Using Shared Procedural Knowledge for Virtual Collaboration Support in Emergency Response

Gerhard Wickler and Austin Tate, *University of Edinburgh*

Jeffrey Hansberger, *US Army Research Laboratory*

A framework is described for developing and deploying procedural knowledge via a wiki in emergency situations where collaboration is needed. This approach helps reduce uncertainty during emergencies.

Emergency situations usually call for quick and appropriate action to minimize loss of life and property. However, knowing what these actions should be isn't always obvious, even if a current, post-disaster situation were known. One way to prepare for emergencies of a given type is to develop standard operating procedures (SOPs), with manuals containing procedural knowledge describing courses of action that should be followed in a given situation. These SOP manuals represent best-practice knowledge and are usually written by one or more experts with extensive experience in the field. Such procedural knowledge can be used to train emergency managers and others.

There's a significant amount of procedural knowledge for emergency response available today, mostly in physical form, and ranging in size from a few pages to several volumes. Although these manuals are considered valuable, there are several practical problems with such documents:

- *Access time*. Although these manuals are useful for teaching the procedures they contain, they're often not utilized during an actual emergency. This is simply because, during a real emergency, there's no time to search for information in large manuals. Emergency managers might have read through the SOPs, but under stressful conditions, they might forget options or omit steps.
- *Standard structure*. The manuals are often well structured in and of themselves, but there's no structuring standard in place. Thus, an emergency manager who must be familiar with different SOPs from different sources might find them confusing.

1541-1672/13/$31.00 © 2013 IEEE
Published by the IEEE Computer Society

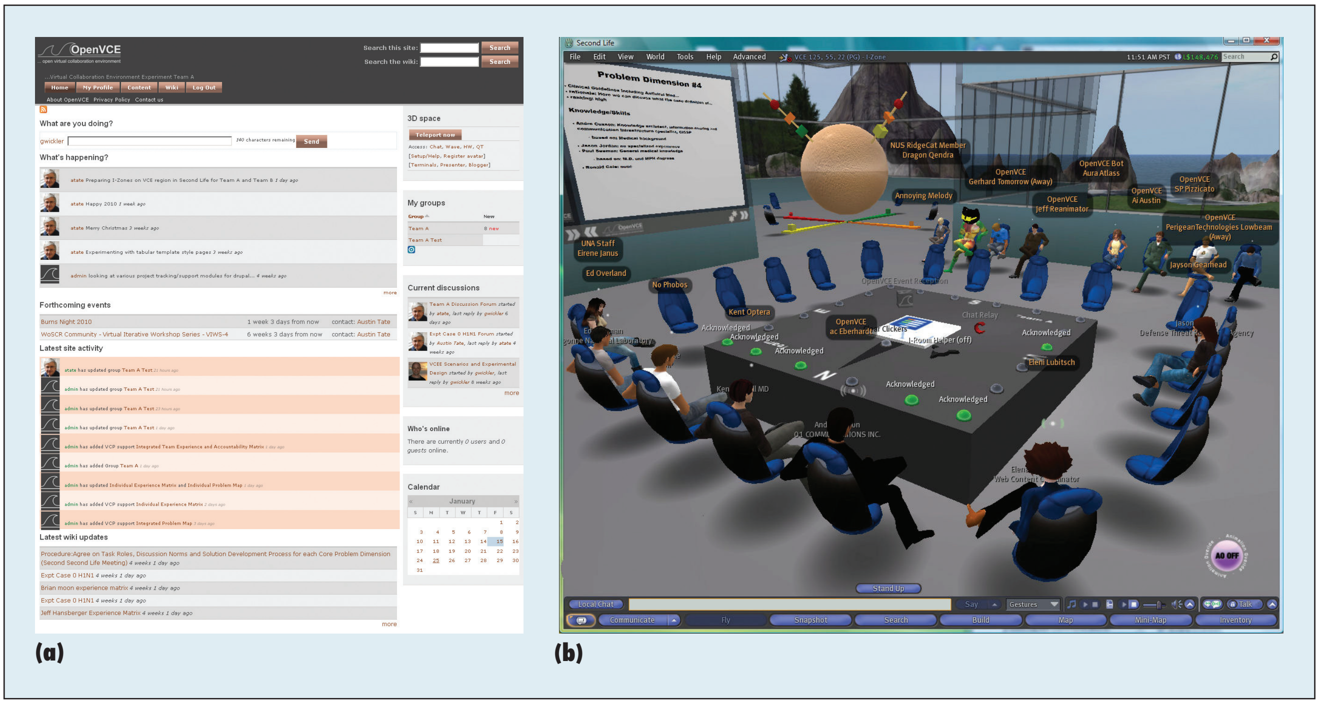

Figure 1. The Open Virtual Collaboration Environment (OpenVCE) space consists of two linked environments: (a) a dynamic website and (b) a 3D space for meetings. OpenVCE links aspects of a Web-based community portal—built on the widely available and established Drupal and MediaWiki open source software—with publicly accessible virtual world 3D spaces in "Second Life."

