Phase II: Elaborate and Evaluate the Solution

Feedback-Driven Ontology Evolution

Elmar P. Wach
(Supervisor: Univ.-Prof. Dr. Dieter Fensel)

Hummelsbüttler Hauptstraße 43, 22339 Hamburg, Germany
Technikerstraße 21a, 6020 Innsbruck, Austria
wach@elmarpwach.com
elmar.wach@sti2.at

1. Problem Statement

1.1 Core dimensions of the field of research

Ontology Evolution, Ontology Versioning, Recommender Systems, Self-Adapting Information Systems

1.2 Unsolved problems and need for exploration

Recommender systems in e-commerce applications have become business relevant in filtering the vast information available in the Internet (and e-shops) to present useful search results and product recommendations to the user.

As the range of products and customer needs and preferences change – and they will change even more frequently – it is necessary to adapt the recommendation process. Doing that manually is inefficient and usually very expensive. Moreover, it is very difficult to predict improvements for given changes of the recommendation process.

Therefore, this research proposes an automated adaptation of the recommendation process by utilising semantic technology and processing user feedback.

The present research tackles an automated process for the first time (to the best knowledge of the author).

1.3 Solution

The shortcomings of a manual adaptation of the recommendation process based on user feedback are aimed to be solved with a system based on ontologies modelling the products offered in the e-commerce application and automatically evolving with processing user feedback. As the ontology describes the products formally, it offers a higher computability than conventional product descriptions and, hence, facilitates automated processing of information.

Implicit and explicit feedbacks provided via feedback channels are evaluated. Implicit feedback is given by the user as a side-effect of her usage behaviour, e.g. by clicking on the product recommended. Explicit feedback is extracted from the Web that could also deliver new information and aspects about the products offered. In order to focus this research on developing an adaptive ontology, the feedback is assumed to be given.

On a more abstract level, this research aims at realising an automated ontology evolution process based on feedback without a human inspection.

1.4 Previous approaches and room for improvement

Previous approaches in the topic of this research can be found in concepts for ontology evolution like formulated frameworks for ontology evolution, e.g. [Stojanovic, L. et al. 2002], [Klein, M. and Noy N. F. 2003], [Stojanovic, N. et al. 2003], [Haase, P. et al. 2005], [Noy, N. F. et al. 2006], [Konstantinidis, G. et al. 2007].

Due to the specific challenges of the present research like the automated ontology evolution process, none of the identified frameworks can be completely used as basis, e.g. all of the frameworks include a step for the human inspection of the ontology changes before they are executed. Nevertheless, it is advisable to respect single steps and aspects of the research done.

In the area of the use case, i.e. ontology-based recommender systems, there have been identified several approaches, e.g. Quickstep [Middleton, S. E. et al. 2001], Foxtrot [Middleton, S. E. et al. 2003], AVATAR [Blanco, Y. et al. 2005], ePaper [Maidel, V. et al. 2008], ISIS [Drachsler, H. et al. 2009], but only SERVOGrid [Aktas, M. S. et al. 2004] is similar to the present research. It recommends earth research resources conversationally. It utilises domain ontologies and content-based filtering. Implicit user feedback is processed in the domain ontology by calculating a ranking when querying a resource. Prioritised questions are presented to the user within a given threshold of difference.

1.5 Critical success factors, minimising risks, worst case strategy and outcome

- Processing user feedback: In case no clear distinction of the feedback channels is feasible, two independent feedback evaluation algorithms will be developed
- Feedback extracted from Linked Open Data (LOD): In case of an insufficient quality, additional restrictions for LOD feedback will be defined, e.g. reducing the importance of that feedback channel
- Automated ontology evolution process: In case this is not feasible respectively does not lead to increased key performance indicators, an “extended” administration interface will be developed. With this the manager can manually select options how the ontology should evolve

1.6 Other communities and their results

Similar approaches in the topic of this research can be found in the area of adaptive and self-management/ self-adaptation systems with an emphasis on ontology-based systems. Basically, the work focuses either on the system architecture [Zhou, Y. et al. 2007, etc.], e.g. reconfiguring of components, or on personalisation [Peyton, L. 2003, etc.], e.g. of websites.

