Collaborative Problem Solving in Virtual Communities of Practice – A Case Study in Disaster Prevention and Handling

María Clara Casalini(1) **Elsa Estevez**(1,2) **Tomasz Janowski**(2)

(1)Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur
Bahía Blanca, 8000, Buenos Aires, Argentina

(2)United Nations University – International Institute for Software Technology
P.O. Box 3058, Macau
mcca@cs.uns.edu.ar, ece@cs.uns.edu.ar, tj@iist.unu.edu

Abstract

The paper presents a case study illustrating a collaborative problem solving process in Virtual Communities of Practice. The process relies on the existence of an underlying shared knowledge base – a repository of resources, and is carried out collaboratively by members of the community in six defined steps. The case study presents a Virtual Community of Practice dedicated to Disaster Prevention and Handling and explains the process for solving a problem posted by a member. The process is applied step-by-step showing its applicability to solve a real case and how the process enables enriching the repository with the new knowledge generated. The main contribution of this work is to show how the problem-solving process can be systematically applied to real problems within communities of any domain.

Keywords: Communities of Practice, Collaborative Problem Solving, Knowledge Repository.

1 INTRODUCTION

Communities of Practice (CoPs) are groups of people with a shared interest in a particular area of knowledge that interact regularly sharing experiences and taking part in joint activities [15] with the objective of learning from each other and developing the area. Virtual Communities of Practice (VCPs) are communities supported at least partially by technology [11]. Many of these communities are based on on-line portals in which all or most of the interactions among the members take place. VCPs have a positive impact in organizations as well as individuals, providing the shared context for people to interact, discuss and learn from each other; making existing knowledge explicit; generating new knowledge and ideas; introducing collaborative processes, and promoting professional development.

Communities of practice are characterised by three elements [15]:

1) *domain* – subject of interest that brings people together; the area of knowledge they share
2) *community* – defined by members interacting through different activities; these activities define and shape the community and the relationships that are built among the members
3) *practice* – repertoire of resources, such as experiences, stories, tools, ways of addressing recurring problems, etc. shared by the community members.

Among all the possible activities carried out by members of VCPs, one which provides a major benefit to its members is the problem-solving process, since it allows members to collaborate and share expertise to find solutions to problems in the domain. A problem is defined as an intricate unsettled question. Problem-solving deals with the processes involved in finding solutions to problems. Particularly, in VCPs, the advantage is that the process of problem solving is done collaboratively, meaning that the process is done by peers, performing the same actions, having a common goal and working together [3].

In order to support all the activities carried out by members in VCPs, ICT (Information and Communication Technologies)-based solutions must be provided. The scope of these solutions depends on the type of activities supported, and may include: member registration, chat rooms, e-mail forums, browsing and updating the repository of resources, etc. Although for several of these activities there are well-known solutions provided, is not the case for the problem-solving process. One reason is the shortage of methods to carry out creative activities through computer-supported processes.

In [1], we present a process-oriented model for cooperative problem solving in Virtual Communities of Practice. The model describes a systematic process for building a solution for a given problem description. The model relies on a repository of various kinds of web resources – articles, case studies, problems, solutions, etc.; properties – data about or relations between resources; and statements. Statements are triples of a subject (resource), predicate and object (resource or data). Problems and solutions are all expressed as resources. Problem-solving is about connecting a problem description to the existing resources, creating new statements, properties and resources in the process. In particular, the problem and the new sub-problems are identified and solved through the six-phase process: i) problem registration, ii) problem exploration, iii) problem matching, iv) solution design, v) solution refinement, and vi) solution integration.

In this paper we present a case study illustrating the use of the two main components supporting the collaborative problem-solving process: the repository of resources, and the problem-solving process itself. The aim of developing a real case study is to show how the process enables enriching the repository with new knowledge – creating new resources, properties and statements; and how the solution is progressively built based on the knowledge stored in the repository, therefore demonstrating the application of the process. The case study refers to a Virtual Community of Practice about Disaster Prevention and Handling where a member posts a problem about specifying a plan for disaster handling for a flooding taking place in a city located in a sea area.

