Collaborative Features in Content Sharing Web 2.0 Social Networks: A Domain Engineering Based on the 3C Collaboration Model

Lucas Santos de Oliveira and Marco Aurélio Gerosa

Computer Science Department, Institute of Mathematics and Statistics,
University of São Paulo (USP), Zip 05508-090
{lucasso,gerosa}@ime.usp.br

Abstract. Researchers and developers still replicate ideas with low reuse when developing Web 2.0 applications. A domain engineering identify and document communalities and variabilities of an application family fostering reuse. In this work, we used a domain engineering approach for content sharing features of social networks. We used as a method the FODA (Feature Oriented Domain Analysis) with patterns for computer-mediated interaction to describe the collaborative features and the 3C collaboration model to classify them. To implement the commonalities, we defined and developed a component kit based on an infrastructure named Groupware Workbench. We conducted an experiment and a case study to evaluate the artifacts generated by the domain engineering.

Keywords: Collaborative Systems, Domain Engineering, 3C Collaboration Model, Interaction Patterns, Social Networks, Web 2.0, Groupware.

1 Introduction

Web companies that survived the dotcom crisis were using the internet as a platform, offering collaborative sites based on communities [1]. This concept was called Web 2.0. Differently from the previous model, in which companies produce content for users' access, the Web 2.0 involves the user in this process. Users produce, organize, share, mix, criticize, and update the content. According to Prescott [2], the increasingly amount of web content is a result of the penetration of broadband, web cam, cell phone, and personal cameras. Users usually share the produced content in social networks. Although these sites have several recurrent collaborative features, there are very few solutions that allow the reuse of the code that implement these features.

Greenberg [3] positioned the collaborative systems development in the Replication phase of the BRETAM model [4], which describes how a technology evolves over time. During the Replication phase, researchers explore ideas of one another implementing, changing, and innovating. The process of building tools is not well

A.S. Vivacqua, C. Gutwin, and M.R.S. Borges (Eds.): CRIWG 2011, LNCS 6969, pp. 142–157, 2011.

established yet and there are lots of uncertainties and rework. Greenberg [3] argue that this stagnation in the Replication phase takes place because of the lack of toolkits that support reuse and encapsulation of technical complexities. This scenario illustrates the opportunity for a Domain Engineering (DE) to develop reusable software, reducing the need for redevelopment and keeping focus on systems assembly.

In this work, we present a domain engineering for collaborative features of content sharing social networks. We use the FODA [5] method extended with the 3C collaboration model [6] to classify, and interaction patterns to describe the collaborative features. The interaction patterns are based on a language for computer-mediated interaction proposed by Schummer and Luckosh [7]. To develop a component kit for the assembling of content sharing social networks, we used the Groupware Workbench [8] platform. We also conducted a case study and an experiment to evaluate the results.

The remainder of this paper is organized as follows. We present an overview about domain engineering in Section 2 and the related work in Section 3. Section 4 presents the proposed approach for domain engineering for collaborative features in content sharing in Web 2.0 social networks. The experiment and the case study are presented in Section 5. Section 6 concludes the paper.

2 Domain Engineering

Applications from the same domain have recurrent features that can be identified and analyzed to promote reuse. Aharoni and Berger [9] define domain as a set of applications that uses common concepts to describe requirements, problems, and capacities. The domain is bordered according to the similarities between applications [10]. Clements [11] defines Domain Engineering as a set of systems sharing a common and managed set of features that satisfy the specific needs of a particular market segment. There are many benefits provided by the reuse provided by a domain engineering, for example, the reduction of development costs and improvement of quality [12].

Alaña and Rodriguez [13] present a domain engineering process divided into domain analysis, domain design, and domain implementation. This division appears in others works as Blois and Becker [14] and Harsu [10]. The domain analysis is a process of identifying, collecting, organizing, and representing the relevant information of a domain, based on the study of existing systems and their development histories, knowledge captured from domain experts, underlying theory, and emerging technology [5]. The domain design uses results of the domain analysis to identify and generalize solutions for common requirements, by means of a software architecture specification [15]. According to Blois and Becker [14] this is an activity in which the project model is built based on the analysis model, the knowledge obtained studying the domain, and the reference architecture. The domain implementation includes the development of reusable components [15] [10]. This activity aims to build reusable software components based on the domain and on the reference architecture proposed in the domain design activity.

