

Resolving Requirement Ambiguities Using ‘Self-Organized Reuse’ Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim

Assi. Lec. Zeyad Farooq Lutfi

Al-Mustansiriya University / College of Science

Abstract

Requirement is the very starting point of software system where clients declare what they expect software system to provide. Crucial challenge facing software developer is to reveal valid specification from the requirement which most frequently holds ambiguity. The ambiguity is due to different understanding and mentality of both the customer and the developer.

This paper presents a frame work to resolve requirement ambiguity automatically by recruiting social interaction revenue to create global self-organized reuse semantic wiki. Multi-Agent system has been designed to provide dynamicity to proposed semantic wiki due to the social, semantic, and negotiable attributes of Multi-Agent platform.

Keywords: Software Agent, Semantic Wiki, Social context, Social Software Engineering, Requirement Ambiguity

1- Introduction

Social software engineering (SSE) is a branch of software engineering that is concerned with the social aspects of software development and the developed software. In SSE focuses are on the socialness of both software engineering and developed software. Two domains are holding main approaches to exploit revenues of SSE [1]:

First: Collaborative software development where collaboration is a central activity in software engineering due to the fact that software products are the revenue of multiple software engineers teamed together to produce software. Software engineering collaboration is model-based, centered on the creation and negotiation of shared meaning within the project artifacts that contains the models that describe the final working system [1]. Global software engineering introduces many forms of

Resolving Requirement Ambiguities Using ‘Self-Organized Reuse’ Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

distance-spatial, temporal, socio-cultural-into existing pathways of collaboration [2, 3].

Second: adaptability and dynamicity of software due to the effect of social contexts at runtime (i.e., shaping software adaptation decisions at runtime). System requirements can change implicitly due to the change in the social context, therefore, self-adapting systems have to be able to identify new contexts [3]. Once the context is identified, utility functions evaluate trade-offs between the properties (goals) aligned with the context; this eventually leads to increase the adaptability factor of the software as a response to the social participation [3, 4].

The participants of the 1st international workshop on social software engineering and applications proposed the following characterization [2, 3, 4]:

- Community-centered: Software is produced and consumed by and/or for a community rather than focusing on individuals
- Collaboration/collectiveness: Exploiting the collaborative and collective capacity of human beings
- Companionship/relationship: Making explicit the various associations among people
- Human/social activities: Software is designed consciously to support human activities and to address social problems
- Social inclusion: Software should enable social inclusion enforcing links and trust in communities

Thus, SSE can be defined as “the application of processes, methods, and tools to enable community-driven creation, management, deployment, and use of software in online environments [1].

2- Semantic Wiki and Knowledge Discovery

Traditionally, a Wiki is a website which allows people from anywhere to add, modify, or delete information; this information are represented by data structures organized and indexed according to articles, and it is up to the reader to reveal further knowledge and information, hence wiki does not provide inner relationships among Wiki contents [5]. By the fact that Wiki imposes a huge challenge when it comes to search Wiki for specific article with certain keywords, the outcome of the search will be, most frequently, irrelevant items. The ambiguity of natural language is what Wiki engine still suffer from, and this demands the human intervention to select pages that fit his/her needs[5].

The reason that Wiki does not be considered yet as a resource for knowledge, despite the fact it holds rich information is knowledge

Resolving Requirement Ambiguities Using ‘Self-Organized Reuse’ Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

representation scheme, where knowledge should be represented in a machine-readable form [5,6]. In this paper we have chosen XML based schemas due to its popularity and its readability by most software products.

Due to the shortage in traditional Wikis semantics is proposed to be embedded into Wiki's pages so that the information possessed in these pages are structured enough for machine accessibility and understanding. Anyway, semantic Wiki is defined as a Wiki site powered by a semantic Wiki engine; where users are able to add semantic markup to Wiki pages, and the added structured information can then be used to pragmatically search and investigate these information. Traditional Wikis do not support *information reuse* due to nature of the representation for this information; where information is only represented in natural language; this way it cannot be automatically reused and human intervention is required all the time [5, 6].

In semantic Wiki resources are annotated using formal descriptions; Wiki resources are the Wiki pages. Semantic annotation is a structural directive that offers additional features over the regular Wikis where annotations can be treated as the interface to Wiki page and users can query these annotations directly or create views from such queries. Figure (1) presents semantic Wiki architecture.