The rest of the paper is organized as follows. Section 2 introduces communities of practice; the structure of the repository and how knowledge is stored in it; and the problem solving process. Section 3 presents and develops the case study. First, the community is described; second the problem to be solved is presented; third, the problem solving process applied to the problem is developed step by step; fourth, the processes for solving the sub-problems are depicted; finally a summary of the case study is included. Section 4 formulates some conclusions.

2 COMMUNITY-BASED PROBLEM SOLVING

Virtual Communities of Practice (VCPs) enable distributed knowledge workers to share experience and seek solutions to concrete problems in a given field of interest. Two features distinguish the VCPs activities: i) interactions among members are supported by ICT, ii) activities are carried out collaboratively. The proposed model for the problem-solving process enables to automate the interactions among members and facilitates the collaborative work. Collaboration refers to the direct or indirect process of intertwined activities of two or more actors while solving the same problems [2]. Direct collaboration requires a commitment from the participants and occurs through direct communication between them. Indirect collaboration occurs when actors use shared data (results, information) that other actors made available. In the context of VCPs both styles of collaboration take place, the shared data involved in indirect collaboration is materialized in the proposed model by the repository of the community.

2.1 Community of Practice

The term Community of Practice has been formalized in [15] but the phenomenon is not new; communities exist in various forms and areas and are becoming more popular every day. While there are communities that span a particular organization, others include members of different organizations; some communities are formally recognized and sponsored, and some are informal communities initially not even visible to their own members. There is not yet a general consensus on how to classify communities of practice, however Porter in [11] gives a typology of virtual communities which suits VCPs. The proposed typology categorises communities at two different levels. At the first level two categories are recognised according to how the communities have been established: *member-initiated* - communities established by the members that remain managed by members; and *organization-sponsored* - communities sponsored by organizations, commercial or not. The second level of the typology categorises the communities based on the orientation of the relationships between members fostered by the community. Member-initiated communities show two types of relationships: *social* and *professional* relationships; while organisation-sponsored communities foster relationships among members as well as relationships between members and the sponsoring organisations.

The composition of members of VCPs is usually very heterogeneous, and although every member participates actively in the activities carried out by the community some of them are given particular responsibilities and tasks. Specific roles are defined in each community depending on the kind of activities but some common roles identified in general for all communities include:

1) *Members* - They populate the community, interact with each other sharing knowledge, information and experiences, raising questions, problems and concerns about common needs. Their responsibility is to participate in the community, learn and share.
2) *Facilitator* - This member is responsible for ensuring and facilitating communications among members, by clarifying communications, ensuring that every point of view is heard and understood, if necessary posting questions and keeping discussions on topic.
3) *Coordinator* - The role of the coordinator is to organize meetings, moderate discussions, and support community projects.
4) *Sponsor* - The sponsor engages with the community helping it to obtain resources and funding activities. The sponsor removes barriers for progress in the community.
5) *Experts* - Expertise in different areas of interest is needed in a CoP, some of the regular members of the community can provide this expertise. On particular occasions, experts are invited to community meetings.

The list of roles is expanded and customized for each particular community adding specific roles for the domain of the VCP and assigning the corresponding responsibilities.

2.2 Knowledge Repository

The collaborative problem-solving process proposed in [1] relies on a repository of resources. We model the repository in terms of three components: resources, properties and statements.

Resources are categorized into types. The types of resources managed by the community depend on its domain, for instance: papers, tools, documents, conferences, people, etc. The types of resources are organised as a hierarchy. Together with the hierarchy of resources the community may define a hierarchy of data types for defining values to some attributes of resources. As an example, the data type `date` can be defined to assign the value of the day a disaster happens to the resource type `Instance`. Resources are identified by Unique Resource Identifiers (URIs) [13].

Properties and statements are defined following the Resource Description Framework (RDF) [12] a language defined by the World Wide Web Consortium used to represent information and knowledge in the web. Properties express information about resources and establish relationships among them; they map a resource type to another resource type or to a simple data type. For instance, the property `country` can be defined to map the resource type `Instance` with the resource type `Country` associating the country where the disaster took place; the property `ocurred` for mapping the resource type `Instance` to the simple data type `date`.