2.1 FODA (Feature Oriented Domain Analysis)

Many domain-engineering methods have been proposed. Alaña and Rodriguez [13] classify these methods in three different groups. Methods based on domain analysis aims to analyze and to model the domain for reusing similar parts. Some examples are ODM [16], FODA [5], FORM [17], FeatureRSEB [18], DARE [19], and Odyssey-DE [20]. Product Lines methods have been extended or connected to others to cover all the production process. Some examples are FAST [10] and PuLSE [21]. Domain Engineering and OOA/D combine domain engineering with object-oriented analysis and design. Some examples are OOram [22], JODA [23], and RiDE [24].

FODA [5] was developed at SEI (Software Engineering Institute). It is arguably the most well-known domain engineering method. It is based on the identification, analysis, and documentation of the system features, resulting in generic domain products widely applicable in a domain. The FODA method approach is relevant from the component-based development point of view, because the process of getting relevant features helps the identification of possible software components [15].

We used the FODA method because it is based on feature domain analysis and it is focused on the domain, rather than in the implementation and architecture details, as we do not have access to the internal implementation of the analyzed systems.

2.2 Extension of FODA for Collaborative Systems Analysis

In this work, the FODA method was extended with the 3C Collaboration Model and patterns for computer-mediated interaction to better support collaborative systems analysis. The first extension aimed to classify the collaborative features and the second one aimed to describe them.

Ellis et al. [6] used the 3C dimensions (communication, coordination, and cooperation) to classify the computational support for collaboration. This model is widely used in CSCW literature [25] [26] [27]. Communication involves exchange of messages and commitment negotiations. By means of the coordination, people, activities, and resources are managed to deal with conflicts and prevent loss of communication and cooperation efforts. Cooperation is the joint operation in a shared environment, generating, and manipulating cooperation objects [28].

In this work, we adopted the 3C collaboration model to classify and organize the collaborative features. Once classified, these features were mapped to software components, which are used to build collaborative applications based on communication, coordination, and cooperation requirements. We used the following criteria to classify features according to the 3C collaboration model:

- Communication: when the feature is used by people to send messages among them;
- Coordination: when the feature is used by people to manage themselves, or to become aware of the activities and its effects on the collaboration;
- Cooperation: when the feature is used by people to manage the shared space or to interact with shared artifacts.

Based on pattern language [29] and on software design patterns [30], Schummer and Luckosh [7] proposed patterns for computer-mediated interaction that describe recurrent support for group interaction based on the following structure: name, sensitizing picture, intent, context, problem, scenario, symptoms, solution, dynamics, rationale, check, danger spots, known uses, and related patterns. In this proposal, interaction patterns were used to describe the collaborative features.

3 Related Work

SPLCW2.0 is a domain engineering work for synchronous application [31]. According to Gaspar et al. [31] during the implementation of web applications for the Tidia-Ae project, many intersection points were found. Some of those points were easy to identify, for example, a generic communication service for synchronous messages exchange. Others points, for example, a rich window framework (RWIF) was identified after the building of an application set. The set of identified components was developed using the Tidia-Ae architecture and it enabled building several systems.

Gadelha et al. [32] present an approach to solve some problems identified in the development of Groupware Product Lines (GPL), which does not cover the domain analysis activity, does not encompass traceability, domain differences, and communalities. The main objective is to develop GPL with the benefits of Software Product Lines. The GPL domain analysis and requirements elicitation are based on the RUP 3C-Groupware [32].

No other work was found conducting a domain engineering in a groupware domain. However, there are related platforms that provide collaborative features for social network building, but these platforms are not based on components as described in CBE (Component Based Engineering) literature and its architecture and focus were not similar to this work.

Noosfero (www.noosfero.org) is a web platform for social and solidarity economy networks with blog, e- Porfolios, CMS, RSS, thematic discussion, events agenda and collective intelligence for solidarity economy in the same system. It has three kinds of profiles: person, communities, and enterprises.

Elgg (www.elgg.org) is open source social networking software that provides to individuals and organizations some components to build online social environments. Some features are blogging, user profile, recent activities, chat, social bookmark, photo, and video gallery.

4 Domain Analysis

The first activity of the FODA method is the context analysis, which defines the scope of a domain that is likely to yield exploitable domain products. In this work, the context is defined as “content sharing Web 2.0 social networks.”

A social network is defined as a group of people connected, in which users can build a public profile, possess a connected user list and navigate through it [33].

According to McCann [34], 2/3 of internet users use social networks. Their main uses are related to content sharing, which have been growing since the first studies in 2006. Between 2008 and 2009, 76% of users made photo upload and 33% video upload. In addition, the study shows that 96% of users visit their virtual friends profiles and 66% update their profiles [34].

4.1 Social Network Collaborative Features

Fig. 1 illustrates the identification of collaborative features at Facebook (www.facebook.com) following the collaborative features classification criteria, based on the 3C collaboration model. Rectangles are communication elements, ellipses, coordination, and arrows, cooperation. “Commenting” is a communication feature, “report abuse,” a coordination feature, and image “marking,” “sharing,” and “evaluation,” cooperation features.

Fig. 1. Mapped features based on the 3C collaboration model

We conducted this analysis in several social networks to identify the most present collaborative features (Fig. 2). We chose the social networks based on the Alexa (www.alexa.com) classification rank, which have a list of the most popular social networks, based on traffic analysis. As presented in Fig. 2, twenty-one collaborative features were identified. Only one communication feature was identified, four of

SERVICE	FEATURES	SOCIAL NETWORKS														
		FACEBOOK	GROUPSITE	ORKUT	FOTOLOG	DEVIANT ART	FOTOKI	PHOTOBUCKET	PICASA	FLICKR	YOU TUBE	HI5	XANGA	NETLOG	WINDOWS LIVE	FRIENDSTER
COOMMUNICATION	COMMENT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
COORDINATION	RECENT ACTIVITIES	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	SEARCH PEOPLE	X	X	X	X		X	X	X	X		X			X	X
	GROUPS		X		X	X		X	X	X	X		X	X	X	X
	REPORT ABUSE	X							X		X	X		X		X
COOPERATION	SHARED CONTENT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	STATISTICS				X	X		X		X	X					
	RATING					X		X			X			X		
	EXPORT									X	X		X	X	X	X
	DESCRIBE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	RECOMMEND	X	X	X	X	X	X	X	X	X	X		X	X	X	
	UPLOAD	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	MARK	X		X			X		X	X	X	X			X	X
	CATEGORIES				X	X					X					
	CONTENT SEARCH	X	X		X	X	X	X	X	X	X					
	PROMOTE	X		X					X							
	PLAYLIST OR ALBUNS	X	X	X	X	X	X	X	X	X	X	X	X		X	X
	FAVORITES					X	X			X	X	X	X		X	X
	NOTE										X					
	TAGS							X	X	X	X		X	X		
	PERMISSION	X	X	X	X	X	X		X	X	X		X	X	X	

Fig. 2. Content sharing collaborative features classification

coordination, in which the most recurrent was recent activity, and sixteen of cooperation, in which most recurrent were object sharing, description, and upload.

The next phase of FODA is the Domain Model development, which consists of three major activities: feature analysis, entity-relationship modeling, and functional analysis. The purpose of feature analysis is to capture, in a model, the end-user's (and customer's) understanding of the application general capabilities in a domain [5]. In Fig. 4, the collaborative features are represented as mandatory or optional. Their derivations are alternative and exclusive [32]. The entity-relationship model captures and defines the domain knowledge that is essential for implementing applications [5]. Fig. 3 presents the collaborative features class diagram and the relations among them.

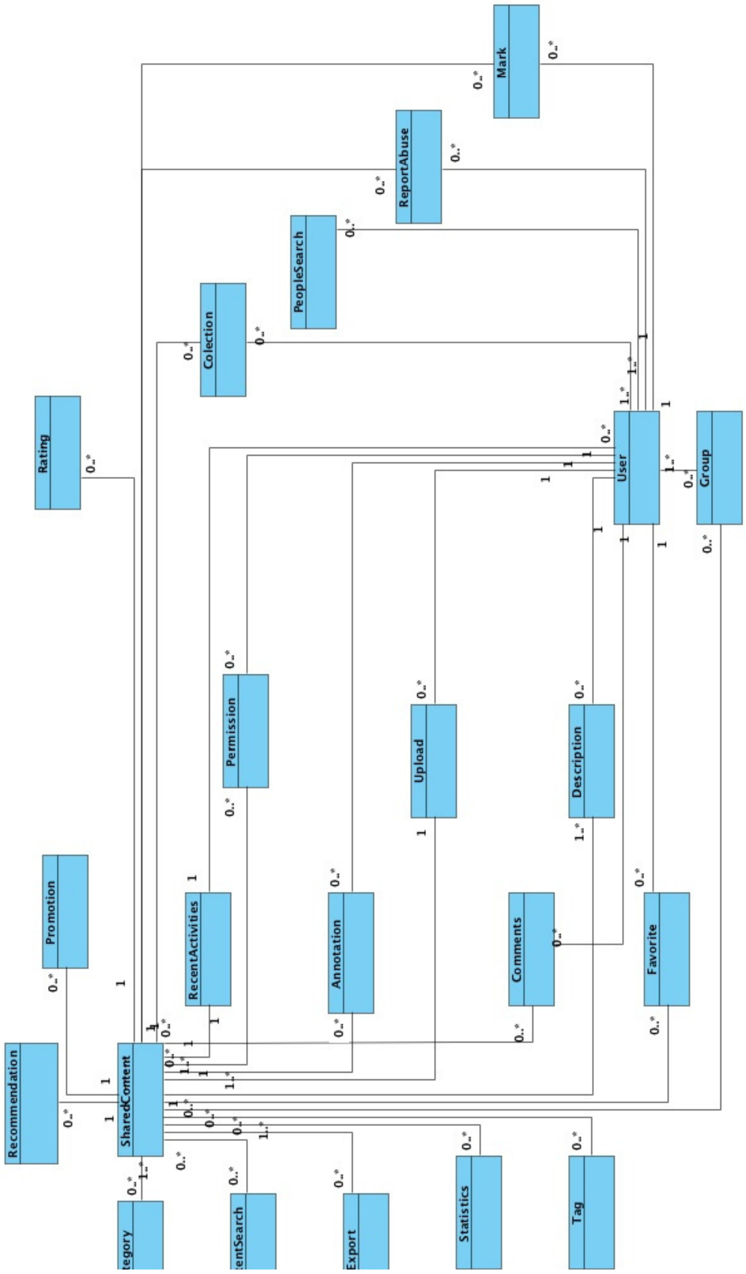

Fig. 3. Collaborative features class diagram

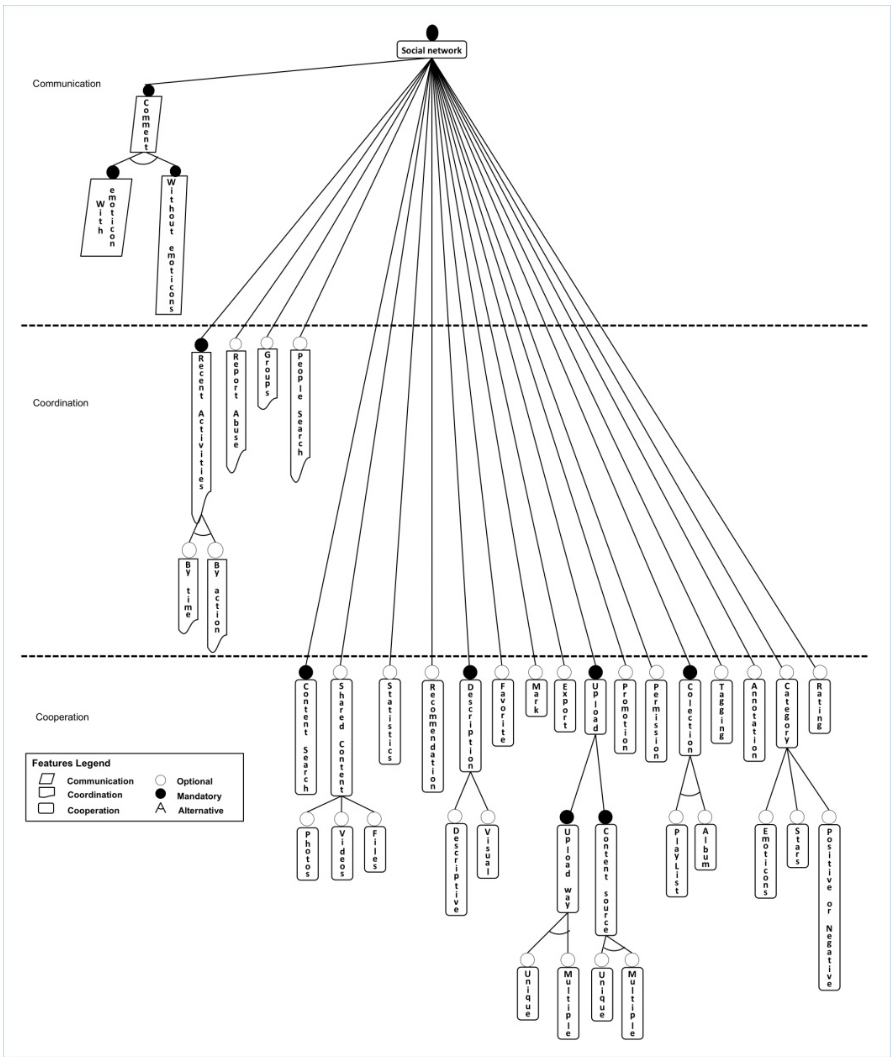

Fig. 4. Collaborative features tree

In this work, we represent functional differences and communalities using patterns for computer-mediated interaction [7]. The description of the “comment” pattern is summarized below. For space reasons, the other descriptions are available on the website http://www.groupwareworkbench.org.br/engenhariadedominio.

Commenting
Intention: enables a user to do an observation about a shared content.

Context: users are using a collaborative tool and wish to leave their opinion about a shared content.

Problem: users wish to leave their opinion about a shared content for other people, but the collaborative tool does not have a mechanism that enables them to leave such contribution.

Scenario: John is navigating in a social network, found an interesting image of a church of his home city, and decided to comment some historical building facts, but he does not found a place to leave his contribution.

Symptoms: You should consider applying this pattern when You wish users contributing, textually, about the content sharedYou wish discussions about the shared content.

Solution: To integrate a comment mechanism that enable users to leave their contributions and to begin a discussion about the shared content.

Dynamics: A user writes a text on a specific field for comments. After that, they should click on send message button, and their contribution appears on the discussion thread. Other users agree or disagree about an opinion by means of reply.

Rationale: the commenting pattern provides users an easy and fast textual commenting mechanism.

Check: When applying this pattern, you should answer these questions:
Does the application should enable comments reply?
Where will it be placed?
Will it have textual resources only?

Danger Spots: You should take care of the text that will be commented and how it will be processed. People could insert malicious source code on the comment, affecting system stability and safety.

Known Users: Facebook, Orkut, Fotolog, and DeviantArt.

Related patterns:
Forum: describes how users could discuss textually about a specific topic.
Discussion Thread: Help people to find related messages when a user replies an existent comment.
Annotation: Enables user to create notes or comments about a shared content.

4.2 Domain Design and Implementation

The last phase of the FODA method is architecture modeling, which provides a software solution to the issues identified in the domain modeling activity. An architecture model (also known as reference architecture) is developed in this phase.

In this work, we used the Groupware Workbench platform [8] as reference architecture and component development tool. It provides a component kit and infrastructure to support the development of Web 2.0 collaborative applications. Component technology has been largely used to support the construction of groupware applications.

Groupware Workbench uses components called *collablets*. Java class composes each collablet, namely: entities, a business interface implementation, and controllers. In the presentation layer, it uses Java server pages (JSP) and some graphical interface widgets encoded as tagfiles. Each application has a main collablet, which represents an entry point and manages all other components by means of subordination and dependency. Advantages of Groupware Workbench are: support to components development, an infrastructure to build and deploy the applications based on the domain, and a reference architecture to build the components and applications. In addition, it enables application composition by means of a graphical interface, which recognizes the installed components and enables to compose applications without any Java code alteration.

In this work, we adopted the mapping one-to-one between the identified feature in the domain analysis and the corresponding software component. As an implementation example, the implementation of the commenting feature as the collablet commentMgr is presented in Fig. 5.

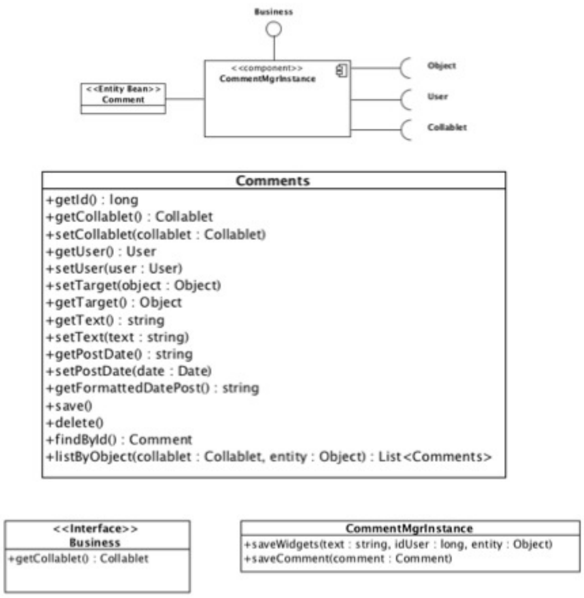

Fig. 5. CommentMgr component

The commentMgr component implements two interface widgets: one for assigning a comment from a user about content, and another to return a list of assigned comments. The widgets are inserted into the collablet JSP. At the bottom of Fig. 6, an example of a commentMgr widget is shown. Along with the commentMgr component, the collablets tag, binomialMgr, descriptionMgr, rating, and photoMgr are composing this interface.

The collaborative feature variabilities are treated in this work by means of interface widgets or XML configuration. For instance, for the emoticon-enabled commentaries, component variability is needed to implement an interface widget that provides its characteristic, without the necessity for component infrastructure changes.

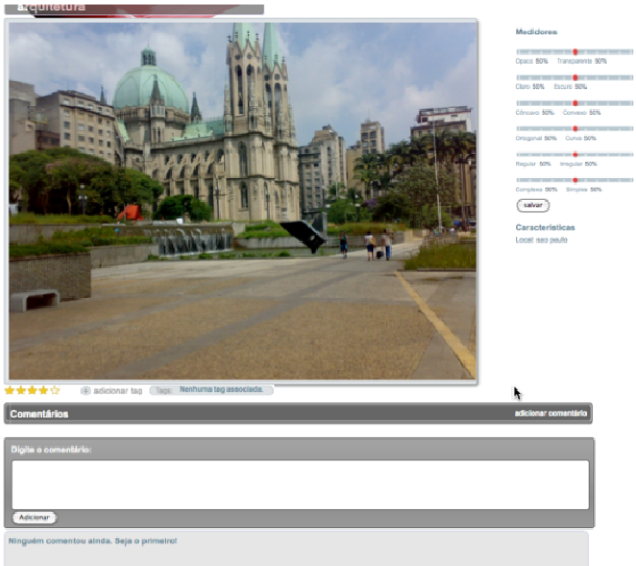

Fig. 6. Arquigrafia Brasil comment component

5 Evaluation

An experiment and a case study were conducted to evaluate the domain. The first one evaluated the artifacts according to ease of use, utility, and understanding. The second one evaluated the application of its artifacts in a real context.

5.1 Experiment

To evaluate the domain engineering, an experimental study divided into two phases was done: one to evaluate the feature model coverage and the patterns description, and other to evaluate the artifacts, ease of use, and utility. The experiment was performed with 10 available under graduation and graduation students of Web development course, with some software development experience. They participated in the experiment near the end of the course.

During the experiment, participants filled out a questionnaire containing questions about educational and social network experiences. After that, they had to evaluate and identify the collaborative features at PhotoBucket, Flickr, and Netlog social networks, as they contained a large number of collaborative features (Fig. 2), within 30 minutes. At the end of these activities, a list containing seven collaborative features with its respective patterns descriptions was presented. They had to read and identify the features in social networks; in addition, they had to evaluate the ease of use, detailing, and ease of identification of the patterns.

After that, they were divided into two groups and they had to add two collaborative features (commentaries and tags) in two applications: a FAQ (frequently asked questions system) developed with the Groupware Workbench and another FAQ developed in a traditional web development way. The groups were inverted after the first hour and ten minutes or when they finished the activity. At the end of the activities, they answered a questionnaire containing questions related to the artifacts and to the process. According to the GQM [39] framework the experiment goal was:

Analyze the domain engineering
For the purpose of evaluation
With respect to coverage, ease of use, and utility
From the viewpoint of collaborative systems developers
In the context of students of Computing Science course

Questions
Q1: Are there collaborative features that were not identified in the domain engineering?
Q2: Do developers understand the artifacts generated in the domain engineering?
Q3: Can developers build a new system using the generated artifacts?
Q4: Do developers consider useful the generated artifacts?

Metrics
Metric M1 – (Q1): the set of features identified by developers that were not identified in the domain engineering
Metric M2 – (Q2): the percentage of interaction patterns evaluated as easy to understand
Metric M3 – (Q2): the percentage of interaction patterns evaluated as easy to identify in others social networks
Metric M4 – (Q3): the number of developers who performed the proposed activity within a time of two hours and twenty minutes with at least two components
Metric M5 – (Q3): the percentage of developers that positively evaluated the artifacts as easy to use
Metric M6 – (Q4): the percentage of developers that evaluated the artifacts as useful

Analyzing the collected data, we got:

For Question 1, the developers did not found new collaborative features in the evaluated social networks. For Question 2, 92% percent of collaborative features are easy to understand, and 87% are easy to identify in others social networks, but some developers pointed the following interaction patterns as hard to understand: description, statistics, upload, mark, and category.

For Question 3, all developers performed the experiment, but less than 80% evaluated as easy to use the generated artifacts. For Question 4, almost all developers evaluated the artifacts as useful, only one evaluated as neutral.

Based on the experiment results, we conclude that the developers had identified the same collaborative features of this work, they can understand the feature description, and classified them as useful, but the usability needs to be improved.

5.2 Case Study: Social Network Arquigrafia Brasil

The Arquigrafia Brasil (www.arquigrafia.org.br) is a social network that aims to offer a huge public collection of Brazilian architecture digital images, not existing yet, assigned and cataloged by users. In this case study, we verify if the components produced in the domain engineering are sufficient to build a social network.

Some independent researchers conducted a focus group for usability study with possible users, students, photographers, and architects, in which some image sharing features for a possible architecture social network were identified. Some of these features can be mapped into features previously identified in this work, as shown in Fig. 7.

	Features (tool or services)	Necessity Scale:1(=less necessary) a 4(=very necessary)	Justification	correspondet collaborative features	3C collaboration model dimension
23	directory criation in personal area, all registres ordered by date talling a story	4	Focus Group	collection	cooperation
6	Videos to registry the architecture evolution, talling a evolutionary story	3	Focus Group	shared content	cooperation
10	Theme photos visualization	4	Focus Group	shared content	cooperation
6	Information about the image(where, who, historical context)	4	Focus Group	description	cooperation
14	Google Earth connection	2	Focus Group	description	cooperation
19	Photo search by theme, historical period, etc.	4	Focus Group	content search	cooperation
21	shared images access control(privace)	3	Focus Group	permission	cooperation
27	security on image storage: persist users notification, abuse, etc.	4	Focus Group	report abuse	coordination
28	comment area about project, image, group shared image, and social network shared images.	4	Focus Group	comment and group	communication, cooperation
31	visualization by categories as regional architecture.	4	Focus Group	category	cooperation

Fig. 7. Focus group identified features and its correspondent collaborative features

Independent researchers also conducted brainstorm meetings for desired features identification. The desired features was classified according to the 3C collaboration Model and compared to the identified collaborative features of this work, resulting in 12 matched features.

We conclude that the identified features in this domain engineering cover a large part of the features for building the Arquigrafia Brasil. However, some specific components were not identified in this domain engineering such as binomial ones (which enable image evaluation), because this component is specific for the Arquigrafia Brasil project.

6 Conclusion

The Web 2.0 increased the possibility of socialization and expression by means of computer-mediated tools. Users not only modify the web pages content, but help to organize, share, criticize, and update it.

We proposed a domain engineering for building collaborative software components, reducing the reimplementation necessity and focusing on the system assembly. For this work, the chosen domain was Web 2.0 social networks, focusing on the content sharing collaborative features.

To perform this domain engineering, we did an adaptation of the FODA method with the 3C collaboration model to classify and patterns for computer-mediated interaction to describe the collaborative features. The Groupware Workbench was used in the activities of domain design and domain implementation to develop the collaborative software components. Although, these components are not limited to the domain of social network, they can be used in other domains.

To evaluate the artifacts generated in this domain engineering, we performed an experiment and a case study. The first one evaluated the artifacts ease of use, utility, and understanding; the second one evaluated the application of its artifacts in a real context: the social network Arquigrafia Brasil. Developers evaluated the artifacts as useful and easy to understand. For the case study, we concluded that the performed domain engineering covered the features previously identified in the focus group and brainstorms.

This work also proposed an approach to perform a domain engineering for collaborative systems, adapting the FODA method with 3C collaboration model and patterns for computer-mediated interaction. In addition, it provided a component kit that enables to build new collaborative social networks in the Web 2.0, increasing the software reuse.

References

1. O'Reilly, T.: What is Web 2.0: Design patterns and business models for the next generation of software (2010), http://oreilly.com/web2/archive/what-is-web-20.html
2. Prescott, L.: Hiwise US consumer generated media report., Hitwise, fev. (2007)
3. Greenberg, S.: Toolkits and interface creativity. Springer Science + Business Media (2007)
4. Gaines, B.: Modeling and forecasting the information sciences. Information Sciences 57/58, 13–22 (1999)
5. Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, A.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. CMU/SEI (1990)
6. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware - Some Issues and Experiences. Communications of the ACM 42 (1991)
7. Schummer, T., Lukosch, S.: Patterns for Computer-Mediated Interaction. John Wiley & Sons Ltd., Chichester (2007)
8. Groupware Workbench: Groupware Workbench (2010), http://www.groupwareworkbench.org.br (accessed 2010)
9. Aharoni, A., Reinhartz-Berger, I.: A Domain Engineering Approach for Situational Method Engineering. University of Haifa (2008)

10. Harsu, M.: A survey on domain engineering. Institute of Software Systems Tampere University of Technology (2005)
11. Clements, P.: FAQs: An Introduction to Software Product Lines (accessed 2005)
12. Pohl, K., Böckle, G., Der, F.: Software Product Line Engineering, Foundations, Principles, and Techniques. Springer, Heidelberg (2005)
13. Alaña, E., Rodriguez, A.: Domain Engineering Methodologies Survey. GMV Inovvating Solutions (2007)
14. Blois, A., Becker, K.: A Component-Based Architecture to Support Collaborative Application Design. In: Proceedings of the 8th International Workshop on Groupware: Design, Implementation and Use, London, pp. 134–146 (2002)
15. Werner, C., Braga, R.: A Engenharia de Domínio e o Desenvolvimento Baseado em Componentes. Editora Ciência Moderna, Rio de Janeiro (2005)
16. Simos, M.: Organization domain modeling and oo analysis and design: Distinctions, integration, new directions. In: STJA 1997 Conf. Proc., pp. 126–132 (1997)
17. Kang, K., Kima, S., Lee, J., Kim, K., Shin, E., Huh, M.: FORM: A feature-oriented reuse method with domain-specific reference architectures. Annals of Software Engineering 5 (1998)
18. Griss, M., Favaro, J., d'Alessandro, M.: Integrating Feature Modeling with the RSEB (1998)
19. Frakes, W., Prieto-Diaz, R., Fox, C.: Dare: Domain analysis and reuse environment. Ann. Softw. Eng., 125–141
20. Braga, R., Werner, C., Mattoso, M.: Odyssey-Search: A multi-agent system for component information search and retrieval. Journal of Systems and Software 79, 204–215 (2006)
21. Bayer, J., Flege, O., Knauber, P., Laqua, R., Muthig, D., Schmid, K., Widen, T., DeBaud, J.-M.: Pulse: a methodology to develop software product lines. In: Proceedings of the 1999 Symposium on Software Reusability, pp. 122–131 (1999)
22. Hoeydalsvik, G.: OORAM: Object-Oriented Role Analysis and Modeling. Wiley-QED Publishing, Somerset (1994)
23. Holibaugh, R.: Joint Integrated Avionics Working Group (JIAWG) Object- Oriented Domain Analysis Method (JODA). Carnegie Mellon University (1993)
24. Almeida, E.: RiDE: The RiSE Process for Domain Engineering. Ph.D. Thesis in Computer Science, UFPE (2007)
25. Castellani, S., Ciancarini, P., Rossi, D.: The ShaPE of ShaDE: a Coordination System., Technical Report UBLCS (1996)
26. Sauter, C., Morger, O., Muhlherr, M., Thutchytson, A., Teusel, S.: CSCW for Strategic Management in Swiss Enterprises: an Empirical Study. In: 4th European Conference on Computer Supported Cooperative Work (ECSCW 1995), Stockholm, Sweden, pp. 117–132 (1995)
27. Escovedo, T., Melo, R.N.: Applying a methodology for collaborative assessment in learning groups. 2010 IEEE Education Engineering (EDUCON), 1205–1210 (2010)
28. Gerosa, M.A., Pimentel, M.G., Fuks, H., de Lucena, C.J.P.: Development of Groupware Based on the 3C Collaboration Model and Component Technology. In: Dimitriadis, Y.A., Zigurs, I., Gómez-Sánchez, E. (eds.) CRIWG 2006. LNCS, vol. 4154, pp. 302–309. Springer, Heidelberg (2006)
29. Alexander, C., Ishikawa, S., Silvertein, M.: A pattern language. Oxford University Press, Oxford (1977)
30. Gamma, E., Helm, R., Viissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)

31. Gaspar, T., Yaguinuma, C., Do, A.: Software product lines for Web 2.0 synchronous collaboration. In: WebMedia 2009 Proceedings of the XV Brazilian Symposium on Multimedia and the Web (2009)
32. Gadelha, B., Nunes, I., Fuks, H., Lucena, C.J.P.: An Approach for Developing Groupware Product Lines Based on the 3C Collaboration Model. In: 15th Collaboration Researchers' International Workshop on Groupware, pp. 328–343 (2009)
33. Kazienko, P., Musial, K.: Social Capital in Online Social Networks, pp. 417–424 (2006)
34. Universal McAnn: Power to the people - social media tracker wave 4. Universal McAnn (2009)
35. Grudin, J.: Why groupware applications fail: Problems in design and evaluation. Information Technology & People 245
36. Hill, J., Gutwin, C.: He maui toolkit: Groupware widgets for group awareness. Computer Supported Cooperative Work (CSCW), 539–571 (2004)
37. Biemans, R.: Component groupware: a basis for tailorable solutions that can evolve with the supported task. In: Proceedings of the International ICSC Conference on Intelligent Systems and Applications (2000)
38. Wulf, V., Pipek, V., Won, M.: Component-based tailorability: Enabling highly flexible software applications. International Journal of Human-Computer Studies, 1–22
39. Solingen, R., Berghout, E.: The Goal/Question/Metric Method A Practical Guide for Quality Improvement of Software Development. McGraw-Hill, New York (1999)