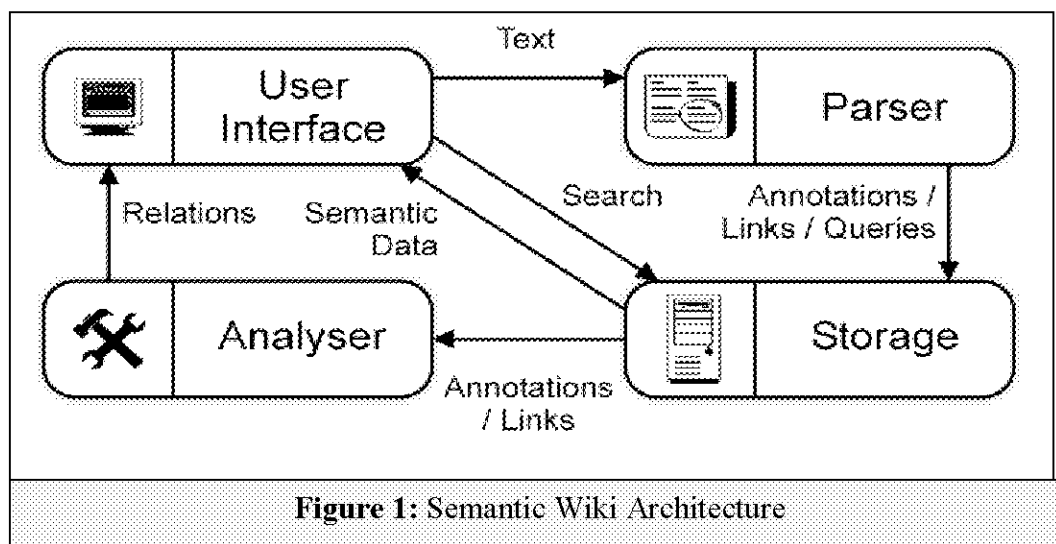


Figure 1: Semantic Wiki Architecture

From figure (1) it is clear that semantic Wiki consists essentially, from user interface, a parser, a data analyzer, and a data store, and those components are interacting each others to annotate certain Wiki page which is an individual node within the semantic network of Wiki.

3- Multi-Agent and Social Network Systems

Intelligent agent is a software module that is capable of perceiving events occurred within the environment through its sensors and acts back using its effectors. Software agent has significant attributes that discriminate it from other software modules such as autonomous, platform independent, mobile, social, and negotiable [7, 8]. According to these features multi-agent systems can play crucial role in promoting social software system to the next level of socialness, and this is due the intuitive sociality behavior of multi-agent platform, and this is obviously enforced by their ability to communicate according to semantic way and later on to develop trust relationships among participants in multi-agent platform. Furthermore, agents have the ability to express their communication acts through acknowledged standards for interoperability among diverse systems, and exchange messages directly, in peer-to-peer scheme [8]. This makes multi-agent platform as the highest candidate to serve in social network systems. Multi-agent platform has been deployed most frequently to develop advanced social platforms. In particular, multi-agent systems have been used as: an underlying layer or a middleware for developing social networking platforms, a technology to increase the autonomous and intelligent behavior of existing systems and a tool to develop simulation environments for studying both online and offline human social networks [7].

The most significant feature exploited in this paper is the semantic communication established over multi-agent platform, where semantic is the outcome of deploying domain ontology that represents the specification of conceptualization within the environment. Applying ontology makes it possible to create complex negotiation protocols which simulate human socialization and provides robust environment for developing trust among users (i.e., users are represented by agents).

Finally, multi-agent systems are a powerful tool for simulating the behavior of online social networks, in the same way they have been used for simulating the behavior of persons in real social environments for a long time. In fact, multi-agent systems have proved to be very effective in the simulation of social networks, both during their initial creation and development and during their further operation. They allow to describe the behaviors of individuals, mimicking the actions of human users in similar contexts, and to analyze the associated emerging behavior of the network as a whole. This way, multi-agent systems can provide precious insights for further improvement of existing social platforms [7].

Resolving Requirement Ambiguities Using 'Self-Organized Reuse' Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

4- Problem Statement

Software life cycle starts over by introducing requirements to be processed by other layered stages of software development process. Anyway, introducing requirements is a complex process in term of maximizing the following mapping function:

$$F : R \rightarrow S$$

R: requirement

S: specifications

SRS (Software Requirements Specifications) has been the success parameter of any software project due to the fact that all software development stages within the lifecycle are working collaboratively to fulfill and satisfy the SRS.

