REUSES: Questionnaire-driven design for the automatic generation of Web-based Collaborative Applications

Ruth Cobos, Rafael Martín, Jaime Moreno-Llorena, Esther Guerra and Juan de Lara.
Department of Computer Science, Universidad Autónoma de Madrid, Spain
{Ruth.Cobos|Jaime.Moreno|Esther.Guerra|Juan.deLara}@uam.es, rafael.martinj@estudiante.uam.es

Abstract— **In this paper, we present a questionnaire-driven collaborative design methodology for the automatic generation of collaborative applications. The methodology is based on a repository of collaborative components and a knowledge base of its application and use modes. The repository includes components built by our team, as well as third-party components that use heterogeneous technologies like JavaScript, Java, Flash, and services and components provided by companies like Google and Facebook. Based on this methodology, we have built a Web 2.0 tool called REUSES (Rapid End-User Synthesis of Collaborative Applications), which provides facilities for the automatic generation of Web-based collaborative applications tailored to any particular community of end users. Finally, we also report on some experimental work to test REUSES in the context of academic courses at our University.**

Keywords-Collaborative Design; Web-based Collaborative Applications; Mashups; World Wide Web; Collaborative Components.

I. INTRODUCTION

Collaborative work is becoming increasingly used in academia and industry. Therefore, there is an important demand for specific, tailor-made collaborative applications supporting the group work particularities of each user community.

A great amount of collaborative applications exist nowadays [1]-[4]. However, depending on the user community requirements, the choice of the application that provides the best software solution is not easy. Currently, such solutions are usually provided as a set of web services from major Internet companies like Google or Yahoo, or they are built with the help of Web Content Management Systems, or tools for creating and maintaining Weblogs and Wikis. In most cases, the end-users of these applications have to configure their collaborative solution without enough proper advice and knowledge.

Collaborative application design is a complex task [5][6] that must involve both the final users and the software experts. Thus, having a tool that facilitates the design and implementation of collaborative applications taking into account the experience of experts and the participation of the final users seems to be a good approach.

A collaborative application can be seen as an integration of collaborative components that support a set of collaborative features. This proposal was used in [1] in order to classify collaborative applications according to the collaborative features they supported, such as bulletin boards, messaging, audio/video conferencing, contact management, and so on. From this perspective, the idea of having a repository of collaborative components and a knowledge base of its application and use modes for different kinds of collaborative applications is a good starting point towards the achievement of the above mentioned tool.

The Web evolution in the last decade into the known Web 2.0 [7] and the recent proliferation of "*Web 2.0 APIs*" – provided by Google, Yahoo and Facebook, among others – have contributed to make the Internet increasingly more social. In parallel with these APIs, the *mashup* technology has emerged to support web pages or applications in the use and combination of data, presentation and functionality from two or more sources to create new services [8]-[10]. One example of a mashup application is *Woozor* (http://woozor.com/), which integrates *Google Maps* (http:// maps.google.com/) and *Weather.com* (http://www.weather.com/). Another example is *Portwiture* (http://portwiture.com/) that integrates *Flickr* (http://www.flickr.com/) and *Twitter* (http://twitter.com/).

Taking into account this previous discussion, the Web 2.0 is the proper platform where collaborative applications are built, thanks to its popularity, easy use, the great amount of existing resources and the technology available for developing new resources and integrating them in new applications. Unfortunately, the diversity and heterogeneity of the involved technologies makes complicated their direct use even by professional software engineers.

In this work we present a novel methodology for the automatic generation of collaborative applications, which deals with a repository of collaborative components and a knowledge base of its application and use modes. The methodology is supported by a Web 2.0 tool called *REUSES* (**R**apid **E**nd-**U**ser **S**ynthesis of Collaborativ**E** Application**S**), which helps the community of end-users in building a tailor-made Web-based collaborative application addressing their particular needs. For this purpose we follow a guided, adaptive, questionnaire-driven approach to collect and contrast

different user requirements. Then, we use a Model-Driven Engineering approach, which is provided by our own code generator tool called MetaDepth [11], to translate these requirements into an intermediate mashup-like representation, closer to the technological solution domain, from which the code of the final collaborative application is automatically synthesized.

The rest of the paper is organized as follows. Section II shows a general overview of our process for generating web-based collaborative applications. Section III details the proposed questionnaire-driven methodology. Section IV deals with tool support, presenting both the REUSES tool and the MetaDepth environment and code generator. Section V illustrates the approach with a case study. Section VI compares with related work and Section VII ends with the conclusions and lines for future work.

II. THE REUSES COLLABORATIVE PROCESS FOR THE GENERATION OF COLLABORATIVE WEB APPLICATIONS

Our REUSES tool supports a process for the collaborative generation of collaborative web applications by a user community in three phases, which are summarized in Figure 1.

In the first phase, REUSES provides each user in the community with an adaptive questionnaire for establishing the requirements that the application should include, in his/her opinion. These requirements are synthesized in his/her *User Collaborative Application Model* (UCAM). A UCAM is a vector of Boolean values (with '1' for 'yes' and '0' for 'no') of the form:

$$UCAM = (cc_1, cc_2, \dots, cc_m) \qquad (1)$$

where the size m of the vector is the number of components in the REUSES repository, and each vector component cc_i represents the user opinion about the necessity of having a concrete component (e.g. Google Talk or Blogger) in the collaborative application.