Following RDF, statements are composed by three parts: subject, predicate and object. The subject identifies the resource described by the statement; the predicate identifies a property describing the subject; and the object identifies the value of the property - it can be a value of a data type or another resource depending on the definition of the property used in the statement. For instance, the statements: `[Bam,disaster,earthquake]`, `[Bam,located,Iran]`, `[Bam,occurred,26/12/03]`, `[Bam,measuredSystem,RichterScale]`, `[Bam,measureValue,6.5]` describe the earthquake occurred in Bam located in Iran on 26/12/03 having a magnitude of 6.5 on the Richter scale. Figure 1 depicts instances of these resources, properties and statements. It also depicts a flood that occurred on Grand Forks in United States on 18/04/97.

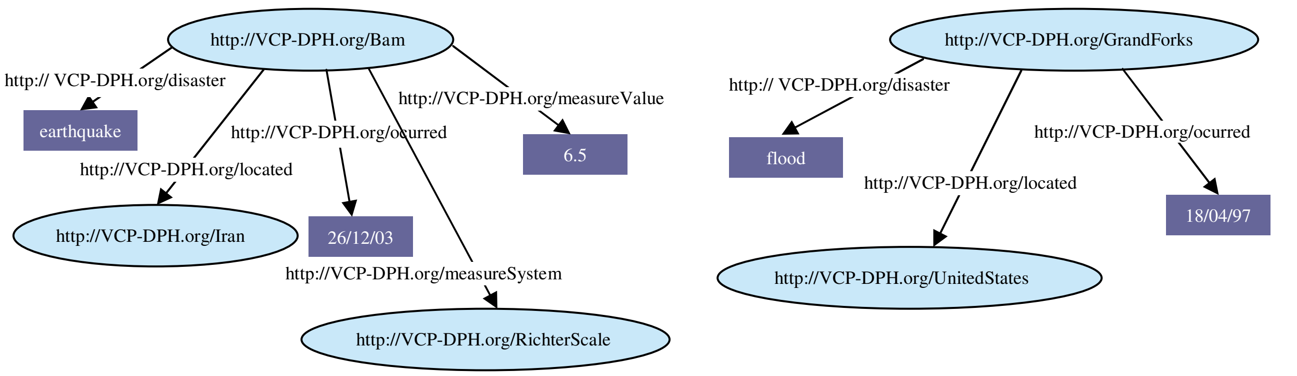

Figure 1: Resources, Properties and Statements

2.3 Problem Solving Process

The proposed problem solving process relies on the community repository using its contents and enriching it generating new resources, properties and statements as part of the process. The process is carried out collaboratively by members as a series of six steps:

1) *problem registration* - Registering a new problem by a member.
2) *problem exploration* - Analyzing and exploring the problem trying to find as much information as possible about the problem.
3) *problem matching* - Comparing the problem against existing problems in the repository to find similarities and dependencies.
4) *solution design* - Decomposing the problem into a set of sub-problems and analyzing dependencies between sub-problems.
5) solution refinement - Refining the solution.
6) solution integration - Linking the solution to the problem.

During the problem-solving process the knowledge generated is captured updating the repository. The updates carried out are shown in Table 1.

Phase	Repository Updates
Problem Registration	adding P – a resource of type Problem
Problem Exploration	adding statements with the problem as subject adding resources, properties and statements
Problem Matching	adding statements comparing the problem with others
Solution Design	adding sub-problems as resources of type Problem adding S – a resource of type Solution adding <S,isSolving,P> adding statements relating the solution with the sub-problems adding statements about relations between sub-problems
Solution Refinement	adding statements with the solution as subject adding statements relating the solutions to sub-problems adding resources, properties and statements
Solution Integration	removing <S,isSolving,P> adding <S,solved,P>