The challenge is to ensure requirement validity, consistency, correctness and completeness before starting over diving into the design and the implementation; this challenge is rapidly increased as the requesting software getting more sophisticated. Here in this paper we are focusing on the ambiguity of requirements, no matter the type of requirement (i.e., functional, performance, interface, design, and development standards); this ambiguity is to be detected, remedy or removed (i.e., maximizing above mapping function). Ambiguity in requirements can turn development process to a real disaster in term of cost and time.

5- Paper contribution

In this paper multi-agent system is introduced and deployed to autonomously update SRS semantic Wiki, where individual agent is installed on developer site to deliver his/her experience and knowledge to the semantic Wiki. Developer experience in revealing specification from requirement is abstracted as XML-based profile sent over the internet to semantic Wiki site, and received profile is inserted into Wiki structure based on the interpretation of its content. Semantic Wiki introduced by this paper has the following significant properties:

- 1- Semantic Wiki = $\sum_{i=1}^N \text{XMLnode}$
- 2- XMLnodes are structured in a hierarchal tree and allocated based on the requirement categorization.
- 3- XMLnodes are removed, re-allocated, updated due to the socialization context produced by Multi-agent interaction.
- 4- XMLnodes hold knowledge of developers to reveal specification from requirement.

Resolving Requirement Ambiguities Using ‘Self-Organized Reuse’ Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim , Assi. Lec. Zeyad Farooq Lutfi

- 5- Contacting XMLnodes helps remedy the problem of requirement ambiguity for both the developer and the customer.
- 6- XMLnodes are annotated due to social context resultant from Multi-Agent interaction session.

6- The proposal

The proposal of this paper is after creating XML-semantic Wiki to hold SRS nodes in wide spectrum of applications. The requested semantic Wiki should provide all functionalities provides by traditional Wikis such as update, edit, delete and add SRS nodes.

The proposal of this paper can be abstracted by

$$\forall(A : A \in \text{MAP}) \forall(R : R \in \text{SWiki}) \exists \text{KnowGain}(A, R) \text{ IFF } \text{Communication}(A, \text{SWiki})$$

Where

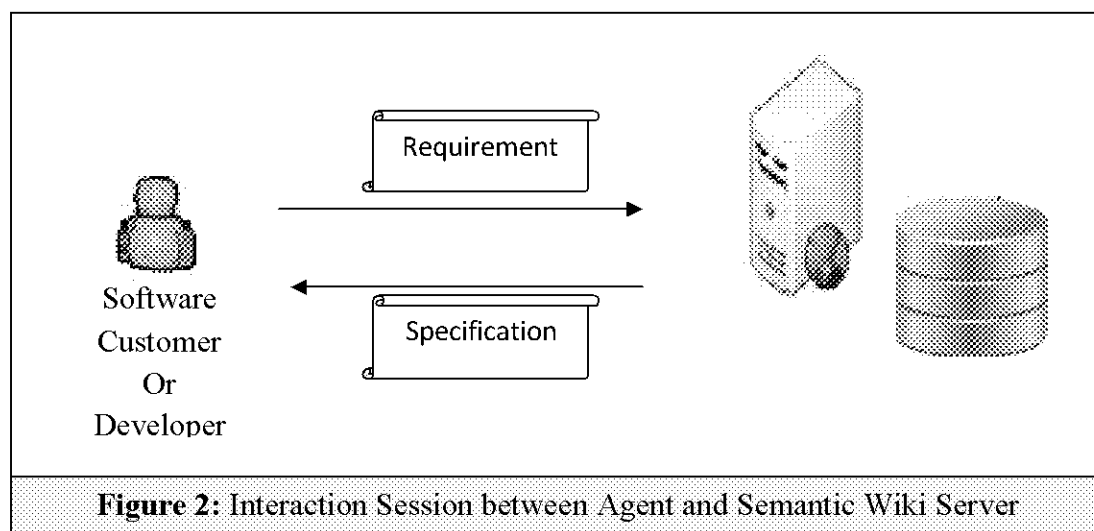
A: Agent

MAP: Multi-Agent Platform

R: Requirement

SWiki: Semantic Wiki

Figure (2) illustrates interaction session between individual Agent and Semantic Wiki server



When software customer is interacting semantic wiki he/she can check for the ambiguity for the requirement by surfing the wiki for the specification already embedded within the wiki by software developers. Developers, in other hands, make use of other developers' efforts in working on the same requirement.