The users do not directly fill this UCAM, as they may not have the knowledge to select which component is more suitable for their needs, and moreover, several components may implement similar or overlapping functionality. Instead, REUSES presents adaptive questionnaires to interrogate the user community about their collaborative *objectives* and their collaborative *tasks*. The flow of questions presented to each user is adapted by a set of rules, which are generated by the conversion of a decision tree. From the answers to these questionnaires, the UCAMs of the different users are automatically derived using the procedures and heuristics that we present in Section III. Therefore, we can say that REUSES supports a questionnaire-driven design.
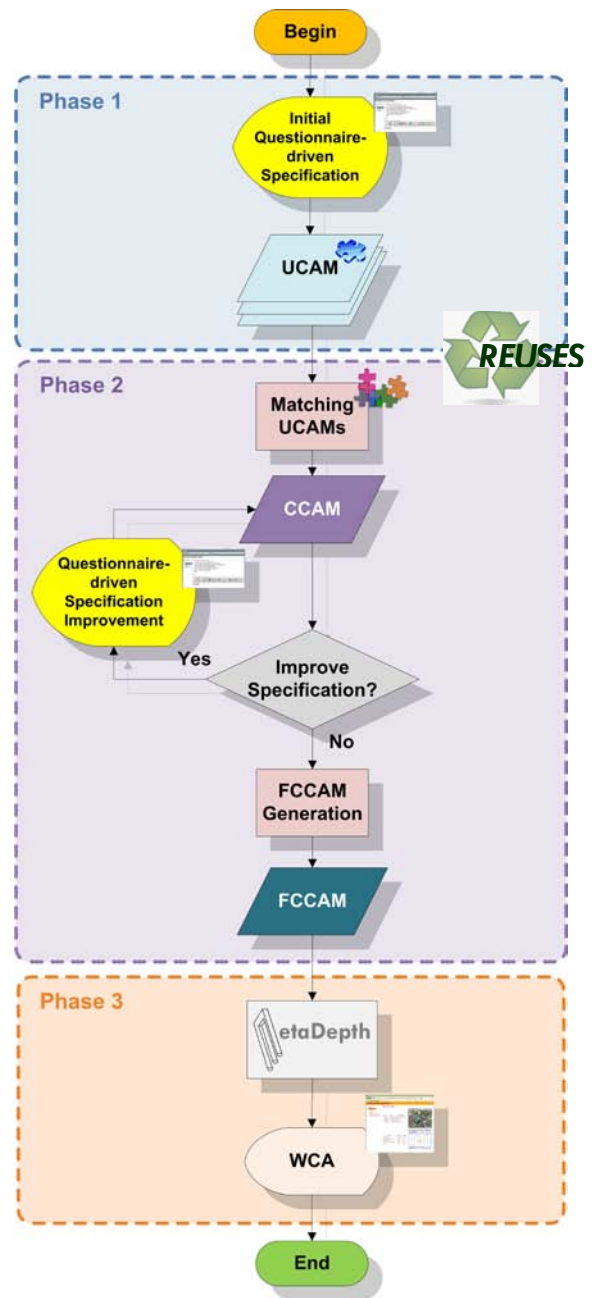

Figure 1. Generation process of a *Web Collaborative Application* (WCA) supported by REUSES.

The second phase starts once REUSES has obtained the UCAMs of all the users of the community. At this moment, REUSES aggregates different user choices in order to obtain a collective view of the community preferences. Hence, it performs the average of the UCAMs shown in (2) in order to calculate the *Community Collaborative Application Model (CCAM)*, which represents the collective opinion about the requirements of the collaborative application.

$$CCAM = \sum_{i}^{N} \frac{UCAM_i}{N} \qquad (2)$$

In Equation (2), N is the number of users in the community, and the resulting CCAM is a vector of size m with real values between 0 and 1.

The goal in this phase is to drive the community towards a consensus on the necessary components. For this purpose, REUSES generates a prototype of the application represented by the initial CCAM, and allows the user community to improve this prototype through a refinement process where each user can answer another questionnaire. The aim is generating a *Final CCAM* (FCCAM) with the community consensus about the requirements of the collaborative application. This FCCAM is a vector of Boolean values indicating the components that have to be supported by the collaborative application tailored by the user community.

In the third and last phase, REUSES invokes the code generator tool, MetaDepth, in order to synthesize the final collaborative application from the FCCAM. This application is deployed so that the user community can access it through a URL. For more details, see Section IV.

III. THE QUESTIONNAIRE-DRIVEN COLLABORATIVE DESIGN METHODOLOGY

The initial point of the present research work was an exhaustive review of different studies, classifications and taxonomies of collaborative systems and applications [1]-[4]. Based on them, we studied 79 collaborative applications (e.g. BSCW http://public.bscw.de/ and Projecturf http://www.projecturf.com/) which can be classified in one of the six selected application types (i.e. we studied about 13 collaborative applications per type). With this information we generated a table relating the collaborative applications with the collaborative features they provide, focusing in 18 collaborative features like messaging or document management. This table was called *AFM* (Application-collaborative Feature Matrix). Next, we generalized our findings in a new table where we related the same collaborative features with the following six application types: Project Management, Document and File Group Handling, Computer Conferencing, Electronic Meeting Systems, Electronic Workspace and Online Social Network. This table, which was called *ATFM* (Application Type-collaborative Feature Matrix), aggregates the averages of the features of each application of the same type. The matrix is calculated as follows:

$$ATFM(AT, f) = \sum_i \frac{AFM(AT_i, f)}{n} \qquad (3)$$

where n is the number of collaborative applications of type AT, AT_i represents a collaborative application of type AT, and f is a selected collaborative feature.

Each ATFM value represents the probability to have a particular feature in a specific application type. For instance, we obtained *ATFM(Project Management, Document*

Management)=1, because all the studied Project Management applications provide the feature (i.e. document management), and *ATFM(Electronic Workspace, Messaging)=0.7*, because 70% of the studied Electronic Workspace applications provide messaging. More details about this previous study are available in [12].

However, as we mentioned above, our proposed methodology deals with collaborative components, not features, because applications are component-based. A component can support one or more features (e.g. Google Talk supports Messaging, Synchronous Discussion and Audio/Video Conferencing) and several components may have overlapping features.

For this reason, we created a new table where we relate collaborative components with their supported features, which was called *CFM* (Component-collaborative Feature Matrix).

Taking into account this previous research work, we present in the next sections the methodology parts.