Table 1 : Repository Updates

The process starts with the *problem registration* phase when a member posts a new problem to the community adding the description of the problem as a new resource of type Problem. During *problem exploration* the problem is categorised and defined in more detail helping to understand it better. While *problem matching* the problem is compared with existing problems. The comparison of two problems is done by looking at the statements describing each of them and defining the degrees of similarities. The degrees of similarity are determined based on the set of properties and their values used in statements having the problems as subject. Two given problems may present the following similarities: the same set of properties with the same set of values; the same set of properties with different values; a subset or superset of the properties with the same values; a subset or superset of the properties with different values. If two problems are described with statements based on the same properties and with the same values for those properties; this may be an indication that the two problems are closely related to one another. *Solution design* consists in decomposing the problem into a set of sub-problems. Sub-problems are registered in the repository as resources of type Problem, therefore recursively initiating the problem solving process. During *solution refinement* all knowledge learnt about the solution is added as statements. In addition, statements specifying how to combine the solutions found for the sub-problems in order to create the solution to the original problem are also added. Finally, once all the sub-problems have been solved, and the constructed solution has been completely documented, *solution integration* enables linking the solution to the problem. Integrating the solution to the problem means that the problem has been solved and the process is concluded.

During the problem-solving process, the repository has been enriched with new resources - the problem, its' solution, and other resources related to them; properties and statements expressing the new knowledge created.

3 CASE STUDY IN COMMUNITY-BASED PROBLEM SOLVING

This section presents a case study illustrating the problem-solving process and the use of the repository. The case study refers to a Virtual Community of Practice about Disaster Prevention and Handling. In Section 3.1, the three components of this particular VCP are presented: the domain, the community and the practice. The structure of the repository is depicted and illustrated by some instances of resources, properties and statements. Section 3.2 describes a problem posted by a member. Section 3.3 describes the problem solving process for the posted problem. Section 3.4 explains the process for the defined sub-problems and Section 3.5 summarizes the case study.

3.1 Community Description

The case study considers a "Virtual Community of Practice on Disaster Prevention and Handling" (VCP-DPH) – A community of people interested in performing research and developing recommendations in the area of prevention and handling of disaster events. Members of this community are people from non-governmental organizations, civil defense, police, medical doctors, lifeguards, researchers and detectives, and everyone involved with the organization and coordination of actions to prevent and handle any kind of disasters, whether natural or man-made. The community grows on the basis of various activities, such us: developing and publishing best practices; organizing workshops, conferences and seminars; compiling data to document cases; carrying out online discussions; and elaborating statistics. Previously, without computer support for collaborative problem solving, the disaster prevention and handling activities carried out by the community were mostly based on isolated dissemination of knowledge and experiences, without systematic reuse of the generated knowledge. Since knowledge and experience were not formally captured most often were not properly shared. Advising activities depended on individuals - the knowledge was shared by consulting practitioners or experts, and reusing experiences was difficult.

VCP-DPH relies in a shared repository of resources. These resources have been gathered by members during the lifetime of the community, and produced through their interactions and the activities in which they participate. The repository is organized according to a hierarchy of resource types and simple data types defined by the community administrator. The root of the hierarchy is the type `All`. `Document`, `Person`, `Organization`, `Instance`, `Country`, `Problem` and `Solution` are descendents of the root. `Problem` and `Solution` are the two pre-defined types of resources used by the problem-solving process. The hierarchy may have several levels. For instance, `Member`, `Staff` and `Volunteer` are descendents of `Person`. The repository also includes the definition of data types. The hierarchy of resource types is shown in Figure 2.

All the resources added to the repository must belong to an existing type. Properties are defined with a name and a description composed by an ordered pair. The first element of the pair is the type of resource the property describes, and the second element may be a type of resource or a data type. For instance, the property `located` presented in Figure 1 is defined with its name and the pair `<Instance, Country>` since it maps an instance of a disaster and the country where it happened.