- *Updating.* Procedural knowledge should be updated with lessons learned after every emergency in which they have been applied. This is a cumbersome task to perform with printed manuals, and even Web-based documents offer limited support for this process.

The Open Virtual Collaboration Environment (OpenVCE) project[1] aims to facilitate collaborative work in a virtual space (see Figure 1). It consists of a dynamic website and a 3D space for meetings.[2] Organizations could use this environment to collaborate on the development of procedural knowledge, for example, or they could use it during an actual emergency to manage information and courses of action. In fact, this environment contains a specific piece of software that supports these two functions: the Issues, Nodes, Constraints, and Annotations (or <I-N-C-A>) extension for MediaWiki.

Technology can sometimes overwhelm novice users who are attempting to collaborate in this space. To avoid this, the project has developed the Virtual Collaboration Protocol (VCP), which is procedural knowledge that describes how this environment can be used to deal with certain types of emergencies. This protocol is also supported by an extension to the website that guides users who are following the protocol.

Collaborative Development of Procedural Knowledge

Dynamic Web technology supports the collaborative development of procedural knowledge in several ways. We've based our collaborative document editing facility, which can be used to write SOP manuals, on MediaWiki.[3] The reasons for this choice are simple: MediaWiki is open source (a project requirement) and scalable (it powers Wikipedia), and there's an active community behind it. However, wiki articles aren't structured to support procedural knowledge, which is why we've developed an extension that enables the structuring of an article according to the principles underlying Hierarchical Task Network (HTN) planning, which provides a "natural" way of decomposing tasks into subtasks, and, as such, is the structure found in many existing SOP manuals.

The CoScripter system also represents procedural knowledge in a wiki.[4] However, its representation isn't based on AI planning and thus doesn't support the automated composition of procedures. The Incidone system[5] uses Task-Oriented Programming (TOP) to represent and use procedural knowledge in emergency response, but the representation is closer to the specific programming language used.

Hierarchical Task Networks and <I-N-C-A>

What domain experts know as *SOPs* are called *methods* in HTN planning.[6] Methods formally describe how a task can be broken down into subtasks. A method's definition consists of four main parts:

- *Name*—the symbolic name of the method (there might be several methods for performing the same task).
- *Objective*—a formal expression describing the task pattern that the method can accomplish.
- *Constraints*—a set of constraints (such as on the current state of the world) that must hold for this method to be applicable.
- *Network*—a description of the subtasks into which this method "refines" or decomposes the given task.

We can use a method's name to refer to it, and to indicate the way in which we can accomplish the task. For example, we might define a method "set up camp for *?x* people," where *?x* is a variable that will get assigned an appropriate value when the method is applied. A method's objective is used for matching methods to tasks that must be accomplished, like a to-do list. For example, a required task might be to "provide shelter for 100 people," and the objective of the method could be "provide shelter for *?x* people."

AI has developed a set of algorithms for building, exploring, managing, and executing a set of HTN plans. The I-X framework is one such toolkit that includes an HTN planner. The representation used in the I-X framework is <I-N-C-A>.[7] In this framework, a refinement corresponds to a method. It consists of a set of issues to be addressed in a viable *plan*—a set of nodes that correspond roughly to the method's network, a set of constraints, and a set of annotations to hold information about the plan's other elements.

The <I-N-C-A> Extension for MediaWiki

The problem with HTN planning "domains" (the formal computer-processable expressions of SOPs) is that they're rather difficult to write. Domain experts usually aren't capable of producing these formal descriptions. Experts in AI planning, on the other hand, know the formalism, but don't have the knowledge that needs to be encoded. The approach taken with the first version of the SOP extension for MediaWiki was to keep the representation quite simple, at least initially, to encourage domain experts to encode their knowledge directly. Only a few tags existed to allow for a basic structuring of a set of methods. (These tags are implemented as MediaWiki parser functions.) The first one allows for the explicit specification of an objective:

```
{{#objective:...}}
```

We must use this tag if there are multiple methods that accomplish the same objective. If there's only one method (in the library), then the objective is taken to be the same as the name of the method, which is the title of the wiki article. There can be only one objective per method. The other tag provided by the extension allows the explicit specification of subtasks that the method must accomplish:

```
{{#subtask:...}}
```

There can be any number of subtask tags added to an SOP article. The order of the tags in the article is taken to be the order in which the subtasks are to be accomplished. Users must complete a subtask before they begin the next.