A. Basic Models for Collaborative Application Types

The REUSES process needs an initial model of the application types in terms of the collaborative components. This model realization generates the Basic Models (BMs), which are vectors of real values between 0 and 1, and each one indicates the necessity of having a concrete component in the application type. They are calculated as shown in (4), where Var(AT,c) is a variance, as in (5), and Ave(AT,c) is an arithmetic mean, as in (6).

$$BM(AT, c) = min\big(1, Var(AT, c) + Ave(AT, c)\big) \qquad (4)$$

$$Var(AT, c) = \frac{\sum(ATFM(AT, CFM(c)_i) - \frac{\sum ATFM(AT, CFM(c)_i)}{|CFM(c)|})^2}{|CFM(c)| - 1} \qquad (5)$$

$$Ave(AT, c) = \frac{\sum ATFM(AT, CFM(c)_i)}{|CFM(c)|} \qquad (6)$$

In the Equations (5) and (6), CFM(c) is the set of features supported by the collaborative component *c*. For instance, *BM(Project Management, Google Cabinet)=1*, because the Google Cabinet component is necessary for any application of type Project Management, and *BM(Project Management, Google Talk)=0.5*, because the Google Talk component is necessary with a probability of 0.5 for an application of this type, see Table I.

These BMs are used to dynamically adapt the questionnaire presented to the users of the community, for a selected application type, as follows:

- if BM(AT,c)>=0.8 then the component c will be included in the collaborative application, so that it is not necessary to ask about its inclusion.

- if 0.2<BM(AT,c)<0.8 then including the component c in the application will depend on the user community opinion, therefore it is necessary to ask about it in the questionnaire.

- If BM(AT,c)<=0.2 then the component c won't be included in the collaborative application, so that it is not necessary to ask about its inclusion.

Our previous research work, on classifying 79 collaborative applications has shown that most applications of the same type have in common components with a BM value higher than 0.8. In contrast, the components with a BM value between 0.2 and 0.8 may be included in applications of the same type, but they aren't typically associated with the type.

TABLE I
BASIC MODELS (BMs), WHICH INDICATE THE NECESSITY OF HAVING A SPECIFIC COMPONENT IN EACH APPLICATION TYPE

CC x AT*	Project Mana-gement	Document & File Group Handing	Computer Conferen-cing	Electronic Meeting Systems	Electronic Workspace	Online Social Network
Login	1.0	1.0	0.9	1.0	1.0	1.0
GoogleCabinet	1.0	1.0	0.0	0.0	0.6	0.0
GoogleTalk	0.5	0.5	0.8	0.7	0.5	0.6
GoogleCalendar	1.0	0.4	0.8	0.9	0.6	0.4
RSS	0.1	0.9	0.0	0.0	0.0	1.0
GoogleContacts Data	0.6	0.6	0.3	0.7	0.4	1.0
Blogger	0.6	0.8	0.0	0.1	0.2	0.6
GoogleGroups	0.9	0.9	0.5	0.4	0.8	1.0
Opensocial	0.6	0.7	0.8	0.9	0.7	0.8
GoogleSearch	0.5	1.0	0.0	0.0	0.0	0.0
GoogleDocs	0.6	0.4	1.0	0.7	0.5	0.3
Gmail	0.9	0.9	0.5	0.6	0.7	1.0
Google Poll	0.3	0.5	0.7	0.7	0.1	0.1
Historic	0.2	0.8	0.0	0.8	0.4	0.0
GoogleCharts	0.6	0.0	0.0	0.0	0.0	0.0
GoogleTable	0.6	0.0	0.0	0.0	0.0	0.0
GoogleMaps	0.6	0.0	0.0	0.0	0.0	0.0

*CC x AT, rows are Collaborative Components and columns are Application Types.

Finally, we also need to model any particular collaborative application according to the components using (7), where similarly to the previous case Var(a,c) is a variance, as in (8), and Ave(a,c) is an arithmetic mean, as in (9).

$$BM(a, c) = \min\big(1, \text{Var}(a, c) + \text{Ave}(a, c)\big) \quad (7)$$

$$Var(a,c) = \frac{\sum (AFM(a, CFM(c)_i) - \frac{\sum AFM(a, CFM(c)_i)}{|CFM(c)|})^2}{|CFM(c)| - 1} \quad (8)$$

$$Ave(a,c) = \frac{\sum AFM(a, CFM(c)_i)}{|CFM(c)|} \quad (9)$$

where a is the specific application. We will use this BM for a collaborative application in the next section.

B. Collaborative Components Correlations

In order to shorten the length of the questionnaire presented to the users and to avoid asking non-relevant or redundant questions, we perform some reasoning on the features of the collection of components. In this way we are able to know if with the inclusion of a specific component c_1 in the application

we do not need to take into account another component c_2, because c_1 and c_2 realize the same features.

For this purpose, it is necessary to calculate the correlation of all the components in our repository. Hence, we have applied the *Pearson Correlation* among the BMs of all the studied collaborative applications, between each two components c_1 and c_2, as shown in (10), where A is given by (11), and B(c) is as in (12).

$$CrrCC(c_1, c_2) = \frac{A}{B(c_1)B(c_2)} \quad (10)$$

$$A = n \sum BM(a_i, c_1)BM(a_i, c_2) - \sum BM(a_i, c_1) \sum BM(a_i, c_2) \quad (11)$$

$$B(c) = \sqrt{n \sum BM(a_i, c)^2 - \left(\sum BM(a_i, c)\right)^2} \quad (12)$$

where a particular application is denoted by a, and the total number of applications is n (actually we have 79 applications in our case). For instance, *CrrCC (GoogleCharts, GoogleTable) =1, CrrCC(Blogger, Google Groups) = 0.74 and CrrCC(Google Calendar, Google Talk)= 0.04*.

The calculated correlation matrix (CrrCC) in (10) is used to generate clusters of components, which we call the *CCC* matrix (Collaborative Component Clusters). Each two components c_1 and c_2 are in the same cluster when *CrrCC(c_1,c_2)>0.22*. The value 0.22 was obtained empirically, because after the creation of the CCC matrix, we observed that the clusters generated with this restriction were the best possible ones. An example of CrrCC matrix can be seen in Table III of Appendix.

