

ArchReco: Software Architecture Design Tool Enhanced by Context Aware Recommendations for Design Patterns

George A. Sielis
University of Cyprus
Department of Computer Science,
P.O.Box 20537, 1678,
Nicosia, Cyprus
sielis@cs.ucy.ac.cy

Aimilia Tzanavari
University of Nicosia,
Department of Multimedia and Design,
P.O.Box 24005, 1700,
Nicosia, Cyprus
tzanavari.a@unic.ac.cy

George A. Papadopoulos
University of Cyprus
Department of Computer Science,
P.O.Box 20537, 1678,
Nicosia, Cyprus
george@cs.ucy.ac.cy

ABSTRACT

This work presents ArchReco, a tool that supports Context Aware Recommendations for Design Patterns while a designer performs a Software Architecture Design. The prototype tool presented in this work takes into consideration the contextual elements that characterize a Software Architecture model and use them for the computation of the most suitable Design Patterns that can be applied during the process of Software Architecture diagram design. The tool currently supports three Context Aware Recommendation methods for the computation of recommendations for the most suitable Design Patterns.

Categories and Subject Descriptors

D.2.11 [Software Engineering]: Patterns

General Terms

Design

Keywords

Design Patterns, Context Aware Recommendations, Software Architecture Design

1. INTRODUCTION

Design Patterns, as defined in [1], *is a nowadays a well-known and used software engineering problem-solving discipline that emerged from the object-oriented community*. Design patterns are designed as “templates” that intend to solve particular problems in a specific context. The solution of such problems can adopt the solutions provided by more than one individual design pattern or their combination, depending on the kind or type of the module/component designed. Therefore, by having well-defined requirements that satisfy the required specifications of new designed software, reusing existing Design Patterns can facilitate its overall Architectural Design.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s). *ECSAW '15*, September 07-11, 2015, Dubrovnik/Cavtat, Croatia ACM 978-1-4503-3393-1/15/09. <http://dx.doi.org/10.1145/2797433.2797502>

The current work refers to a Software Architecture Design prototype tool, the ArchReco, which supports Context Aware Recommendations for Design Patterns. ArchReco is a Software Architecture Design tool enhanced by three Context Aware Recommendation Algorithms for the suggestion of the most proper Design Pattern or combination of Patterns that can be applied in a Software Architecture diagram design. The ArchReco development emerged from the examination of some of the most known Software Architecture Design tools where the lack of recommendations and particularly the recommendations for the Design Patterns was obvious.

The rest of this document is structured as follows: The prototype tool is part of a research work related to Context Aware Recommendation Algorithms and their application in the case of the Software Architecture Design process. Part of this work was the examination of Software Architecture Design tools, in order to track how Context Aware Recommendations can influence the design process. The examination of some of the most known tools and a reference to the survey that motivated the current work are presented in section 2. In section 3 the ArchReco prototype is introduced. Section 3 describes the technologies that were used as well as the Context Aware Recommendation algorithms for Design Patterns. Section 4 refers to the related work and finally section 5 summarizes the current status of the work that has been done and refers to the future work.

2. MOTIVATION

The motivation for the development of a tool enhanced by Context Aware Recommendations for Design Patterns is based on the results of a survey [12] that was performed aiming to identify the needs of the Software Architecture Design professionals in terms of aiding mechanisms that they would like to have in Software Architecture Tools that they use. In the survey we collected opinions from 28 participants, from which 14 were coming from the private sector (11 International Private Corporate, 1 Small Medium Enterprise and 2 Self-employed), 1 from the public Sector and 13 from the Research/Academic sector. Their ages were between 26 years old and 55 years old. In particular, the survey was answered by 12 Software Developers, 1 Software Architect, 3 Software Development Team Leaders, 2

Project Managers, 1 Manager of Software Development Department, 5 Research Associates, 3 Academic Professors and 1 System Administrator. The main outcome of this survey was that the experts in the field usually need additional resources (human or knowledge) during the design process but they usually find those resources from the web or by asking other people. Therefore the need for recommendations and in particular Context Aware recommendations in such tools became an interesting and challenging topic to investigate.

Studying the field of Software Architecture Design and in particular the benefits that Design Patterns can offer, led to the decision to develop a software design tool enhanced by two attributes: Context Aware Recommendations for Design Patterns and Context Aware Recommendations for social activities, such as group composition, related projects, professional network, etc. This decision was taken after the examination of some of the most known Software Architecture Design Tools that the survey professionals mentioned as the most common tools that they use. The list of the examined tools and their characteristics is depicted in Table1.