[Tran, T. et al. 2006] have developed the Ontology for the Domain of Adaptive Systems (ODAS) that represents relevant adaptivity dimensions in terms of user, domain, task, environment, and system model. The focus of the present research, however, is to accomplish adaptivity by implementing a functionality that changes the ontology itself. [Kadlec, T. and Jelínek, I. 2007] discuss the Adaptation Anywhere and Anytime framework (A3) that creates an adaptive website based on ontological user profiles. With a user ontology manager, the web browser retrieves the user ontology, captures and saves changes from the user, and uploads it back to a user ontology server where the profile gets shared by merging different information sources. The adaptation is done on the bottom XSL/ XSLT layer based on XSL rules designed manually.

1.7 Hot or obsolete topic

Ontology evolution and versioning are researched well. Automatically processing of implicit and explicit user feedback and creating an adaptive ontology are relevant topics especially in feedback-based domains and applications like in e-commerce.

1.8 Impact of the potential solution on the community

Accomplishing an adaptive ontology based on feedback could mainly have two impacts on the community. Firstly, it signals that an automated ontology evolution is feasible and, thus, can induce further research in this direction. Secondly, the adaptation strategy developed can be utilised and adapted in similar efforts.

1.9 Application scenarios

Semantic (e-commerce) recommender systems, semantic search engines.

2. Main Questions of the Thesis

The main research question is: How can an adaptive ontology be created? This question can be split in two sub-questions:

1. How can user feedback be transformed into ontology input?

2. How can an automated[1] ontology evolution be realised?

These questions imply the development of the following:

1. An adaptation strategy that formulates the conceptual aspects of an adaptive ontology
2. An application that implements the adaptation strategy and creates evolved ontology and instance data (i.e. an adaptive ontology)

Most of times, ontologies in recommenders are used for the user profiling. The approaches research the impact on the recommendation result by using the different recommender types (i.e. content-based filtering, collaborative filtering, hybrid approaches) and mostly utilising domain and user ontologies, whereas the feedback gets processed in the latter one. There has been put less effort in researching the use of domain ontologies to achieve an adaptive user interaction.

3. General Approach

A generic adaptation strategy will be developed addressing the transformation of the feedback into ontology input (i.e. feedback transformation strategy) and how the changes will be executed in the ontology (i.e. ontology evolution strategy).

By following the principles of adaptive systems [Broy, M. et al. 2009], the adaptation strategy is implemented in a new adaptation layer (confer figure 1) consisting of components in which the user feedback gets transformed (i.e. feedback transformer) and the respective actions are decided and initiated (i.e. adaptation manager).

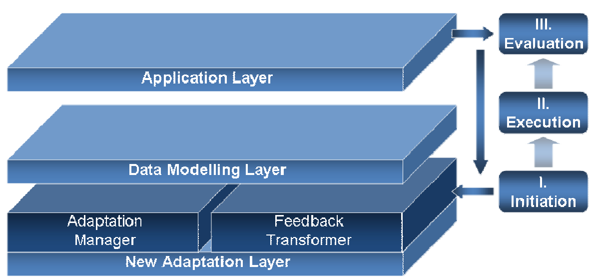

Fig. 1. Evolution cycle with a new adaptation layer

3.1 Research methods and path towards the achievement of the objectives

According to the research methodology by Hevner and March, type "Methods", this research is based on the following process:

---

[1] Without human inspection

- Discussing the pros and cons of existing approaches to ontology evolution and recommenders
- Formulating the research question
- Defining an adaptation strategy (as meta-model) and reference to other constructs and models (if applicable); it consists of a feedback transformation strategy for transforming user feedback into ontology input and an ontology evolution strategy for consistently changing the ontology
- Discussing intended applications and use cases
- Formulating conditions of applicability
- Implementing the strategy by programming an application consisting of the feedback transformation and the ontology evolution components creating evolved ontology and instance data (i.e. an adaptive system)
- Evaluating the solution with the use case provided with regard to the impact of the ontology evolution on the key performance indicators

3.2 Expected results

According to the research methodology by Hevner and March the expected result is a "method". The adaptation strategy should guide its users in identifying a solution for a similar research question of how to create an adaptive ontology that automatically processes user feedback.

The adaptation strategy can be seen as a new methodology for developing adaptive ontologies.