C. Questionnaire Construction

As we mentioned above, REUSES supports a questionnaire-driven design. Each user has the opportunity of tailoring the collaborative web application through his/her answers to a questionnaire, which is dynamically adapted thanks to a set of rules that manage the questions' flow.

The questions are presented to the user following these steps:

1. The first question is the application type that the user prefers for his/her community work. REUSES can generate automatically six application types, which are modeled with its BMs in Equation (4). With this first choice, the BM associated to the application type is selected. Then, the following three elements are initialized:

   - UCAM: the components in the BM with value higher than 0.8 are assigned a value of 1 in the corresponding UCAM dimensions (see Section II);

   - We create a set NC, with the components that have values between 0.2 and 0.8 in the BM; and

- F: the set of features that are supported by the components in NC, but are not supported by the components in UCAM at the moment. These are the features we need to ask about.

2. The components contained in NC are extracted one by one. This extraction follows these steps:

    - First, we extract from F the feature with the highest ATFM value, which is denoted by *fe*. Recall that the higher the ATFM value is, the most likely the feature is necessary in the application type.

    - Second, from the components in NC that can support feature *fe*, we extract the component supporting another feature in F having the highest ATFM value at the moment. This selected component is denoted by *co*. This will be the most useful component if the user wants to include the feature. Furthermore, the user is asked whether he/she wants to include the feature *fe* in the Web application.

3. If the user answers that *fe* is needed, then the UCAM is updated, assigning a value of 1 to the corresponding dimension of *co*, and the features supported by *co* are extracted from set F and retained.

4. If the user answers that *fe* is not needed, it is examined if *co* is not in any cluster of the assigned components in UCAM. In this case, all the components in the same cluster as *co* are removed from NC, because they are similar to *co* (in this way, the number of questions are reduced).

5. This process continues until the NC set is empty.

These steps are summarized in Table II.

IV. TOOL SUPPORT

In this section, we present the two tools we use to generate the collaborative applications: the Web 2.0 REUSES Tool and the MetaDepth Framework. A REUSES prototype is available at *http://mcolab.ii.uam.es.* More detailed information about MetaDepth is available at *http://astreo.ii.uam.es/~jlara/ metaDepth/*.

A. Web 2.0 Reuses Tool

REUSES is a Web 2.0 Tool, which is composed by these three related modules (see Figure 2):

- The Decision Manager, which coordinates the process needed to generate the collaborative application and facilitates the communication among the other REUSES modules;

- The Specifications Manager, which constructs the adaptive questionnaires based on the presented set of rules;

- The Repository of Collaborative Components, which contains both components developed by our team and external components, like Google Cabinet and Google Calendar.

TABLE II
QUESTIONNAIRE ADAPTATION ALGORITHM

```
While Q = No
   Q ← Question about Application Type
   If Q = Yes
       AT ← answer(Q)
       For (c'; BM(AT,c')>0.8)
          UCAM ← UCAM U {c'}
       Endfor
       For (c''; 0.2<BM(AT,c'')<0.8)
          NC ← NC U {c''}
          For (f';f' ϵ CFM(c''))
             F ← F U {f'}
          Endfor
       Endfor
   Endif
Endwhile
While ((NC ≠ ∅)  && (F ≠ ∅))
   For(f'; f' ϵ F)
       If ATFM(AT,f')=(max(ATFM(AT,f_i)), ∀ f_i ∈ F)
          fe ← f'
          Break
       Endif
   Endfor
   F ← F – {fe}
   For (c'; c'ϵ NC)
       If (fe ϵ CFM(c'))
          CCs ← CCs U {c'}
       Endif
   Endfor
   For (c'; c'ϵ CCs)
       For (f''; f'' ϵ CFM(c'))
          If ATFM(AT,f'')=(max(ATFM(AT,f_i)), ∀ f_i ∈ F)
             co ← c'
             Break
       Endfor
   Endfor
   NC ← NC -{co}
   Q ← Question about fe
   If Answer(Q) = Yes
       UCAM ← UCAM U {co}
       For (f';f' ϵ CFM(co))
          F ← F - {f'}
       Endfor
   Else
       For (c'; c' ϵ NC)
          If |CrrCC(co,c')| > 0.22
             Cl ← 1
             Break
          Else
             Cl ← 0
          Endif
       Endfor
       If Cl == 0
          For (c'; c' ϵ NC)
             If  CrrCC(co,c') > 0.22
                NC ← NC -{ c' }
             Endif
          Endfor
       Endif
   Endif
Endwhile
```
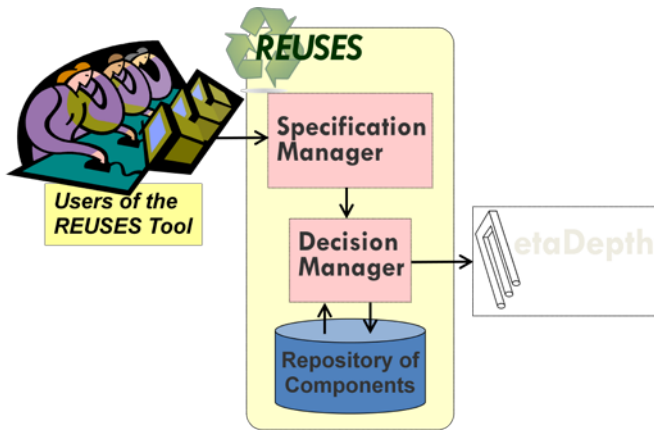
Figure 2. REUSES Modules and MetaDepth.

The Decision Manager facilitates the interaction of the users with the Specification Manager. The latter is in charge of obtaining the UCAMs, which are used by the Decision Manager to obtain the CCAM. The Decision Manager enables the consensus of the user community through a refinement process in order to obtain the FCCAM.

The repository contains components implemented by our team (e.g. a login component, a radar awareness component) as well as Facebook components and Google Gadgets, e.g. Google Cabinet, Google Charts, Google Docs, among others. All these components are connected and integrated using mashup technology.