Table 1. Examined SAD Tools and their Attributes

Tool name	¹ Licensing	User-Support Mechanisms	Social attributes	² Type
Eclipse	OS	Code Generation for Java, C++ and C	Code Sharing between a team	D
Microsoft Visio	C	None	None	D
Enterprise Architect	C	None	Team based repositories and version control tools	D
Magic-Draw	C	Data Definition Language (DDL) generation	Server for real-time modeling by team	D
StarUML	OS	None	None	D
Argo-UML	OS	Design Critics	None	D,W
IBM rational	C	None	IBM Cloud	D

ArchReco is a prototype tool that attempts to cover the two deficiencies that were tracked in the existing tools. Therefore the objectives that ArchReco has to fulfill is firstly to provide mechanisms that support users in using Design Patterns and second, make the user aware of what other professionals in the field are doing in order to collaborate with them. In this work we

¹ Commercial (C), Open Source (OS)

² Desktop Application (D) or Web Application (W)

demonstrate ArchReco in its current status, which refers the first objective while the second is still under development.

In the existing literature several attempts have been made to provide recommendations on the use of Design Patterns [1, 3, 4, 5]. Taking into account existing methodologies, this work suggests three methodologies from which one is a variation of an existing methodology [6] and the other two are new approaches.

The common attribute of the suggested methodologies is the use of context-awareness as the decision-filtering factor having in each case different context elements to take into account. The selection of the context elements varies according to the process phase that the recommendations of patterns may be requested and based on the available information at the time of request.

3. ARCHRECO PROTOTYPE

ArchReco is built on top of the jGraph framework [2] by adding additional functionality to support the recommendations for Design patterns in real time, during the design of Software Architecture Diagrams. The recommendation algorithms and the methods of presenting the recommendation results were built as individual components and integrated with the basic graph editor tool, which exists in the jGraph package.

3.1 ArchReco Features

The Software Architecture prototype is a complete graph editor consisted of the following modules/components:

- *Graph Editor Canvas* where designer is able to design Architecture Diagrams (Support of styling and editing the diagrams).
- *Palette of diagram components* where all the diagram components exist and the designer is able to drag to and drop the components in the canvas.
- *Architecture Diagram Information Area* where the designer can write the metadata information of the working model such as description or title.
- *Diagram Actions Area module*, which is changing according to the selected recommendation algorithm. In this area the designer is able to see information of the selected nodes (from the designed diagram) or interact with the tool according the recommendation algorithm requirements.
- Recommendation of Design Patterns Area module: In this area the Design Patterns Recommendation results are shown. It is structured in such a way that the designer -by selecting a recommended pattern- can get a full knowledge of its characteristics, its generic UML diagram and is also able to insert the selected pattern in the diagram by pressing a button.

3.2 Technologies

For the development of the components that developed and integrated with the prototype software a combination of methods and technologies were used. For modeling the data, 2 ontologies were designed and used in combination with database engine, sparql ontology query language as well as dynamic transformation of data into ontology based description language,

whenever this was necessary. The use of semantic web technologies, especially for modeling the data, was done for two reasons: Firstly, for reusability reasons, since the conceptual model that designed can be reused for the development of similar applications. Secondly, semantic web technologies offer the flexibility to retrieve data from multiple sources, handle and structure the way it suits the application and easily query the data to retrieve the necessary information. Like for example the Design Patterns ontology is currently containing the GoF [3] Design patterns, but in case that a new Design Patterns data source has to be added, the new data source can be added in the tool only by small modifications (most of the cases configuration changes only). The two ontologies that were designed are 1. *Design Patterns Ontology model* and 2. *Ontology for modeling the context based on the contextual elements we identify for the current domain of work*, the Software Architecture Design. Figure 2, depicts how the technologies that ArchReco uses, exchange data information between each other. The data are retrieved from the database, (in the cases that the database is used) and it is transformed into ontology-based language. Then the transformed data is parsed by the Context Aware Recommendation Filtering component using sparql query language. After that, depending on the selected Context Aware Recommendation algorithm the filtering component generates the list of recommendations, which is presented on the ArchReco tool. The interaction between user and ArchReco consists of a set of actions, which will be used in the future for generating personalized recommendations, and thus those actions are sent to the Database. When the tool is used offline, the sql queries for the collection of data into the database, are stored into local files in order to be synchronized when the tool goes online.