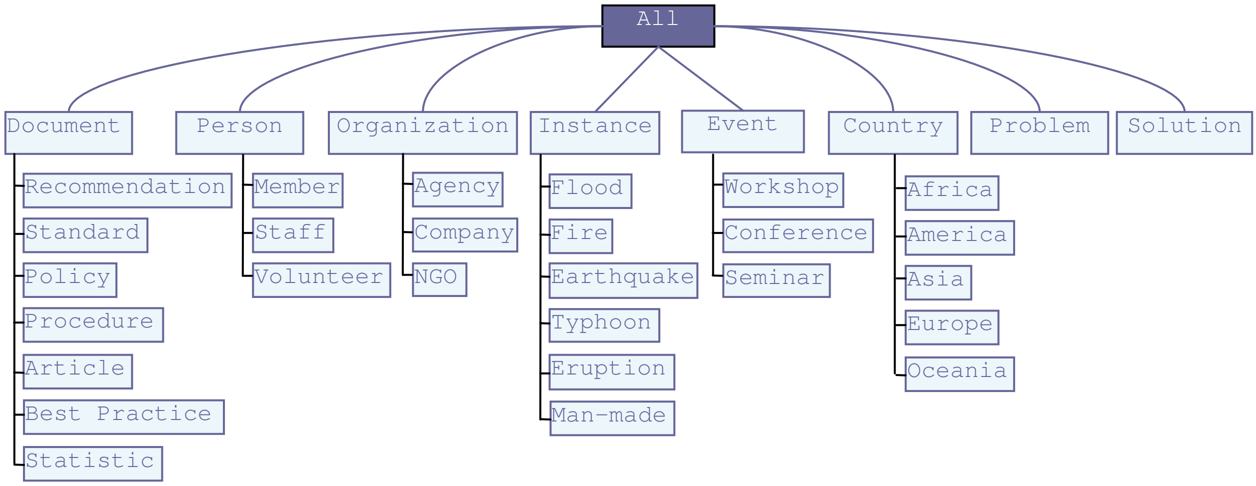

Figure 2: Repository Structure – Resource Types

Figure 3 shows some of the resources, properties and statements stored in the repository. References to instances, recommendations and standards were obtained from [4][5][9].

	Type	Instances
Resources	Flood	Flood1: 1928-Saint Francis Dam, Los Angeles Flood2: 1997-GrandForks, North Dakota
	Earthquake	EarthQuake1: 1995-Hyogo-Ken Nanbu, Kobe EarthQuake2: 2003-Bam, Kerman
	Country	Country1: Japan Country2: United States Country3: Iran
	Recommendation	Recommendation1: What to do before a flood, FEMA Recommendation2: What to do during a flood, FEMA Recommendation3: What to do after a flood, FEMA
	Standard	Standard1: 2006-Standard Flood Hazard Determination Form
	Problem	ProblemX: specify list of equipment required in rescue operation ProblemY: define a general chain of commands for a rescue operation
	Solution	SolutionX: list of equipment EQ SolutionY: chain of commands CC
Properties	**Name**	**Description**
	about	<Problem, Topic>
	coversDisaster	<Problem, Disaster>
	coversArea	<Problem, Area>
	isSolving	<Solution, Problem>
	solved	<Solution, Problem>
	contains	<Solution, Problem>
Statements	**Definition**	
	[ProblemX, about, equipment] [ProblemX, coversDisaster, generalDisasters] [ProblemX, coversArea, anyArea] [ProblemY, about, humanResources] [ProblemY, coversDisaster, generalDisasters]	[SolutionX, solved, PX] [SolutionY, solved, PY]

Figure 3: Sample Content of the Repository

Members use the repository while searching solutions to problems. The solutions found and the knowledge acquired are incorporated to the repository allowing the community to grow and expand.

3.2 Problem Description

When a member of the community presents a new problem to the community the collaborative process starts. All the resources of the community and all the members can be involved in the search of a solution to the problem. Let us assume a member posts the following problem:

```
ProblemP: Specify a plan for disaster handling for the case of a flooding
taking place in a city located in a sea area.
```

The problem solving process starts with the registration of the problem done by the member. The actions performed during the six steps of the process are described below. The presented problem is only an example and is not meant to be a real guide on how to specify a plan for disaster handling. Parts of the example were written following the guidelines taken from [7].