4. Proposed solution (approach, achieved results, to-dos)

The adaptation strategy defines how to achieve an adaptive ontology and has to answer the questions when (i.e. feedback transformation strategy, confer chapter 4.1) and how (i.e. ontology evolution strategy, confer chapter 4.2) the ontology has to change by evaluating the impact of the evolution in the precedent feedback cycle.

4.1 Feedback transformation strategy

In order to automatically process feedback, i.e. transforming it into ontology input, an adequate feedback transformation strategy has to be formulated and implemented. It has to allow for different feedback channels as well as different kinds of feedback. This strategy is implemented in the feedback transformer component depicted in figure 2.

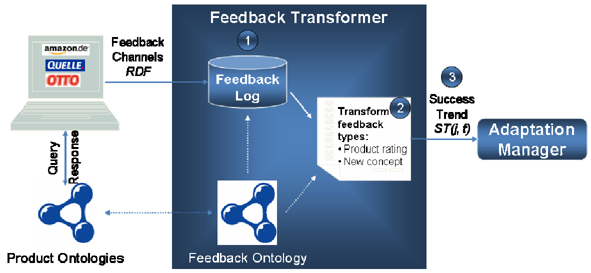

Fig. 2. Conceptual architecture of the feedback transformer component

The strategy comprises the following steps:

1. Gather feedback from the different channels
2. Transform different feedback types
3. Report transformed feedback to the next component

4.1.1 Gather feedback from the different channels. Each feedback channel provides user feedback as RDF triples at separate SPARQL endpoints. The RDF triples are retrieved by the feedback transformer and captured in a semantic feedback log as instances of the feedback ontology (confer next paragraph).

4.1.2 Transform different feedback types. The feedback ontology models the feedback at the product level and additionally contains all product names of the product ontologies. The structure of the feedback ontology enables reasoning about a product and its ratings including the historical development as well as identifying concepts and relations to be newly added to the product ontology. Accordingly, we distinguish between the three feedback types “KPI trend”, “product rating”, and “new concept”. The first two feedback types are converted by either a simple transformation or a feedback evaluation algorithm to values in the range [+1…-1] relating the current transformed feedback to the one in the precedent cycle.

For the feedback type “new concept” the RDF feedback includes the product name and a new potential concept to be eventually added to the product ontology, e.g. information like aspects or relevant features of a product. A new sub-concept for the aspect/ feature is created in the feedback ontology and its count gets related to the count of all concepts in the respective product ontology. When reaching a defined threshold, the new concept is added to the respective product ontology.

The semantic feedback log captures the exact sequence of the reported feedbacks. Each feedback is associated with the respective product (i.e. the RDF feedback contains the corresponding product name) and represented as instances of the sub-concepts of “Feedback”. These instances contain the product name, feedback channel, date and time of the feedback, rating, and the certainty of the rating as well as the number of concepts contained in the product ontology. The log allows the analysis of the feedback development.

4.1.3 Report transformed feedback to the next component. After having transformed the different feedback types, the calculated metrics relating the transformed feedback in the current feedback cycle to the one in the precedent cycle are reported to the next component, i.e. the adaptation manager.

4.2 Ontology evolution strategy

The ontology evolution strategy defines how the product ontologies change. It associates the Success Trend values to evolution actions and ensures a consistent new version of a product ontology. This strategy is implemented in the adaptation manager component depicted in figure 3.

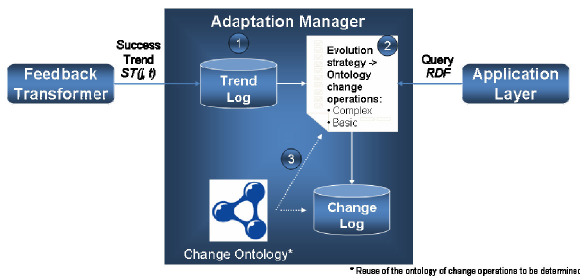

Fig. 3. Conceptual architecture of the adaptation manager component

The strategy comprises the following steps:
1. Gather feedback trends
2. Associate ontology changes with evolution strategies
3. Ensure a consistent ontology evolution

4.2.1 Gather feedback trends. In each feedback cycle the transformed feedback gets reported to the adaptation manager. The feedback is based on the product level. Each reported feedback is captured in a trend log at the product level.

4.2.2 Associate ontology changes with evolution strategies. The central task of the ontology evolution strategy and the adaptation manager is to choose the right evolution, i.e. ontology changes, for the transformed feedback.

Derived from user scenarios, evolution strategies are defined reflecting different behaviours and associating ontology changes, namely:
- Risky Evolution (“always evolve differently”): Regardless of the feedback trend between two consecutive feedback cycles, other complex ontology change operations are executed
- Progressive Evolution (“learn from the past”): Depending on the leap of the trend, same or different complex ontology change operations are executed; in case of a

negative trend, it is optional to either do a different complex ontology change operation or a rollback; additionally, with a threshold indicating the increase of the trend between the current and the precedent cycle the "risk" of the evolution can be adjusted and the strategy tuned towards the Risky Evolution (with a higher threshold)
- Safe Evolution ("only revert negative trends"): In case of a negative trend, a rollback is executed
- Rollback ("undo the ontology changes"): Reverts the ontology changes from the precedent feedback cycle and is based on any reason or decision of the manager; it is executed only once but can be manually chosen multiple times

4.2.3 Ensure a consistent ontology evolution. To execute ontology changes, an ontology evolution algorithm has to be formulated respecting the basic and complex ontology change operations, the ontology consistency model, and the belief revision principle of minimal change [Konstantinidis, G. et al. 2007].

The preferred concept of ontology versioning is change-based versioning (i.e. each state gets its own version number and additionally stores information about the changes made), because it facilitates change detection, integration, conflict management [Mädche, A. et al. 2003], and it allows the interpretation how ontology changes influence the KPI. A change-based versioning can be best realised by tracking the ontology changes in a semantic log [Mädche, A. et al. 2002].

The change ontology models the applicable changes and meta-information and provides the semantics of all possible ontology changes. The root concept is "Change". Its hierarchy consists of the sub-concepts "complex ontology change operation" and "basic ontology change operation". Appropriate relations like "previousChange" model the history of the ontology changes and construct the sequence of the required changes. The structure of the change ontology enables reasoning about changes including their historical development.

The semantic change log captures the exact sequence of the ontology changes executed. Each change is represented as instances of the sub-concepts of "Change". The log allows the analysis of the change development including realising a rollback.

5. Evaluation

According to the research methodology by Hevner and March for "methods", the following evaluation methods and criteria are proved and tested:
- Evaluation method: "Case studies"
- Evaluation criteria: "Implementability"

The required research artifacts for "methods" are listed in chapter 3.1 and are elaborated along the structure of this report and the later dissertation. These particulars on the solution enable a feasible evaluation procedure.

The case study/ use case is a "real-world" conversational content-based e-commerce recommender system, the domain modelled is the product category "digital cameras" (and if necessary further ones), and two feedback channels – from the Web application and from Linked Open Data – are utilised. As the recommender is already used in live e-commerce applications, the evaluation of the system adaptations is a real-world scenario.

Implementability is proven by developing an adaptive ontology giving better recommendations to the user of the e-commerce recommender.

The success of an e-commerce recommender system is usually defined by the achieved conversion rate (i.e. customers-to-recommender users ratio) or click-out rate (i.e. clicks-to-recommendations ratio).

The evaluation scenario is to test and evaluate the impact of the ontology evolution by utilising the formulated evolution strategies, i.e. Risky Evolution, Progressive Evolution, and Safe Evolution. The impact of the ontology evolution will be analysed and evaluated with regard to the key performance indicators at the application level after each to be defined number of accomplished recommendation processes. According to the respective results and reported feedback, the ontology evolves, and eventually adapted recommendations are presented to the user. The feedback circle of the automated system concludes with re-evaluating the key performance indicators after having again reached the defined number of recommendation processes. This evaluation procedure will be executed for all three evolution strategies and evaluated analogously.

At this stage of the research, the adaptation layer has not been developed. Hence, the evaluation cannot start yet.

An interesting result of the evaluation scenario, though, would be that one of the three evolution strategies leads to a higher increase of the key performance indicators.