This simple SOP extension has several serious limitations that prohibit the representation of more complex procedural knowledge. For example, tasks and objectives can't be parameterized; methods are linear only, and no applicability constraints can be expressed.

To address these problems, we developed another extension for MediaWiki based on the <I-N-C-A> representation. Its major drawback, however, is that domain experts can no longer be expected to formally markup procedural knowledge as the tool is quite complex. The <I-N-C-A> extension uses an XML syntax instead of parser functions, introducing a number of new tags—including tags for issues, nodes (activities), constraints, and annotations. In addition, a set of refinements can be grouped into a domain to allow for a higher level structure of the procedures defined. The content of each of these new elements must be a formal description of the procedural knowledge in the syntax used by the I-X framework. Thus, domain experts can still use the wiki to develop an informal description of their procedural knowledge, but an AI planning expert familiar with I-X is required to mark up the informal descriptions with formal representations (see Figure 2).

When a marked-up procedure is saved, the system uses the formal representation to populate tables in the wiki's database corresponding to the different elements of the planning domain.

More recently, we identified a number of domain features that can be used to aid the knowledge-engineering process for procedural knowledge. The features can be efficiently and automatically extracted from the formal description and then compared to declarations found in the formal description. This can be used to validate the procedural knowledge and highlight potential problems.[8] However, we haven't yet integrated