3.3 Problem Solving

The process for solving problem `ProblemP` includes the following steps:

Step 1: Problem Registration
The problem is registered in the repository adding it as a new resource of type `Problem`. Now the repository has one more instance of resource and members can start working on the problem.

Step 2: Problem Exploration
During problem exploration members analyse the problem by determining the topics to which it is related, categorising it, adding the information gathered to the repository as statements. In particular for the posted problem, members determine that the problem is about three particular topics: equipment, human resources and rescue techniques; it focuses on one type of disaster: flooding; and it may take place in a specific area: seaside. The statements added are:

```
1) [ProblemP,about,equipment]
2) [ProblemP,about,humanResources]
3) [ProblemP,about,rescueTechniques]
4) [ProblemP,coversDisaster,flood]
5) [ProblemP,coversArea,sea]
```

Step 3: Problem Matching
The result of comparing `ProblemP` against all the problems in the repository shows that it has the same properties as `ProblemX` and a superset of the properties of `ProblemY`. Two statements are added to the repository to reflect this fact:

```
1) [ProblemP,sameProps,ProblemX]
2) [ProblemP,supProps,ProblemY]
```

Step 4: Solution Design
The design of the solution starts by adding a new resource of type `Solution` to the repository; this new resource is called `SolutionS`. The statement `[SolutionS,isSolving,ProblemP]` is added to the repository linking the solution to the problem.
Next, `ProblemP` is decomposed into the following set of sub-problems:

```
1) ProblemP1: Define the chain of commands and the tasks for each position in
   the chain for a rescue operation in the case of a flooding
2) ProblemP2: Design communications between team members
3) ProblemP3: Specify a list of equipment required
4) ProblemP4: List rescue operations
```

The four problems are registered in the repository initiating the problem solving process for each of them. Four statements are added relating `SolutionS` to the sub-problems:

1) [SolutionS,contains,ProblemP1]
2) [SolutionS,contains,ProblemP2]
3) [SolutionS,contains,ProblemP3]
4) [SolutionS,contains,ProblemP4]

Since the communications between members of the team can only be designed after the chain of commands is defined, there is a dependency between `ProblemP1` and `ProblemP2` - the problem solving process for `ProblemP2` waits for the results of `ProblemP1` process. This fact is reflected in the repository by the statement `[ProblemP2,dependsOn,ProblemP1]`.

Step 5: Solution Refinement
Once all the problems identified on Step 4 have been solved, the solutions for all of them can be combined in order to conclude the construction of `SolutionS`. More information is added to the repository in this step in the form of statements, such as `[descriptionS,documents,SolutionS]`, where `descriptionS` is a resource of type `Document` containing a detailed explanation of how the solution is constructed. This document will explain that the chain of commands defined in the solution to `ProblemP1` must be set in place; that communications between members of the chain must follow the guidelines of the document produced in the solution to `ProblemP2`; and the equipments provided to the working teams are detailed in the list produced as outcome of the process applied to `ProblemP3`. Finally the document will describe how the rescue operations explained in the solution to `ProblemP4` should be applied in the event of a flooding.

Step 6: Solution Integration
The last step in the process, involves removing the statement `[SolutionS,isSolving,ProblemP]` added in Step 4, and adding the statement `[SolutionS,solved,ProblemP]`.

3.4 Sub-Problem Solving

While solving `ProblemP`, four sub-problems were identified. The processes for solving the sub-problems are explained in the following sections.

3.4.1 Sub-Problem 1

`ProblemP1` was added to the repository as a resource of type `Problem`. Its description is "`Define the chain of commands and the tasks for each position in the chain for a rescue operation in the case of a flooding`".

Step 1: Problem Registration
`ProblemP1` is registered in the repository adding it as a new resource of type `Problem`.

Step 2: Problem Exploration
The exploration of `ProblemP1` gives members the knowledge that the problem is about human resources for operations in the case of a flooding. The statements `[ProblemP1,about,humanResources]` and `[ProblemP1,coversDisaster,flood]` are added to the repository.

Step 3: Problem Matching
Matching the problem against all the problems in the repository shows that it shares the same properties as `ProblemPY`, so `[ProblemP1,sameProps,ProblemPY]` is added as a new statement.