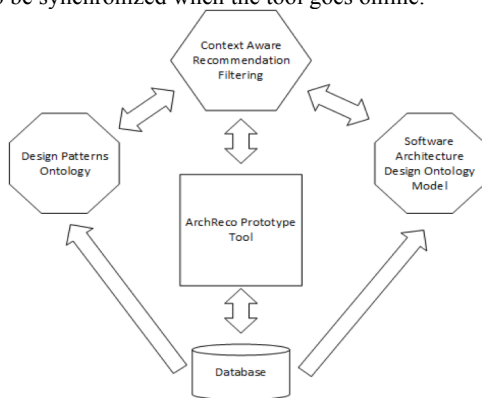


Figure 2. High Level Representation of the Technologies used

3.3 Context Aware Recommendation Algorithms

ArchReco currently supports three Context Aware Recommendation algorithms for recommending Design Patterns: 1. *Text Based filtering*, 2. *Rule Based filtering* and 3. *Utility-Based filtering*. Text Based Algorithm uses the description of the diagram elements, given by the designer, and with the use of TF-IDF filtering generates the list of Recommended Design Patterns that exist in a complete set of Design Patterns modeled in Design Patterns ontology. Rule based Filtering generates

Recommendations for Design Patterns by analyzing the connections between the graph elements that a diagram contains and produces the list of recommended Design Patterns based on rules that are extracted from the related literature. Utility Based algorithm uses the input description of all graph elements and also the metadata information of a diagram (Title, Description) and generates the list of recommendations based on a Utility Function. The Utility Function is using weights of importance for each context element that they are used for the computation of the utility factor of each Design Pattern based on the user preferences. The designer is able to define the weights for each particular context element through the interactive component of the prototype tool, when the utility-based algorithm is selected.

3.3.1 Text Based Algorithm

The contextual elements for the patterns recommendation filtering are defined with the description that the designer gives to the individual nodes of the diagram. The text description of each individual graph element is mandatory for the production of recommended Design Patterns, which will be related to the particular node. The recommended Design Patterns list is generated when the designer describes in free text the role of the design element in the overall design select the element and then request recommendation of Design Pattern that is related to the particular element description.

3.3.2 Rule Based Algorithm

From the existing Design Patterns literature [3, 4, 5] it is possible to define the context of each pattern based on the type of the design elements (e.g. class, constructor, package, interface etc.) and the relation between them through their connection edges. The definition of each pattern is based on rules that must be satisfied in order to make the decision that a specific pattern is suitable enough to be applied for the solution of a problem.

3.3.3 Utility Based Algorithm

Similar to the other two methods, the Utility-based Recommendation method uses text filtering or diagrammatic rules for the computation of the recommendations. The difference is that this method is more dynamic and complex -in relation to the other two- since the user has the control of which context elements will be used in the filtering process and with what importance level. This is done by providing the designer with a form on which all context elements are listed and the designer has the option to select whether an element should be included in the recommendation filtering or not. For each context element the designer can also define the weight of importance for each factor. Those values are passed into a utility function, which produces the utility factor for each pattern. The results are ranked based on the utility factor ($U = \text{Sum}(W_i * C_i)$ where W_i is the weigh value and C_i is the contextual factor comparison result) that each pattern receives.

4. RELATED WORK

In the latest years, Design Patterns for Software Architecture design are increasing in number with new patterns appearing to cover wither general functionalities or more dedicated domains (e.g., mobile application design, or user interface design). Some

previous works have focused on providing recommendations on the appropriate usage of Design Patterns. Gueheneuc et al [7] proposed a methodology of recommending design patterns through the textual analysis of each pattern into the most important words and computing the similarity distance between those words and the words of the query given by the user. Gomes et al. [8] proposed a Case Based Reasoning (CBR) Recommendation system for the recommendation of Design Patterns based on previous experiences using a Design Patterns Knowledge Base and related taxonomies. A similar system developed by [9] that recommends patterns using the Implicit Culture Framework (ICF). The recommendations are produced based on the users' previous actions, based on conventional Information Retrieval and CBR methods. Palma et al. in [10] propose a DPR framework, which recommends patterns based on predefined questions that the designers have to answer, and based on the given answer the framework has a weighting mechanism for the selection of the appropriate pattern. The initial identification of patterns that can be used through the DPR framework [10] are selected through LUCENE indexing and TF-IDF filtering of the query given and the intent description of each pattern.

The current work differs from the above approaches in the following aspects: 1. *It supports recommendation methods used for recommending Design Patterns*, and 2. *the prototype tool applies the proposed methods and offers to the users the support in finding and using Design Patterns*. In particular, the two of the methods used in this work differ from the existing methods found in the literature, since in this work designed by taking into account the most important context elements that are related to the Design Patterns. These methods are the rule based and the utility based recommendations for Design Patterns. The Text-based method was already used in similar research works [7, 8, 9] with the difference that in these works for the text comparisons only specific Design Patterns attributes were used e.g. Implementation or Intent. In this work the text comparison is using all Patterns attributes to find commodities with the text defined by a user. In addition to that, Rule-based recommendations for Design Patterns is based on rules that were extracted from the description of each pattern's context, taken from literature sources, such as [11, 3] and formalized according to the needs of this work. Finally, the use of Utility-based recommendations is a well-known method in Recommendation Systems research area but not commonly used for the current topic and context, the recommendation for Design Patterns.

5. CURRENT STATUS & FUTURE WORK

ArchReco is a Research prototype tool that was developed as part of a research work related to the topics of Context Aware Recommendations in Creativity Support Tools and in particular the case of Software Architecture Design as Creative process. At the time being, the development of Context Aware Recommendations for Design Patterns is completed and a first round of evaluation regarding the usefulness, the usability and user experiences was contacted. The tool is a semantic web

desktop application and can be found and downloaded at <http://www.cs.ucy.ac.cy/~sielis/ArchReco.zip>.

As future work the task is to complete the second objective, which is the development of the social network functionality, and integrate this with the current prototype tool. This will add new features to the tool such as Context Aware Recommendations of professionals to monitor and guide an Architecture Design process and Context Aware Recommendations for knowledge resources like documents or urls to be linked with a diagram design project.

6. REFERENCES

- [1] Alexander, Christopher, Sara Ishikawa, Murray Silverstein. A Pattern Language: Towns, Buildings, Construction. New York: The Oxford University Press, 1977.
- [2] Mutzel, P., Jünger, M., Leipert, S., JGraph - A Java Based System for Drawing Graphs and Running Graph Algorithms, Graph Drawing, Lecture Notes in Computer Science, doi: 10.1007/3-540-45848-4_45, isbn: 978-3-540-43309-5, pp. 459-460, 2002
- [3] Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995
- [4] Stal, M, Sommerlad, P., Rohnert, H., Meunier, R., Buschmann, F., Pattern-Oriented Software Architecture, Volume 1, A System of Patterns, John Wiley & Son, ISBN: 9780471958697, 1996
- [5] Braude, E., J., Software Design: From Programming to Architecture, ISBN: 978-0-471-20459-6, 2006
- [6] Gueheneuc, Y.,G., Rabih.,M., A Simple Recommender System for Design Patterns, Proceedings of the 1st EuroPLoP Focus Group on Pattern Repositories (2007)
- [7] Gueheneuc, Y.,G., Rabih.,M., A Simple Recommender System for Design Patterns, Proceedings of the 1st EuroPLoP Focus Group on Pattern Repositories (2007)
- [8] Gomes, P., Pereira, F. C., Paiva, P., Seco, N. P., Carreiro, Ferreira,J.,L., Bento, C., 2002. Using CBR for Automation of Software Design Patterns. In Proceedings of the 6th European Conference on Advances in Case-Based Reasoning, pp:534-548.
- [9] Weiss, M. Birukou, A. 2007. Building a pattern repository: Benefitting from the open, lightweight, and participative nature of wikis. In Workshop on Wikis for Software Engineering at ACM WikiSym, 2007 International Symposium on Wikis (WikiSym), October 21-23.
- [10] Palma, F. Frazin, H., Gueheneuc, Y., Moha, N., "Recommendation System for Design Patterns in Software Development: An DPR Overview", in Proceeding of 2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE), pp. 1-5, 2012.
- [11] Braude, E., J., Software Design: From Programming to Architecture, ISBN: 978-0-471-20459-6, 2006
- [12] Sielis, A. G., Tzanavari, A., Papadopoulos A.,G., Investigating the State-of-the-art in Software Architecture Design Tools and Their Support in Personalized Recommendations: A Survey in Software Architecture Process, Tools and Practices, UCY-CS-TR-15-1, 2015

APPENDIX

A1. Tool Maturity

ArchReco is a research prototype tool is a Desktop Application supported by Semantic Web Technologies. It is a cross platform graph editor, which is delivered as a jar file. The tool at its current status is a complete architecture design tool, which supports the three types of Context Aware Recommendation algorithms for the provision of recommendations for Design Patterns during the Architecture diagram design. The tool at its current phase does not need Internet connection and can be used offline. The ArchReco software can be downloaded from the following url: <http://www.cs.ucy.ac.cy/~sielis/ArchReco.zip> .

A tool demo video can be found here:

<http://youtu.be/hMM0UXAKHic>

A2. Presentation Setup

The presentation will be conducted with the use of computer and projector. There will be a presentation in powerpoint introducing the tool and then a demonstration of the tool will be executed. The tool prerequisites the installation of JVM and jdk 7 or 8.

A3. Tool Category

ArchReco is a Research tool.