Requirement is embedded in the semantic Wiki using XML-template; this makes it easier to be interpreted by third software parties and to build a semantic network of these templates. Figure (3) presents

Resolving Requirement Ambiguities Using ‘Self-Organized Reuse’ Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

wiki's node constructed from XML-based templates sent by software developers.

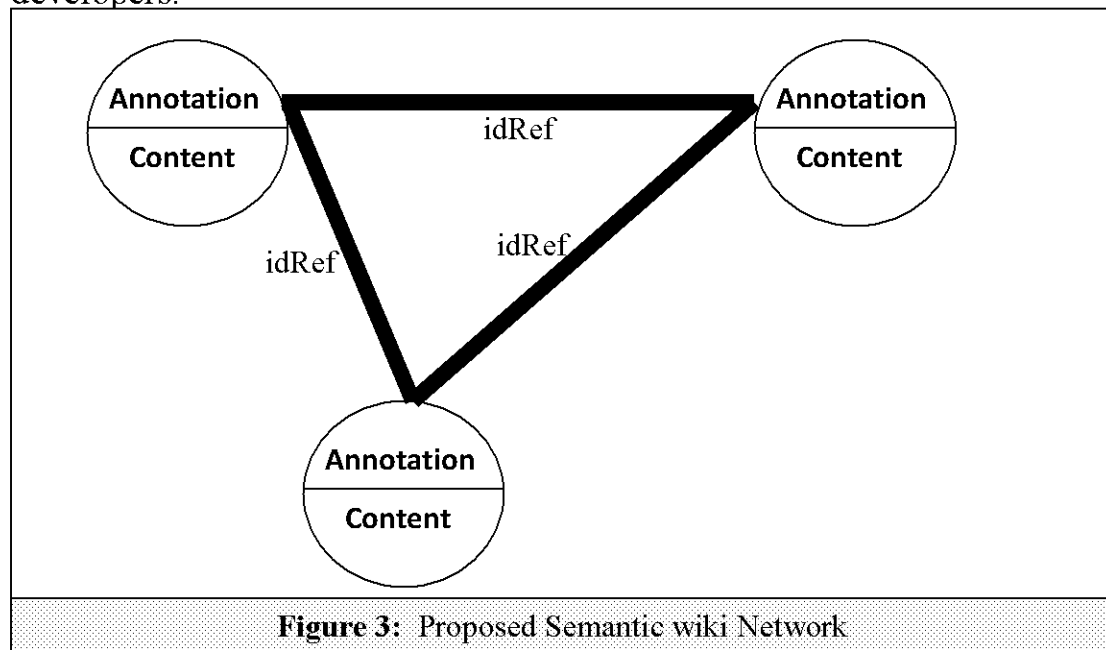


Figure 3: Proposed Semantic wiki Network

As it is obvious from figure (3) each node within the semantic wiki is composed of two parts; the first one is the annotation where all social interaction is about. Annotation is an XML-based text describing the content part; the description includes but not limited to author profile, requirement declaration, and reference pointer to other annotations, as it is presented in figure (4).