6. Future work

The following issues remain to be approached in the context of the thesis:
- Investigate related work in the area of machine learning/ artificial intelligence
- Define interfaces for RDF feedbacks (ratings, proposals for new concepts)
- Formalise definitions, algorithms, and axioms
- Define complex and basic ontology change operations needed
- Evaluate the reuse of the ontology of change operations
- Elaborate evolution strategies
- Select the rule language and define rules for the ontology evolution
- Specify the evaluation method, scenario, and key performance indicators

7. References

[Aktas, M. S. et al. 2004] *A Web based conversational case-based recommender system for ontology aided metadata discovery*, Proceedings of the Fifth IEEE/ ACM International Workshop on Grid Computing (GRID ‘04), pp. 69-75.

(Supervisor: Univ.-Prof. Dr. Dieter Fensel)

[Blanco, Y. et al. 2005] *AVATAR: An approach based on semantic reasoning to recommend personalized TV programs*, Special interest tracks and posters of the 14th International Conference on World Wide Web, pp. 1078-1079.

[Broy, M. et al. 2009] *Formalizing the notion of adaptive system behavior*, Proceedings of the 2009 ACM Symposium on Applied Computing (SAC '09), pp. 1029-1033.

[Drachsler, H. et al. 2009] *Effects of the ISIS recommender system for navigation support in self-organised Learning Networks*, Journal of Educational Technology and Society, Volume 12, Number 3, pp. 122-135.

[Haase, P. et al. 2005] *A framework for handling inconsistency in changing ontologies*, Proceedings of the 2005 International Semantic Web Conference (ISWC05), pp. 353-367.

[Kadlec, T. and Jelínek, I. 2007] *Ontology-based approach to adaptation*, Proceedings of the 2007 International Conference on Computer Systems and Technologies (CompSysTech '07), Volume 285, pp. VI.23-1-VI.23-4.

[Klein, M. and Noy N. F. 2003] *A component-based framework for ontology evolution*, Proceedings of the IJCAI-03 Workshop on Ontologies and Distributed Systems.

[Konstantinidis, G. et al. 2007] *Ontology evolution: A framework and its application to RDF*, Proceedings of the Joint ODBIS & SWDB Workshop on Semantic Web, Ontologies, Databases.

[Mädche, A. et al. 2002] *Managing multiple ontologies and ontology evolution in Ontologging*, Proceedings of the IFIP 17th World Computer Congress – TC12 Stream on Intelligent Information Processing, pp. 51-63.

[Mädche, A. et al. 2003] *Managing multiple and distributed Ontologies on the Semantic Web*, The VLDB Journal – The International Journal on Very Large Data Bases, Volume 12, Issue 4, pp. 286-302.

[Maidel, V. et al. 2008] *Evaluation of an ontology-content based filtering method for a personalized newspaper*, Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08), pp. 91-98.

[Middleton, S. E. et al. 2001] *Capturing knowledge of user preferences: Ontologies in recommender systems*, Proceedings of the First International Conference on Knowledge Capture (K-CAP '01), pp. 100-107.

[Middleton, S. E. et al. 2003] *Capturing interest through inference and visualization: Ontological user profiling in recommender systems*, Proceedings of the Second International Conference on Knowledge Capture (K-CAP '03), pp. 62-69.

[Noy, N. F. et al. 2006] *A framework for ontology evolution in collaborative environments*, Proceedings of the 2005 International Semantic Web Conference (ISWC05), pp. 544-558.

[Peyton, L. 2003] *Measuring and managing the effectiveness of personalization*, Proceedings of the Fifth International Conference on Electronic Commerce (ICEC '03), pp. 220-224.

[Stojanovic, L. et al. 2002] *User-driven ontology evolution management*, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW '02), pp. 285-300.

[Stojanovic, N. et al. 2003] *The OntoManager – a system for the usage-based ontology management*, LNCS 2888, pp. 858-875.

[Tran, T. et al. 2006] *Rules for an ontology-based approach to adaptation*, Proceedings of the First International Workshop on Semantic Media Adaptation and Personalization (SMAP '06), pp. 49-54.

[Zhou, Y. et al. 2007] *Applying ontology in architecture-based self-management applications*, Proceedings of the 2007 ACM Symposium on Applied Computing (SAC '07), pp. 97-103.