Step 4: Solution Design
A new resource of type solution - SolutionS1, and the statement [SolutionS1,isSolving, ProblemP1] linking it with ProblemP1 are added.
The similarities between this problem and ProblemPY, and further analysis of the problem lead to the conclusion that this problem is simple enough to be solved directly using the solution of problem ProblemPY, so it is not further decomposed into sub-problems.

Step 5: Solution Refinement
A new document descriptionS1 describing the solution to this problem is written and added as a resource. The solution to this problem is added to the repository with the following statement [descriptionS1,documents,SolutionS1].
To construct this solution, the resource SolutionY – solution to ProblemPY – is reused. The chain of commands defined for a general disaster event is taken as a base design, necessary modifications are introduced to create the appropriate chain of commands for this particular case and define with precision the responsibilities and tasks assigned to each position in the chain. The resource descriptionS1 explains this in detail.

Step 6: Solution Integration
The statement [SolutionS1,isSolving,ProblemP1] is removed from the repository and the new statement [SolutionS1,solved,ProblemP1] is added.

3.4.2 Sub-Problem 2

ProblemP2 – "Design communications between team members" was added to the repository. A dependency between this problem and ProblemP1 was found during step 4 of the problem solving process for ProblemP. The dependency indicates that ProblemP2 could only be solved after some results from the problem solving process for ProblemP1 were obtained. In this case, since ProblemP2 consists in designing the communications between the members of the team designed in ProblemP1 the solution for the latter must be ready to start the process for the former.

Step 1: Problem Registration
ProblemP2 is registered in the repository adding it as a new resource of type Problem.

Step 2: Problem Exploration
The exploration of the problem gives members the knowledge that the problem is about designing communications, so the statement [ProblemP2,about,communication] is added.

Step 3: Problem Matching
The comparison of ProblemP2 against the problems in the repository does not find useful matchings, so no statements are added in this case.

Step 4: Solution Design
This problem is not decomposed in smaller sub-problems, in this case only the new resource SolutionS2 is added, as well as the statement [SolutionS2,isSolving,ProblemP2].

Step 5: Solution Refinement
The members in charge of solving this problem search for documentation about communication designs for teams of workers. In this process members may acquire books, articles, recordings, and all this resources will be added as new instances to the repository. Experts who are resources of the

community will be consulted, and based on their advices and the research done, members in charge of the solution building will design the pattern of communications for the work teams. The results will be included in the description of the solution for this problem which will be added to the repository as well as the following statement `[descriptionS2,documents,SolutionS2]`.

Step 6: Solution Integration

The statement `[SolutionS2,isSolving,ProblemP2]` is removed from the repository and the new statement `[SolutionS2,solved,ProblemP2]` is added.

3.4.3 Sub-Problem 3

The process for `ProblemP3` is similar to the one for `ProblemP1` - similarities between `ProblemP3` and `ProblemPX` are found; the problem is not further decomposed into sub-problems; and the solution for `ProblemP3` is constructed reusing the solution for `ProblemPY`.

Based on the list of equipment provided by `SolutionSY` and with the advice of the members of the community with expertise in flooding events and members with knowledge about geographical areas close to the sea, a new list is written. The list is added as a new resource to the repository and a document is written explaining all this in detail generating the resource `descriptionS3`.

3.4.4 Sub-Problem 4

The approach for solving `ProblemP4` is similar as the one for `ProblemP2` without reusing other existing solutions. For solving this problem, members search for documentation about techniques to apply in rescue operations, in particular for operations that need to take place in the indicated area and under a flooding event. In this process members may acquire books, articles, recordings, and all this resources will be added as new instances to the repository. A new list is written with the rescue operations required to solve this problem and included in the description of the solution.

3.5 Summary

The problem-solving process was developed step-by-step for the posted problem. During the solution design phase of this problem, four sub-problems were identified and created as resources of type `Problem`. Therefore, the process is recursively applied for solving sub-problems. Table 2 summarises the knowledge generated while solving sub-problems, depicting some of the resources and the statements added to the repository during the six phases of their corresponding processes.