Finally, the decision manager extracts from the repository the specific information about the components needed to model the FCCAM. Using this information, it generates a model of the collaborative application, which is fed into MetaDepth. The latter synthesizes code for the final collaborative application as shown next.

B. The MetaDepth Meta-Modelling Environment

MetaDepth [11] is a multi-level meta-modelling framework which integrates languages for model manipulation and code generation. We have used this tool to develop a mashup-like family of domain specific modelling languages (DSMLs) to describe different aspects of collaborative applications, such as its users and their roles, its component-based functionality, and the style and layout of the components in the application. We have also built a code generator that, given a set of models describing the collaborative system, produces the final application from a repository of predefined components, as explained in the previous section. The final application integrates heterogeneous technologies such as HTML, CSS,

JavaScript (to access third-party APIs from Google or Facebook) and PHP (for DataBase access).

However, as discussed in the introduction, the direct use of this family of DSMLs for building collaborative applications is difficult for non-technical users as the languages we have developed use concepts of the solution domain (components, pages, etc.) but not of the problem space (goals, activities, end-user functionality). Nonetheless, as the DSMLs are closer to the solution space, it is easier to generate code from them in an automatic way in order to speed up the development of applications.

As we mentioned at the end of Section II, in the third and final phase of the proposed approach REUSES invokes MetaDepth, which is in charge of synthesizing the final application. Figure 3 shows the tool chain in this phase. In the first step, REUSES is in charge to transform the high-level preferences of a user community, into the lower level DSMLs understood by MetaDepth, from which code is generated (step 2). The final application is automatically deployed in the server where REUSES and MetaDepth are running (step 3). Hence, REUSES is used to build the FCCAM based on the user decisions (problem domain), from the FCCAM we generate a lower-level model that realizes the FCCAM in a set of inter-related web components (solution domain), from which the code of the final application is generated.

V. CASE STUDY

Some courses of the new Grade in Computer Science of the Universidad Autónoma de Madrid are project-oriented and students need appropriate tools to support group-work. Currently, the collaborative activity is supported by Moodle [13] and various applications and services that exist on the Internet (e.g. Google Docs, Google Talk, among others). However, integration between these tools and the information they use is needed for specific job tasks. In this academic context, we propose using our REUSES to develop an application to support students' group-work.

Considering the Programming Project course, which is a first-year course in the second semester, the students must work in groups of 3-4 people to carry out a whole software development project (specification, design, coding and testing). The professors of this subject are the REUSES tool users, in charge of specifying and generating the collaborative application to support students' group-work throughout the project.
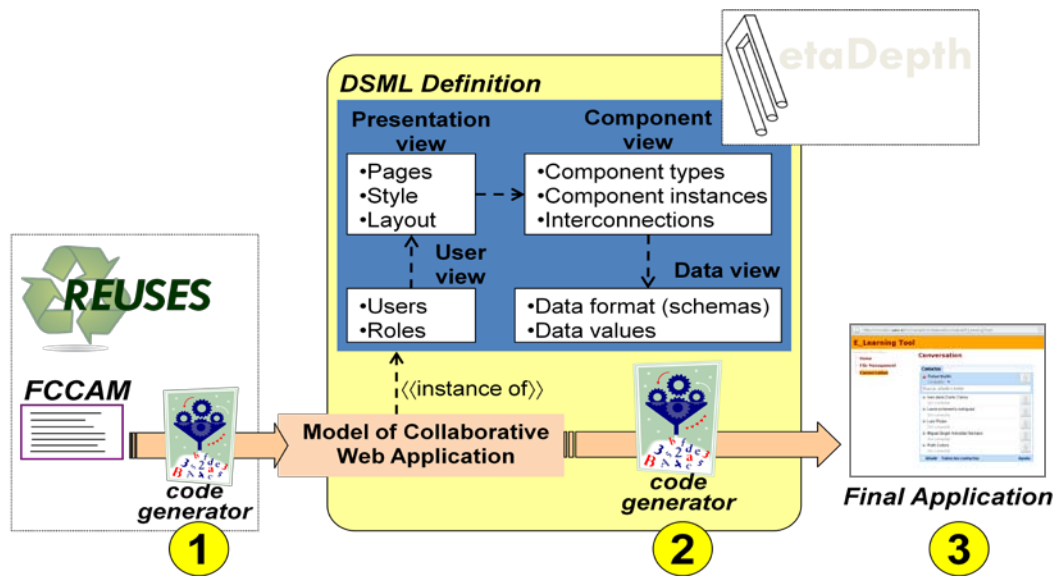
Figure 3. Code Generation from FCCAMs using MetaDepth

In this case, the REUSES tool supports the collaborative generation process of a Project Management collaborative application by a user community in the following stages:

1. REUSES provides to the user the first question of the questionnaire to start completing the UCAM. In this case, the user selects 'Project Management Application' (see Figure 4). This action assigns a value 1 to the UCAM dimensions that have more than 0.8 in the corresponding BM dimensions of Project Management application type.

2. REUSES shows the next question of the questionnaire (see Figure 5). In this step, the components that were selected taking into account the answer to the previous questions are shown with green color on the UCAM representation. Then, in the same way, REUSES shows the appropriated questions of the questionnaire step by step.

3. When all UCAMs are obtained by REUSES, the tool performs the matching process to generate a tentative CCAM (see Figure 6). In this case, the selected components for the collaborative web application are: Login, GoogleCabinet, GoogleCalendar, GoogleGroups and Gmail (they have maximum acceptance value, i.e. 1). On the other hand, the components with low acceptance value are: GoogleTable, GoogleChart and GoogleMaps (see that they have value of 0.5).

4. During the refinement process the components with low acceptance value are definitely not useful for the user community. Consequently, the collaborative application will be constructed with some of the others components.

5. REUSES collects the needed information about the selected components in the FCCAM from the repository of components and invokes MetaDepth (see Figure 7), which generates the final Web Collaborative Application (WCA). Its URL is provided to the professors.