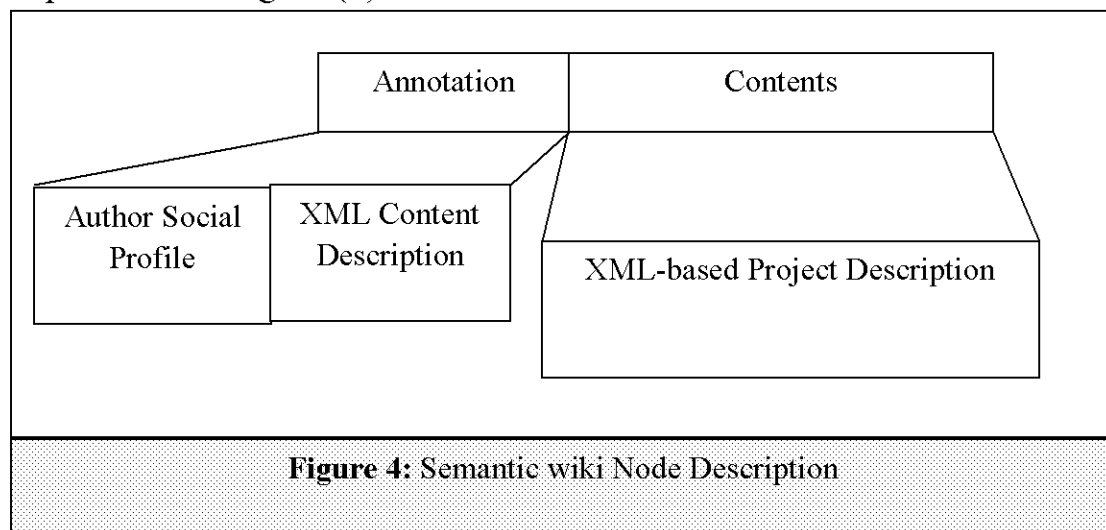


Figure 4: Semantic wiki Node Description

Resolving Requirement Ambiguities Using 'Self-Organized Reuse' Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

```
<Requirement uid="****" >
<Id>
<Identifier id="****" idContextRef="****" idRoleRef="****" />
</Id>
<ApplicationDomain uid="****" >
<Definition uidRef="****">
</ApplicationDomain>
<LifeCycleStage uid="****" >
<Definition uidRef="****" >
</LifeCycleStage>
<Task>***</Task>
<Functional>***</Functional>
<Requires>***</Requires>
```

Figure 5: XML-Template for Semantic Wiki's Node Content Description

Figure (7) illustrates the system scheme introduced by this paper.

```
<Author>
<Name> ***</Name>
<Title> ***</Title>
<Organization> ***</Organization>
<OriginatingSystem>***</OriginatingSystem>
</Author>
```

Figure 6: Author Social Profile Template

Resolving Requirement Ambiguities Using 'Self-Organized Reuse' Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