	Sub-Problems			
1	P1: Define chain of commands and responsibilities for a rescue operation in case of flooding	P2: Design communications between team members	P3: Specify list of equipment required	P4: List rescue operations
2	[P1,about,humanResources] [P1,coversDisaster,flood]	[P2,about, communications]	[P3,about,equipment] [P3,coversDisaster,flood] [P3,coversArea,sea]	[P4,about, rescueTechniques]
3	[P1,sameProps,PY]		[P3,sameProps,PX]	
4	[S1,isSolving,P1]	[S2,isSolving,P2]	[S3,isSolving,P3]	[S4,isSolving,P4]
5	[descriptionS1,documents,S1]	[descriptionS2, documents,S2]	[descriptionS3,documents,S3]	[descriptionS4, documents,S4]
6	[S1,solved,P1]	[S2,solved,P2]	[S3,solved,P3]	[S4,solved,P4]

Table 2: Problem-Solving Process for Sub-Problems

4 CONCLUSIONS

The main objective of this paper was to present a case study for collaborative problem solving on a VCP referred to Disaster Prevention and Handling based on the process defined in [1].
The components of the community were presented, the repository of resources was described and a complete problem solving process was explained step-by-step. The aim of the case study was to show how the knowledge repository designed and the problem solving process proposed can be applied to a real case of any domain. Through its development it was demonstrated how the shared knowledge of the community grows, and how this knowledge is captured during the process increasing the resources, properties and statements of the repository. The main benefit of the approach is the definition of a systematic process to carry out problem-solving in any domain.
Computer support is provided to organize the process and maintain the repository, while manual intervention is required for building the solutions. Throughout the six steps of the process the addition and removal of information –resources, properties and statements- and the categorization of this information is easily automated providing support to the process. However some of the actions taken on the six steps are naturally carried out by humans. The comparison of one problem against the rest can be automated through different functions based on selected criteria, but the results obtained will be analysed by members to select the relevant and useful results.
Currently, an initial implementation of the repository is already in place in [13]. Future work includes the implementation of the problem solving process and exploring further automation of specific phases.

REFERENCES

[1] Casalini María Clara, Janowski Tomasz, Estevez Elsa. A Process Model for Collaborative Problem Solving in Virtual Communities of Practice. 7th Working Conference on Virtual Enterprises (PRO-VE'06), Springer, Helsinki, Finlandia. 2006.

[2] Conen Wolfram and Neumann Gustaf. Prerequisites for Collaborative Problem Solving. Proceedings of WETICE 96, IEEE 5th Intl. Workshops on Enabling Technologies, Stanford, CA, June 1996.

[3] Dillenbourg Pierre. What Do You Mean By "Collaborative Learning". Collaborative-Learning: Cognitive and Computational Approaches, pp 1-19. Oxford: Elsevier, 1999.

[4] Federal Emergency Management Agency (FEMA), http://www.fema.gov

[5] First American Flood Data Services, http://fafds.floodcert.com/news/index.asp?ID=103

[6] Fred Nickols. Community of practice. Overview. 2000.

[7] Interpol Disaster Handling Procedures.http://www.interpol.int/Public/DisasterVictim/guide/chapitre2.asp

[8] Merriam-Webster Dictionary.

[9] Natural Disasters, http://library.thinkquest.org/03oct/00758/en/home.html

[10] Pioneers of Change. Building communities of practice. A summary guide, July 2005.

[11] Porter Constance Elise. A Typology of Virtual Communities: A Multi-Disciplinary Foundation for Future Research. Journal of Computer-Mediated Communication 10(1), Article 3, November 2004.

[12] Resource Description Framework (RDF) – Semantic Web Activity – World Wide Web Consortium. http://www.w3.org/RDF/

[13] UNeGov.net – Community of Practice in Electronic Governance – http://www.unegov.net

[14] Uniform Resource Identifier (URI) – Architecture Domain – World Wide Web Consortium. http://www.w3.org/Addressing/

[15] Wenger Etienne. Communities of Practice. A Brief Introduction. 2004.