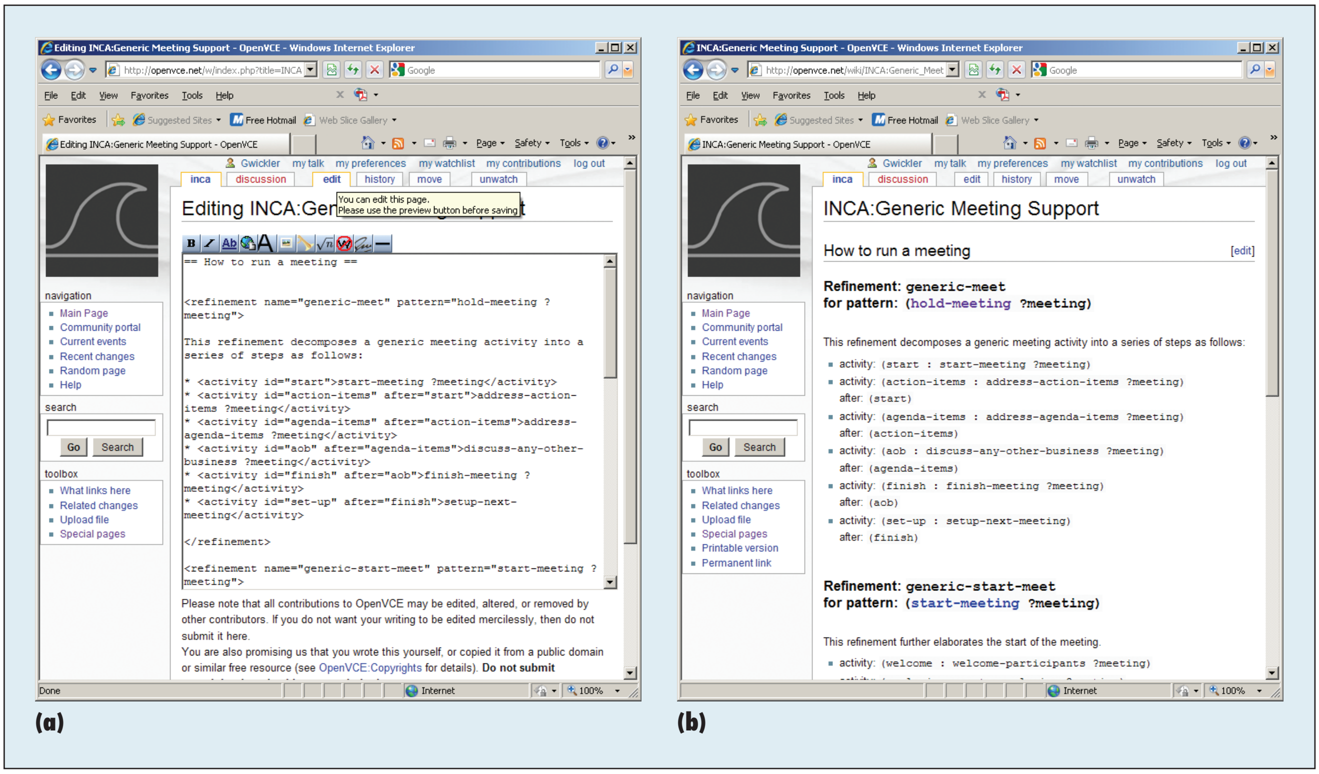

Figure 2. Procedural knowledge in (a) the edit view and (b) the normal view. When MediaWiki displays the marked-up procedural knowledge, the formal description is displayed in a style that makes it easier for a domain expert to assess and verify the description.

this work with the <I-N-C-A> extension available on OpenVCE.net.

Task Support Based on the <I-N-C-A> Extension

Although MediaWiki's concurrent editing facilities are already a useful tool for collaborative development of procedural knowledge descriptions, the more interesting aspect of this work is perhaps the task support provided when the knowledge is to be deployed.

Navigating Procedural Knowledge in the Wiki

The first level of task support is implemented within the wiki itself. We can create a "special page" to search the wiki for procedural knowledge relevant for a given task. Most of the work for this is performed when the procedural knowledge is edited and saved. This extracts all the relevant information and stores it in new tables in MediaWiki's database. The search for procedural knowledge is then translated into a database query that matches a given task to the objectives associated with each method. The result is formatted and presented to the user (see Figure 3).

Exporting Procedural Knowledge to I-X

I-X is a framework for writing applications that support distributed task-centered work. Its principal interface is a process panel, which we describe elsewhere.[9,10] Procedural knowledge written using the <I-N-C-A> extension can be directly imported into this tool, which effectively acts as an intelligent, distributed to-do list. This is implemented as another special page that extracts all the refinements that belong to a given domain from the database and transforms them into a syntax that can be read by I-X (see Figure 4). As a comparison of Figures 2 and 4 show, the syntax transformation is straightforward.

The content of the shown page isn't meant for human readers (other than for debugging purposes), but can be directly loaded into I-X. The HTN planner that is part of I-X can then be used to generate plans. That is, given a specific problem instance (a task), the planner can use the procedural knowledge (methods) to form a course of action that addresses the given problem, taking into account all the constraints defined in the refinements. This can be done automatically or using mixed-initiative planning, where the user defines the high-level strategy and the planner attempts to fill in the detail.

Task Support in the OpenVCE Environment

The third level of task support is part of the OpenVCE website (see Figure 5). The idea is again to provide

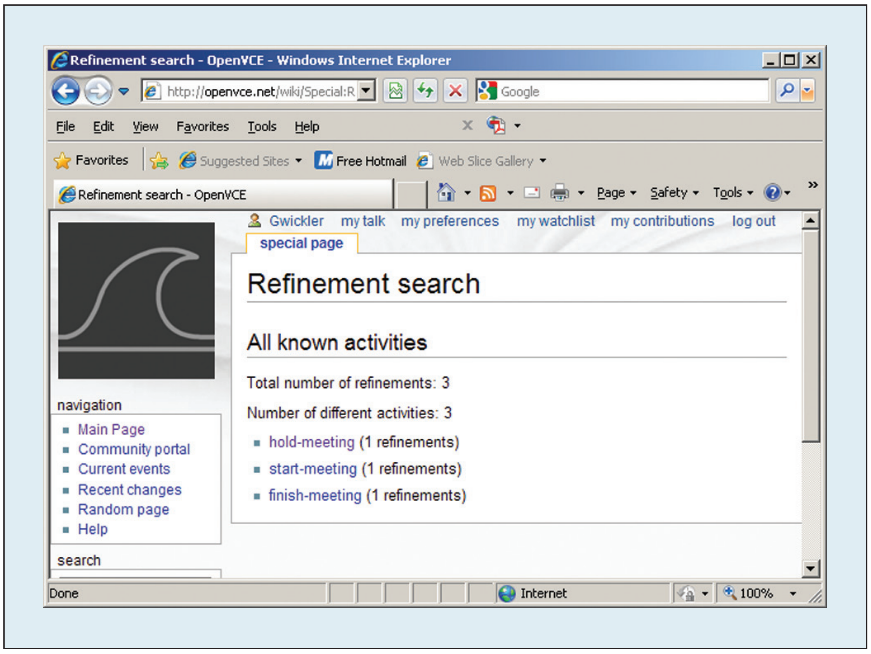

Figure 3. Refinement search in the Issues, Nodes, Constraints, and Annotations (or <I-N-C-A>) extension. Here, no task was provided, and in this case the page lists all known refinements—a total of three for three different tasks. Note that in this simple domain there's no overlap; there's exactly one refinement for each task. Users access the refinement by following the provided link.

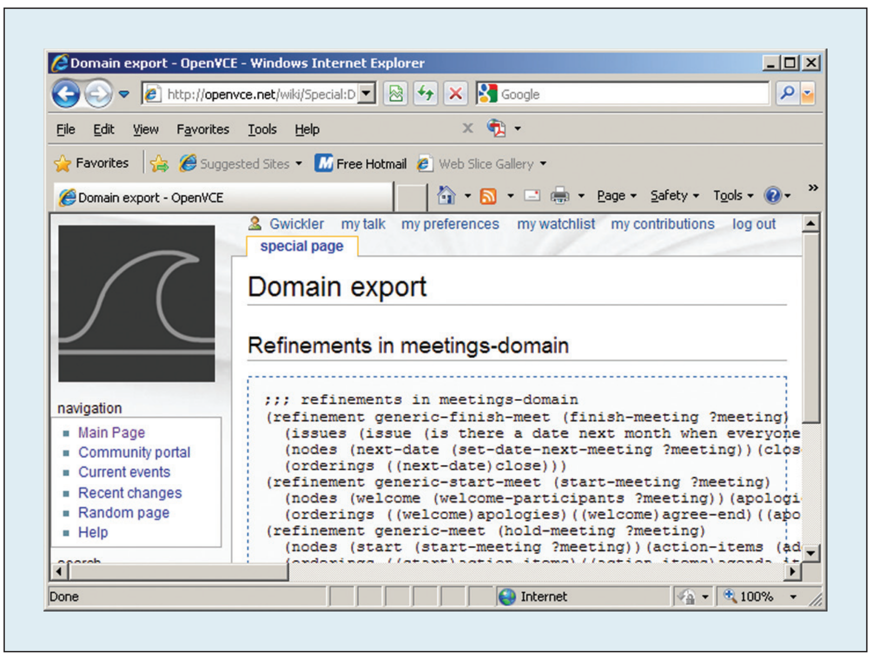

Figure 4. Domain export with the <I-N-C-A> extension. This "special page" transforms information from a database into a format that can be read by I-X. Compare this with Figure 2.

an intelligent, distributed to-do list, similar to the I-X Process Panel. However, the version integrated into OpenVCE does not include the HTN planner; the plan must be provided, for example, by the I-X system.

This plan can then be displayed as part of the OpenVCE website as a hierarchical set of tasks that must be accomplished.

The to-do list (Figure 5) is shown as a table with three columns. The first column is a summary of the tasks. This could include links to other webpages, including forms that must be completed to accomplish this task. The second column provides links to the SOP definition in the wiki so that users can always read the full text describing the current method being applied. The third column contains the check box for this to-do list item.

Each row represents a task to be accomplished. To visualize the hierarchical structure indentation, only the lowest-level tasks have an associated check box to indicate task completion. Higher-level tasks are completed when all their lower-level components are accomplished. Constraints are shown only implicitly in the list by highlighting those tasks that can now be tackled. In the example, the plan is at the beginning of execution; only the first subtask is highlighted as ready.

Case Study: The Virtual Collaboration Protocol

One of the outputs of the OpenVCE project is the VCP by Robert Cross.[11] The VCP is a reasonably generic procedure for collaborative problem solving, tailored to the resources available in OpenVCE—namely, the collaboration website and 3D meeting space. Because the VCP can be seen as an SOP for collaborating using OpenVCE technology, we chose to use it as a test case for the MediaWiki extension described here.

The Virtual Collaboration Protocol

The VCP provides guidance for collaborative problem solving. It consists of seven phases that correspond to the following main tasks:

- Before the first meeting: individuals define problem dimensions.
- First team meeting: team agrees on consolidated problem dimensions.
- Before second team meeting: individuals describe experiences with respect to problem dimensions.
- Second team meeting: individuals discuss experiences and subteams are assigned to address each dimension.
- Before third meeting: subteams formed in the previous step develop solutions for each dimension.
- Third meeting: individuals from the different subteams present and discuss solutions for each dimension.
- After third meeting: the team integrates solutions into a coherent solution document.

A key concept is the problem dimension, which describes an aspect of the problem that the team must address. Each of the phases is described by 1–2 pages of text explaining what must be accomplished by the team in that phase (the later phases tend to be more problem-specific and hence less elaborately described). It also contains a number of forms that provide templates for the outcomes of each phase.

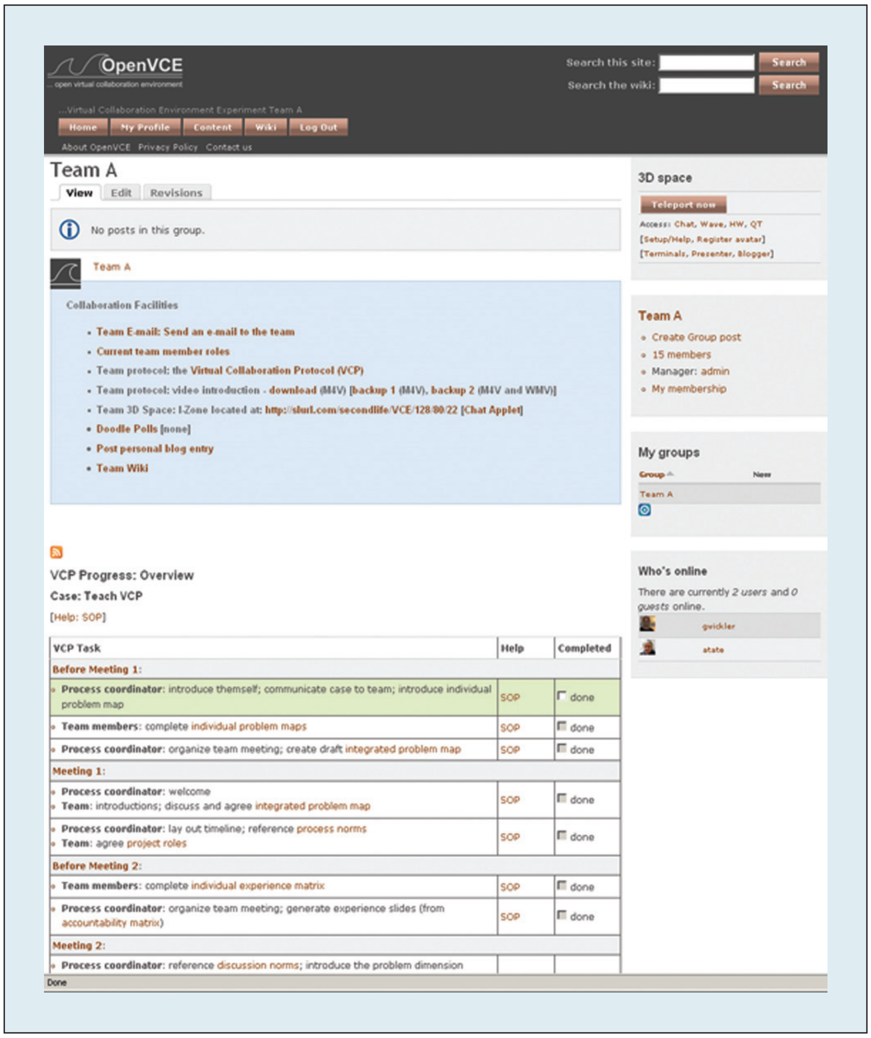

Figure 5. The OpenVCE website with a Virtual Collaboration Protocol (VCP) progress overview (basically, a to-do list) sets a plan into a hierarchical set of tasks.

Encoding the VCP in the Extension

The first step towards encoding the VCP using the MediaWiki extension was to import the text. We did this by creating seven articles on the wiki for each of the phases described in the original VCP document (which was written using Microsoft Word). In addition, we created another "overview" article from the table of contents. Each of the section titles was marked up as a subtask to enable the extension to find relevant methods (that is, those corresponding to the respective sections).

The next step was to go through the individual sections and identify subtasks therein that can be marked up as independent subtasks. In most pages, two to three subtasks could easily be identified and were marked up as such. Because there was no further advice provided by the document (meaning, no suggested methods) for performing many of these subtasks, some additional procedures were written that explained how the website and 3D space could be used to accomplish these tasks.

This resulted in a clear structure in which the top-level task—to collaborate to solve a problem—was broken down into its seven VCP phases, and each of these was broken down into finer steps that were more closely related to the technology of OpenVCE.

Experiments: Emergency Response Using OpenVCE

We conducted two experiments in 2010 to examine the impact that VCE had on crisis planning and collaboration when compared to traditional means of distributed collaboration among crisis response organizations and individuals. Here, we'll discuss

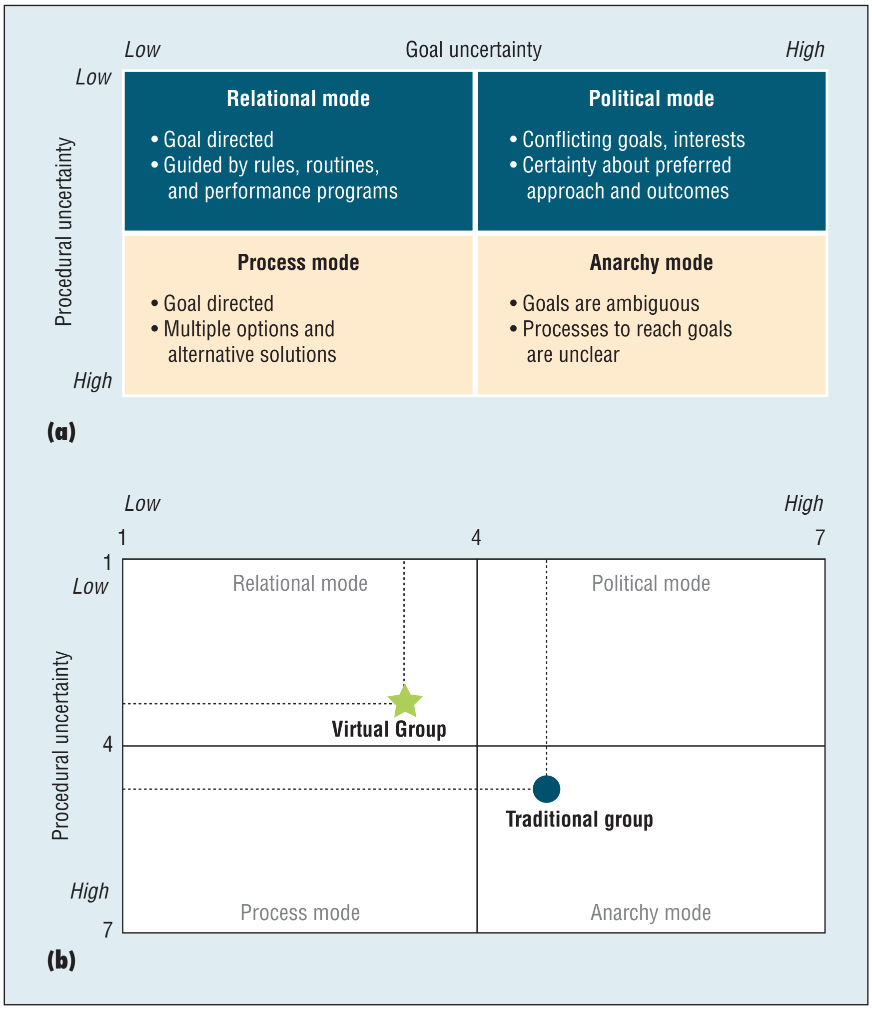

Figure 6. Goal and procedural uncertainty results. Placing (a) the results for goal and procedural uncertainty along the uncertainty dimensions presented (b) a clear picture of how much uncertainty was involved for each group.

the results and conclusions from the second and more comprehensive of the two experiments.

The VCE experiment introduced a biological agent outbreak scenario to two teams comprising crisis expert volunteers distributed across the US, UK, Canada, and Italy. The traditional group (control condition) used technology and means that would normally be used for distributed collaboration across various types of organizations (government, industry, non-government, military, and academia) during a crisis, including email for asynchronous collaboration and telephone and teleconferencing for synchronous collaboration. The virtual group (experimental condition) used the VCE's full capability. The traditional group consisted of seven participants and the virtual group had 10 participants (this difference is due to some subjects having to drop out of the experiment due to involvement in a real emergency); each group had what was considered equal expertise in crisis response and biological outbreaks; no members of either group had any prior experience of working together. We gave each group the same scenario and asked them to generate a crisis response plan over four days.

For the virtual group, this meant following the virtual collaboration protocol, which guided team interactions through a series of semi-structured steps to identify key crisis areas, a division of labor, roles for planning activities, and solutions to address the various elements of the crisis. The protocol guided and supported member activities for both asynchronous and synchronous interactions. The members of the traditional group, on the other hand, were free to organize their activities over the four days however they wished. As such, their success depended on their existing abilities to coordinate and collaborate. In the event, both groups presented plausible solution plans as the outcome of their collaborations. However, initial semantic analysis results of the final plans showed that the virtual group's final plan addressed more crisis-response topic areas and addressed them in greater depth compared to the traditional group's final plan.[12] To better understand these differences in the final plan, we examined some of the components of collaboration.

Among one of the measurements taken each experiment day was participant uncertainty, which we evaluated along two dimensions: goal and procedural. Goal uncertainty is the level of ambiguity a person has about the goals or objectives in their current situation or task. Procedural uncertainty, on the other hand, is how much ambiguity is associated with the steps or procedures necessary to accomplish the defined goals. Two seven-point Likert scale items measured each uncertainty dimension, which were averaged together. Chun Wei Choo has defined these uncertainty dimensions in terms of their interactions with each other.[13] The amount of goal and procedural uncertainty possessed by an individual and group will dictate the mode (see Figure 6a) of interactions and ultimately the success of the group.

The traditional group found themselves interacting in the "anarchy mode," where there was ambiguity about goals and procedures. Group and individual feedback after the experiment confirmed this finding. This group needed considerable effort to establish a common ground and understanding before they could engage in any planning efforts. This is also indicative of collaboration efforts among many different organizations, involving people with different backgrounds and expertise, particularly when they haven't worked together before.

The virtual group using the VCE and collaboration protocol fared much better and found themselves working within the "relational mode," where they clearly understood the goals and procedures. We statistically examined the overall difference between the two groups using repeated measures analysis of variance (ANOVA) and found a significant difference between the two groups ($F = 10.31$, $p < .01$). The virtual group had less goal and procedural uncertainty as they collaborated with their colleagues, which can result in increased efficiency and performance. These findings provide some evidence of the positive influence that can be gained using integrated technologies to support both asynchronous and synchronous collaboration over space and time.

Here, we described a framework for developing and deploying procedural knowledge in emergency situations where collaboration is needed. An extension to a wiki supports collaborative development of procedural knowledge by allowing the marking up of informal SOP knowledge with formal tags that can be processed by an AI planning framework, I-X. Enhanced browsing capabilities in the wiki itself support collaborative deployment of procedural knowledge in I-X, which can import the formal aspects of the marked up SOPs, and in OpenVCE, which supports the enactment of plans based on the procedural knowledge. The contribution here is a system that integrates collaborative development and deployment of procedural knowledge, resulting in enhanced knowledge engineering capabilities.

The experimental evaluation of the OpenVCE framework, including a procedure for virtual collaboration that was defined in the wiki extension, showed that procedural uncertainty can indeed be reduced. We assume that the explicit representation of procedural knowledge in OpenVCE and its availability to guide user actions in an emergency response setting were a major factor in reducing the procedural uncertainty, which in turn resulted in a more thorough response plan. The experiments described here support this conclusion.

A fundamental issue that remains to be addressed concerns the complexity of the underlying formal representations. A simple representation—for example, using just the two aforementioned tags—generally allows domain experts to annotate their own knowledge, whereas a more complex representation such as <I-N-C-A> requires dedicated knowledge engineers who aren't domain experts. Future work attempts to address this issue by adding validation support to the framework, hopefully allowing domain experts to manipulate the more complex representation, thus resulting in a more accurate representation of their knowledge. ■

Acknowledgments

Our work was sponsored by the US Joint Forces Command (USJFCOM)-Army Research Labs parent contract DAAD19-01-C-0065, subcontract no. SFP1196749DP (via Alion Science and Technology), task order no. 118; and the US Air Force Office of Scientific Research, Air Force Material Command, under grant number FA8655-09-1-3090. The University of Edinburgh and research sponsors are authorized to reproduce and distribute reprints and online copies for their purposes notwithstanding any copyright annotation hereon.

References

1. J. Hansberger et al., "Cognitively Engineering a Virtual Collaboration Environment for Crisis Response," *Proc. 22nd Ann. Conf. Computer Supported Cooperative Working*, ACM, 2010; www.aiai.ed.ac.uk/project/ix/documents/2010/2010-CSCW-Poster-Hansberger-Cog-Eng-VCE-for-Crisis-Response.pdf.

THE AUTHORS

Gerhard Wickler is a senior research fellow at the School of Informatics at the University of Edinburgh. His research interests include AI planning and intelligent agents. Wickler has a PhD in artificial intelligence from the University of Edinburgh. He's a member of the board of directors of the Association for Information Systems for Crisis Response and Management (ISCRAM). