs to perform the action is the other peer and line 11 states an explanation for 'credential' described below. In [4] we examined the applicability of policies in open infrastructures for lifelong learning in general. In this paper, we gave an overview of both policy languages and policy engines, which are used to evaluate policies. The declarative nature of policy languages enables users to define *what* the system should do, and do not require knowledge about *how* the system realizes it. Policy engines like Protune [11], which operate on a rule-based policy language, have a declarative nature. In general, policy languages also provide reasoning support. In addition, Protune offers the previously described explanations. Users have the possibility to specifically ask *why* a certain answer was deduced or a decision was taken.

A remarkable feature of (Protune) policies is that they also allow for integrating external (environmental) information into the decision making process. By performing negotiations, the user can be asked for particular preferences, credentials, etc. Furthermore, integration of policies into existing systems can be easy as some policy engines can be called in a service-oriented manner.

4 Implementation of Policies in AEHS

In this section, we show how policies can be integrated into the well-known adaptive educational hypermedia system AHA! [3]. A lower-level integration does not allow for all of the issues which were included in the above scenario, but it can be reached in a less complex way by integrating queries to the policy engine into the AHA! adaptation rule conditions. If a user follows an AHA! course and the AHA! engine evaluating its adaption rules hits such a policy engine query, it just has to pass the query to a connected policy engine.

The policy engine can be connected to any information source – from specific user models to generic resources on the Internet – and will evaluate the query and reason over the existing policies. The result is passed back to the AHA! engine. Such a query that is sent e.g. to the Protune Policy Engine is similar to a query to a Prolog engine. The result can be a boolean value. However, if the query contains variables, the result will return those variables, bound to values. As an example, such a query could utilize external sources to verify if the provided resume fulfills the prerequisites of a learning resource or to check for the user's learning style, as specified in some local user profile. This approach gives a powerful means to the course designers, as they allow for considering some aspects outside the system that AHA! currently can't provide itself.

Additionally, such a query is built in a simple manner, due to the fact that, in most cases, it mainly consists of a meaningful term, like a method call from programming, which hides a complex set of policies and reasoning mechanisms within the policy engine. The policy sets can be edited by a rule designer to provide course authors with both predefined queries for instant use and powerful advanced functionality. By separating the rule design from course authoring, course authors do not need to know in detail how the adaptation functionality is technically accomplished.

5 Related Work

There are already many systems with their own, proprietary rule frameworks. These frameworks allow for complex adaption rules that provide many of the features presented in this paper and in [4]. As these rule systems have different emphasis, they are limited in their functionality and are strongly coupled to their systems. We are not aware of other approaches that rely on using advanced policies like we do here. General learning management systems – such as Moodle or Sakai – have very simple rule systems and offer no or only rudimentary features regarding adaptivity. However, there are already some efforts to enhance generic LMS like Moodle with adaptive functionality (see [5]). In an ambitious research project like Alfanet [8], which aims to make use of multi-agent technologies, learner preferences can be used to steer the behaviour of agents. Managements of interaction between multiple agents is already an objective in policy research. To the best of our knowledge, the idea of adding policies to e-learning in general and especially to AEHS is a largely unexplored area.

6 Conclusions

In traditional systems, the adaptation is based on predefined user and domain models, and fairly restricted adaptation rules. In this paper, we showed how existing adaptive educational hypermedia systems can be enhanced by policies. In particular in the field of lifelong learning, with many stakeholders and potentially many conflicting requirements and preferences, there is a need for adaptive systems that employ flexible rules and conflict resolution mechanisms. We discussed how the approach can be used for integration into the well-known AHA!-system. As a next step, we will conduct a qualitative study, at an Hannover-based institute for higher education, on what implicit policies authors, teachers and learners currently employ - and to what extent this is supported by their current systems. We also plan to implement and evaluate the use of policies within the TENCompetence project.

Acknowledgments. We would like to thank Daniel Olmedilla and Fabian Abel for contributing with suggestions, remarks and feedback. The work reported in this paper is partially funded by the European Commission in the TENCompetence project (IST-2004-02787).

References

1. Hauger, D., Koeck, M.: State of the Art of Adaptivity in E-Learning Platforms. In: ABIS 2007 - 15th Workshop on Adaptivity and User Modeling in Interactive System, pp. 355–360, Halle, Germany (2007)
2. De Bra, P., Aerts, A, Berden, B., de Lange, B, Rousseau, B, Santic, T, Smits, D, Stash, N.: AHA! The Adaptive Hypermedia Architecture. In: Conference on Hypertext and Hypermedia - Proceedings of the fourteenth ACM conference on Hypertext and Hypermedia, pp. 81–84 Nottingham, UK (2003)
3. De Bra, P., Smits, D, Stash, N.: Creating and Delivering Adaptive Courses with AHA!. In: Innovative Approaches for Learning and Knowledge Sharing - 1st European Conference on Technology Enhanced Learning, EC-TEL 2006, pp. 21–33 Crete, Greece (2006)
4. De Coi,J. L.,Kaerger, P., Koesling, A. W., Olmedilla, D.: Exploiting Policies in an Open Infrastructure for Lifelong Learning. In: 2nd European Conference on Technology Enhanced Learning (EC-TEL), volume 4753 of Lecture Notes in Computer Science, pp. 26–40, Crete, Greece (2007)
5. Tiarnaigh, M.: Adaptive Moodle. An Integration of Moodle (Modular Object Oriented Dynamic Learning Environment) with an AHS. Final Year Project, University of Dublin, May 2005.
6. Brusilovsky P.: Developing adaptive educational hypermedia systems: From design models to authoring tools. In: T. Murray, S. Blessing and S. Ainsworth (eds.): Authoring Tools for Advanced Technology Learning Environment. Dordrecht: Kluwer Academic Publishers, pp. 377–409
7. Brusilovsky P., Farzan R., and Ahn J.: Layered Evaluation of Adaptive Search. In: Proceedings of Workshop on Evaluating Exploratory Search Systems. At SIGIR 2006.
8. van Rosmalen, P., Brouns, F, Tattersall, C, Vogten, H, van Bruggen, J, Sloep, P, Koper, R: Towards an open framework for adaptive, agent-supported e-learning. In: International Journal of Continuing Engineering Education and Life-Long Learning, Vol. 15, No. 3/4/5/6. (2005), pp. 261–275.
9. Brusilovsky, P, Henze, N: Open Corpus Adaptive Educational Hypermedia. In: The Adaptive Web 2007, pp. 671–696
10. Paramythis, A. and Loidl-Reisinger, S.: Adaptive Learning Environments and e-Learning Standards. In: Electronic J. of e-Learning 2 (2), 2004, Paper 11
11. Bonatti, P. and Olmedilla, D.: Driving and monitoring provisional trust negotiation with metapolicies. In 6th IEEE Policies for Distributed Systems and Networks (POLICY 2005), pages 14-23, Stockholm, Sweden, June 2005. IEEE Computer Society