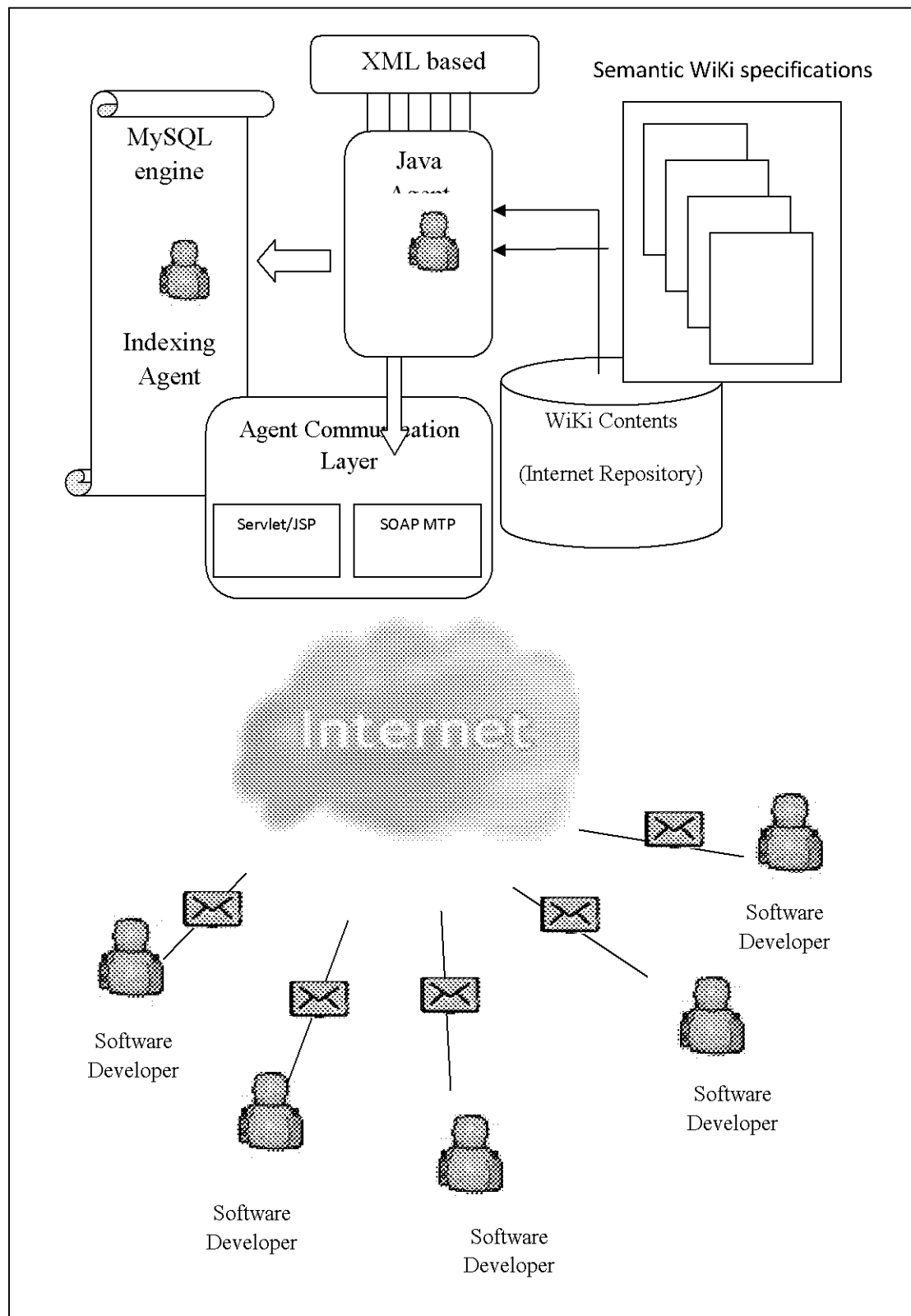


Figure 7: System Scheme Proposed by This Paper

Resolving Requirement Ambiguities Using 'Self-Organized Reuse' Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

In figure (7) SOAP MTP refers to simple object access protocol / Message Transport Protocol which is used by agent to be able to talk to web based components like web services, where these web services are frequently used by software developers toolkits. Semantic Wiki depicted in figure (7) holds ontological represented records which is constructed using XML tags. Agents are able to perceive in that special repository (Semantic Wiki) due to its ability to conceptualize its records. The most crucial part of the proposed system is getting intelligent agent involved in parsing traffic forwarded to web server hosting semantic wiki; after all semantic wiki is a web application. Figure (8) presents the mechanism used by this paper to intercept http traffic.

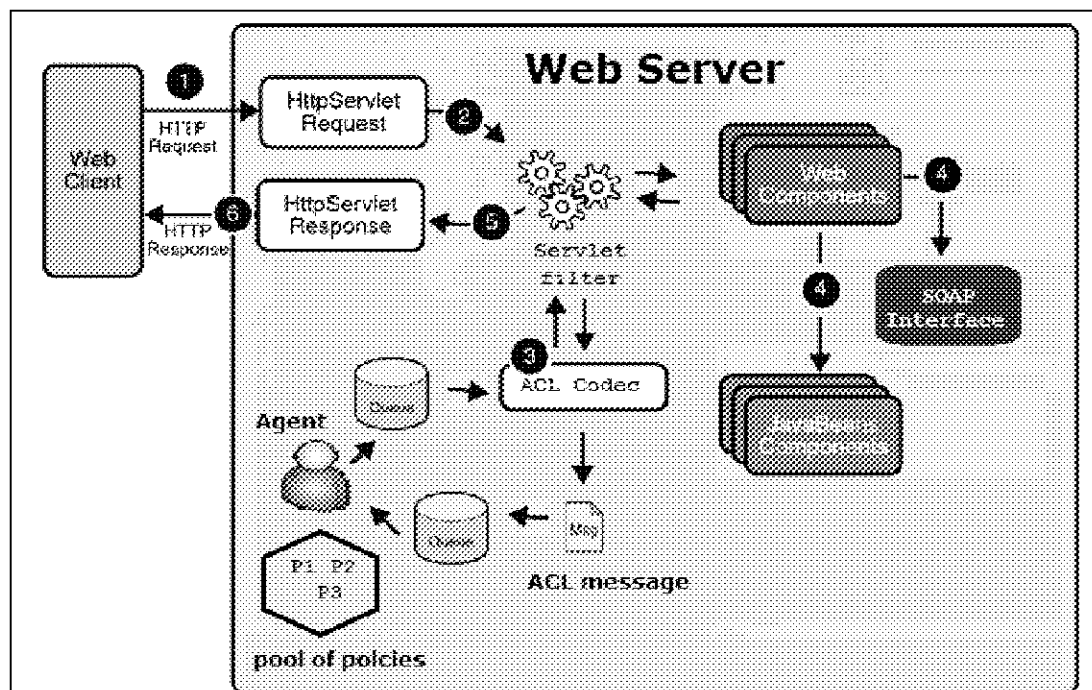


Figure 8: Forwarding Semantic Wiki's Traffic to Server Agent

As figure (8) depicts, by sequence numbers, the traffic is steered out to agent by installing servlet filter which is called first before any web component, and this servlet filter is responsible on forwarding http traffic to agent behaviors. Http packet holds agent XML message as stream of bytes, thus it is up the wiki's agent to interpret it according to predefined ontology which is represented in figure (8) as pool of policies.