6. Clicking on the link to the provided URL the WCA is accessed (see Figure 8).

Hence, overall REUSES enabled the generation of a customized collaborative web application, tailored for the particular problem. Moreover, the tool supported the collaborative design of the final application, resulting in an application with features and components agreed between the members of the user community.
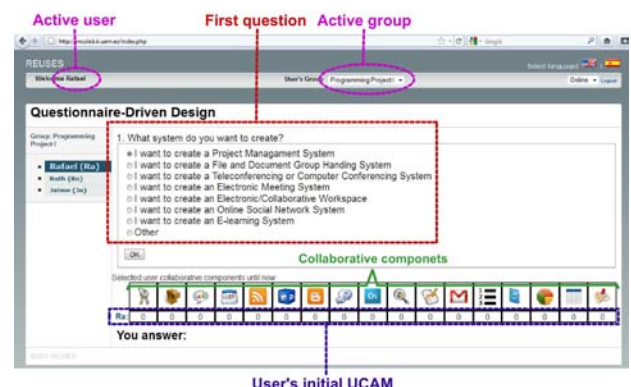

Figure 4. REUSES asks the first question of the questionnaire to the user
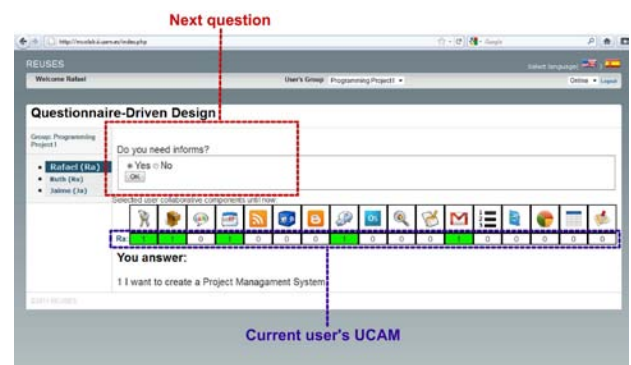

Figure 5. REUSES shows the next questions of the questionnaire to the user, and his/her current UCAM with the proposed components by the application
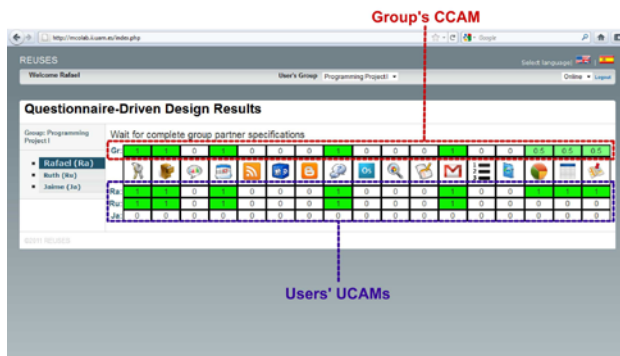
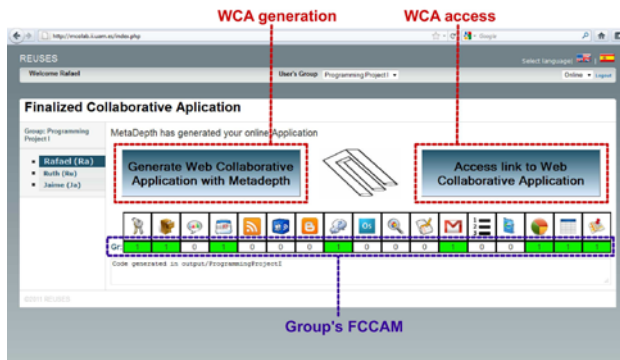Figure 6. REUSES shows the CCAM, and all the UCAMs of the community

Figure 7. REUSES shows the FCCAM and access to MetaDepth to generate the final Web Collaborative Application (WCA) and a link to this

Figure 8. The final Web collaborative application generated by REUSES

VI. RELATED WORK

A large amount of collaborative applications was reviewed in order to determine the collaborative components to be used, their applications and uses. During this review, we found several examples of applications that motivated our actual research work. Most collaborative applications have a specific purpose, although some of them are for general purpose. In this section, firstly we will see some details of specific purpose collaborative applications for these two areas: e-Learning and Knowledge Management –both areas are related with the use case exposed in the previous section–. Secondly, we will see some general-purpose collaborative applications.

Some of the studied e-Learning systems are: the COLLECE System [14], DOMOSIM-TPC [15] and Ontoolsearch [16]. The first one, the COLLECE System, facilitates the collaboration of a group of students (possibly distributed geographically) in learning a programming language. In this way, they can edit, compile and execute programming code in a collaborative way. In addition, the system provides a shared

workspace with a chat. The second one, DOMOSIM-TPC, supports the Learning of Domotical Design. This system facilitates to a group of students the design, discussion and simulation of models of building automation systems. As the previous system, this also provides a shared workspace with a chat. Finally, Ontoolsearch is an interactive system for the search of learning tools. This system can help teachers find the most appropriate collaborative application for the collaborative work of his/her students.

In the Knowledge Management area, among others, we found these systems: KnowCat [17], KM QUEST [18] and Sofia [19]. The first one, KnowCat ("Knowledge Catalyser"), aims at generating high quality educational materials as a result of users' interaction with the materials, in an asynchronous and distributed way, without the need of an editor to manage the task. The system is based on a mechanism called "Knowledge Crystallisation". The second system, KM QUEST, is a collaborative Internet-based simulation game for learning to solve knowledge management problems. Finally, Sofia is a framework for implementing Group Storytelling applications, which stimulate collective knowledge building.

As examples of general-purpose collaborative systems, we can cite Google Sites (http://sites.google.com/) Wookie [20][21] and XoMashup [22]. Google Sites facilitates the creation of Web Sites using templates. It is useful for end-users without knowledge about Web applications design. Although it allows inserting collaborative components such as chats in the generated Web applications, these are mostly for individuals but not for groups. The second system, Wookie, can handle multiple users, making it possible to implement forums, chats and other social applications through widgets. Finally, XoMashups facilitates the generation of Web applications where tools and external sources can be assembled, which can support collaborative work. It uses mashup technology. The use of this system is individual, and it is not possible to configure a Web collaborative application among a group of users.