Figure (9) presents the sequence diagram of the interaction session between software developer's agent from a side and the interaction session of customer's agent from other side.

Resolving Requirement Ambiguities Using 'Self-Organized Reuse' Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

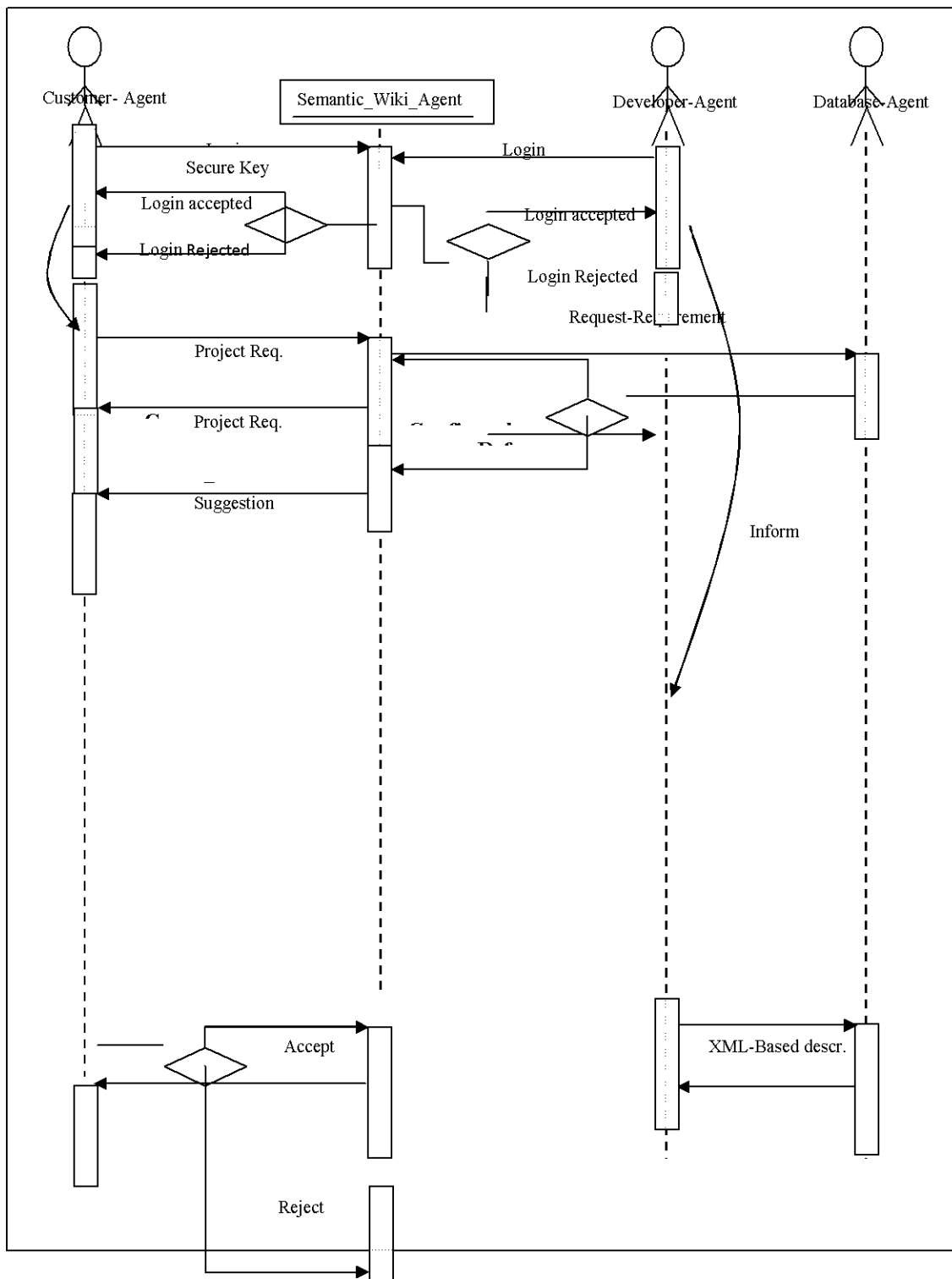


Figure 9: Sequence Diagram of Multi-Agent Based Semantic Wiki Interaction

Resolving Requirement Ambiguities Using 'Self-Organized Reuse' Semantic Wiki and Multi Agent Technology

Assi. Lec. Ethar Abdul Wahhab Hachim, Assi. Lec. Zeyad Farooq Lutfi

7- Conclusions

- 1- Business intelligent is highly affected by social software engineering aspects due to the new parameters added by including social context along the design.
- 2- Interoperability among distributed components can be established throughout using global semantic Wiki, where information about certain data types and formats can be structured as semantic Wikis and provided automatically to requested software modules.
- 3- Security factor still represent a crucial challenge in building trust among distributed agents and the semantic Wikis.
- 4- Fuzziness could be produced in autonomous systems in their attempts to develop knowledge and thus a countermeasure for ontology verification and validation in problem domain should be considered.

8- References

- 1- Ivan Mistrik, John Grundy, Andre van der Hoek and Jim Whitehead, "Collaborative Software Engineering", Springer-Verlag, Berlin, ISBN: 978-3-642-10293-6, 2010.
- 2- Steffen Lohmann and Thomas Riechert, "Adding Semantics to Social Software Engineering: (Re-) Using Ontologies in A Community-oriented Requirements Engineering Environment", Software Engineering Workshops, volume 160 of LNI, page 485-494.GI, 2010.
- 3- Betty H.C. Cheng, Rogerio de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee, "Software Engineering for Self-Adaptive Systems: A Research Roadmap", Springer-Verlag, Berlin, Printed in Germany, ISBN-13: 978-3-642-02160-2, 2009.
- 4- Imed Hammouda, Jan Bosch, Mehdi Jazayeri, Tommi Mikkonen: First International Workshop on Social Software Engineering and Applications (SoSEA 2008). In: Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2008). IEEE, 2008, pp. 531-532.
- 5- Liyang Yu, "A developer's Guide to The Semantic Web", Springer Heidelberg, USA, ISBN: 978-3-642-15969-5, 2011.
- 6- Michael C. Daconts, Leo J. Obrst, and Kevin T.Smit, "The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management", Wiley Publishing, Inc, Indiana, USA, ISBN: 0-471-43257-1, 2003
- 7- Fabio Maccheroni, Massimo Marinacci and Aldo Rustichini, "Social Decision Theory: Choosing within a between Groups", Oxford University Press, vol.79 (4), pages 1591-1636, 2008.
- 8- Fabio Bellifemine, Giovanni Caire, and Domini Greenwood, "Developing Multi-Agent Systems with JADE", John Wiley & Sons Ltd, England, ISBN: 978-0-470-05747-6, 2007.

حل غموض المتطلبات باستخدام ويكي معتمد على المعنى وذاتي التنظيم

وتقنية متعدد الوكلاء

الخلاصة:

تعتبر المتطلبات هي نقطة الشروع الاولى لبناء نظام برمجي حيث ان الزبائن يقوموا باعلان ما يتوقعون من النظام ان يؤديه. تحدي كبير يواجه مطور الانظمة لاستنباط المحددات الصحيحة من المتطلبات التي يوفرها الزبون حيث ان هذه المتطلبات تحوي على الغالب غموض والذي هو ناتج عن التباين في فهم واستيعاب المتطلبات من ناحية التمثيل والتفسير.

هذا البحث يقدم اطار عمل لحل مشكلة غموض المتطلبات بطريقة الية من خلال تجديد التفاعل الاجتماعي والنتائج المستحصلة عنه لبناء ويكي عام يعتمد على المعنى وله امكانية التنظيم الذاتي، وتم هنا استخدام نظام متعدد الوكلاء لتقديم الحركية المطلوبة من هذا الويكي المقدم هنا في هذا البحث وذلك بسبب الامكانيات التي يتصف بها نظام الوكلاء من حيث الاجتماعية، التفاعل على اساس المعنى و امكانية التفاوض.