Hence, our approach has the advantage that it enables the generation of different types of collaborative applications. As a difference with existing systems to build collaborative applications, the functionality of the final application is agreed through a consensus-building mechanism, based on questionnaires.

VII. CONCLUSION AND FUTURE WORK

The present research work started with an exhaustive review of different studies about classifications and taxonomies of collaborative systems/applications [1]-[4]. Based on them, and after our detailed study of 79 collaborative applications, we generated our own classification of collaborative applications and their collaborative features. Moreover, we built another classification where we took into account collaborative applications types and their collaborative features [12].

This previous study was used to propose our Questionnaire-driven Methodology for automatic generation of Web collaborative applications. The methodology first characterizes individual user preferences, and then builds a consensus using the preferences of the whole community. The methodology is supported by REUSES, a Web 2.0 tool, and MetaDepth, a meta-modelling and code generation tool. We rely on a repository of components to realize the final collaborative application.

We are currently organizing some experimental work to test our approach in the context of the new Grade in Computer Science of UAM. It is expected that the use of REUSES in the Programming Project course with their real instructors and students could give evidence about these following issues. Firstly, that this Web 2.0 tool is useful to the teachers in order to achieve consensus about the desired collaborative application and that the generated collaborative application would support the collaborative work of their students in the development of a software project. Secondly, that the students can work with the generated collaborative application properly in their collaborative tasks related with the specification, design, coding and testing of their project. Moreover, all this experimentation will contribute to improve the REUSES tool, in order to achieve that the final generated web applications will support a proper orchestration of their collaborative components.

For the next academic year, we will continue with the experimental work in other courses in order to obtain more feedback from teachers and students about the use and performance of REUSES and their generated collaborative applications. Moreover, we also plan to test REUSES in other user communities, which could benefit from their collaborative work, such as research groups at our institutions.

At the moment, we have taken into account six collaborative applications types. Our idea is to extend our approach with more types (e.g. collaborative e-learning and collaborative knowledge management) and to improve it with automatic learning using the new collaborative applications generated by REUSES as inputs (i.e. adding the characteristics of these new collaborative applications in the same way we configured our approach with the initial 79 applications studied). We are also enriching the repository with new alternative collaborative components of some existing ones at our repository, both for extending the supported collaborative features, as for providing other companies components (e.g. Facebook Widgets).

APPENDIX

TABLE III   CORRELATION MATRIX OF A SELECTION OF COLLABORATIVE COMPONENTS (CrrCC)

CC x CC*	Login	GoogleCabinet	GoogleTalk	GoogleCalendar	RSS	GoogleContacts Data	Blogger	GoogleGroups	Opensocial	GoogleSearch	GoogleDocs	Gmail	Google Poll	Historic	GoogleCharts	GoogleTable	GoogleMaps
Login	1.00	0.08	-0.03	-0.07	0.07	0.14	0.12	0.07	0.21	0.05	-0.16	0.15	-0.14	0.09	0.04	0.04	0.04
GoogleCabinet	0.08	1.00	-0.33	0.09	-0.03	-0.13	0.50	0.40	-0.36	0.60	-0.22	0.14	-0.11	0.15	0.42	0.42	0.42
GoogleTalk	-0.03	-0.33	1.00	0.04	-0.08	0.25	-0.28	-0.17	0.48	-0.20	0.61	0.30	0.24	0.04	-0.18	-0.18	-0.18
GoogleCalendar	-0.07	0.09	0.04	1.00	-0.34	0.05	-0.16	-0.22	0.01	-0.08	0.16	-0.06	0.36	0.16	0.21	0.21	0.21
RSS	0.07	-0.03	-0.08	-0.34	1.00	0.40	0.69	0.52	0.16	0.30	-0.45	0.45	-0.25	-0.16	-0.21	-0.21	-0.21
GoogleContacts Data	0.14	-0.13	0.25	0.05	0.40	1.00	0.20	0.09	0.68	0.07	-0.07	0.78	0.09	0.05	0.15	0.15	0.15
Blogger	0.12	0.50	-0.28	-0.16	0.69	0.20	1.00	0.74	-0.13	0.56	-0.48	0.39	-0.22	-0.04	0.21	0.21	0.21
GoogleGroups	0.07	0.40	-0.17	-0.22	0.52	0.09	0.74	1.00	-0.11	0.14	-0.26	0.44	-0.34	-0.11	0.08	0.08	0.08
Opensocial	0.21	-0.36	0.48	0.01	0.16	0.68	-0.13	-0.11	1.00	-0.14	0.24	0.44	0.24	0.30	-0.10	-0.10	-0.10
GoogleSearch	0.05	0.60	-0.20	-0.08	0.30	0.07	0.56	0.14	-0.14	1.00	-0.23	0.21	-0.01	0.09	0.34	0.34	0.34
GoogleDocs	-0.16	-0.22	0.61	0.16	-0.45	-0.07	-0.48	-0.26	0.24	-0.23	1.00	-0.01	0.38	0.12	-0.02	-0.02	-0.02
Gmail	0.15	0.14	0.30	-0.06	0.45	0.78	0.39	0.44	0.44	0.21	-0.01	1.00	-0.08	-0.01	0.20	0.20	0.20
Google Poll	-0.14	-0.11	0.24	0.36	-0.25	0.09	-0.22	-0.34	0.24	-0.01	0.38	-0.08	1.00	0.41	0.02	0.02	0.02
Historic	0.09	0.15	0.04	0.16	-0.16	0.05	-0.04	-0.11	0.30	0.09	0.12	-0.01	0.41	1.00	-0.05	-0.05	-0.05
GoogleCharts	0.04	0.42	-0.18	0.21	-0.21	0.15	0.21	0.08	-0.10	0.34	-0.02	0.20	0.02	-0.05	1.00	1.00	1.00
GoogleTable	0.04	0.42	-0.18	0.21	-0.21	0.15	0.21	0.08	-0.10	0.34	-0.02	0.20	0.02	-0.05	1.00	1.00	1.00
GoogleMaps	0.04	0.42	-0.18	0.21	-0.21	0.15	0.21	0.08	-0.10	0.34	-0.02	0.20	0.02	-0.05	1.00	1.00	1.00

*CC x CC, rows and columns are Collaborative Components.

REFERENCES

[1] G. Bafoutsou and G. Mentzas, "Review and functional classification of collaborative systems", *Int.J.Inf.Manage.*, vol. 22, no. 4, pp. 281-305, 2002.

[2] C.A. Bolstad and M.R. Endsley, "Tools for supporting team collaboration", *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 47, pp. 374-378(5), 2003. http://www.ingentaconnect.com/content/hfes/hfproc/2003/00000047/000 00003/art00026.

[3] D. Noble, D. Buck and J. Yeargain, "Metrics for Evaluation of Cognitive-Based Collaboration Tools," in *Proceedings of 6th International Command and Control Research and Technology Symposium*, Annapalis, USA., 2000.

[4] D. Mayrhofer, A. Back and R. Hubschmid, Web-Conferencing Software Tools: A Comprehensive Market Survey. Institut für Wirtschaftsinformatik, 2004.

[5] J. Grudin, "Groupware and social dynamics: Eight challenges for developers", *Commun ACM*, vol. 37, no. 1, pp. 92-105, jan 1994. http://doi.acm.org/10.1145/175222.175230

[6] L.A. Guerrero and D.A. Fuller, "A pattern system for the development of collaborative applications", *Information and Software Technology*, vol. 43, no. 7, pp. 457-467, 6/1, 2001.

[7] T. O'Reilly, "What is web 2.0: Design patterns and business models for the next generation of software", *Social Science Research Network Working Paper Series*, aug 2007. http://ssrn.com/abstract=1008839.

[8] V. Hoyer and K. Stanoevska-Slabeva, "Towards a reference model for grassroots enterprise mashup environments", in *Proceedings of the 17th European Conference on Information Systems*, Verona, Italy, 2009.

[9] V. Hoyer and M. Fischer, "Market overview of enterprise mashup tools", in *Proceedings of the 6th International Conference on Service-Oriented Computing/ (ICSOC '08)*, Springer-Verlag, Berlin, Heidelberg, pp.: 708-721, 2008. http://www.alexandria.unisg.ch/Publikationen/47516.

[10] V. Hoyer and K. Stanoevska-Slabeva, "The Changing Role of IT Departments in Enterprise Mashup Environments", in *Service-Oriented Computing – ICSOC 2008 Workshops,* G. Feuerlicht and W. Lamersdorf, Eds: Springer Berlin / Heidelberg, pp. 148-154, 2009.

[11] J. de Lara and E. Guerra, "Deep Meta-modelling with MetaDepth", in TOOLS'10. LNCS 6141, Springer, pp. 1-20. 2010.

[12] J. Moreno-Llorena, I. Claros, R. Martín, R. Cobos, J. de Lara and E. Guerra, "Towards a Functional Characterization of Collaborative Systems", in *Proceedings of the 8th International Conference on Cooperative Design, Visualization, and Engineering* (CDVE'11). LNCS 6874, Springer, pp. 182-185, 2011.

[13] M. Dougiamas and P. Taylor, "Moodle: Using learning communities to create an open source course management system", *World Conference on Educational Multimedia, Hypermedia and Telecommunications* 2003.

[14] C. Bravo, M.A. Redondo, M.F. Verdejo and M. Ortega, "A framework for process–solution analysis in collaborative learning environments", *International Journal of Human-Computer Studies*, vol. 66, no. 11, pp. 812-832, 11 2008.

[15] C. Bravo, M.A. Redondo, M. Ortega and J. Bravo, "Real time collaborative modeling and simulation, an approach for group learning of design domains", *Modeling and Simulation Magazine. Society for Modelling and Simulation Internacional*, vol. 2, no. 2, pp. 12-30 2003. http://portal.acm.org/citation.cfm?id=1289765.1289768.

[16] G. Vega-Gorgojo, M.L. Bote-Lorenzo, J.I. Asensio-Pérez, E. Gómez-Sánchez, Y.A. Dimitriadis and I.M. Jorrín-Abellán, "Semantic search of tools for collaborative learning with the ontoolsearch system", *Comput.Educ.*, vol. 54, no. 4, pp. 835-848, 5 2010.

[17] R. Cobos and M. Pifarré. "Collaborative knowledge construction in the web supported by the KnowCat system", *Computers & Education*, no. 50. pp. 962-978, 2008.

[18] H. Leemkuil, T. Jong de, R. Hoog de and N. Christoph, "KM QUEST: A collaborative internet-based simulation game", *Simulation & Gaming*, vol. 34, pp. 89-111 2003. http://purl.org/utwente/46564.

[19] C.M. Luz, M.R.S. Borges and M.L.M. Campos, "Sofia: a framework for the development of group storytelling tools", in *Proceeding of the 2nd ACM international workshop on Story representation, mechanism and context*, Vancouver, British Columbia, Canada, 2008, pp. 41-48.

[20] P. Sharples, D. Griffiths and S. Wilson, "Using widgets to provide portable services for IMS learning design", *Educational Cybernetics: Conference Papers*, no. 5 2008.

[21] S. Wilson, P. Sharples, K. Popat and D. Griffiths, "Moodle wave: Reinventing the VLE using widget technologies", in *Proceedings of 2nd Workshop Mash-Up Personal Learning Envrionments (MUPPLE'09), workshop in conjunction with 4th European Conference on Technology Enhanced Learning (EC-TEL 2009): Synergy of Disciplines*, Nice, France, pp. 47-58, 2009.

[22] F. Moedritscher, G. Neumann, V.M. Garcia-Barrios and F. Wild, "A web application mashup approach for eLearning", *OpenACS and LRN Conference 2008: International Conference and Workshops on Community-Based Environments*, pp. 105-110. 12-16 Feb 2008.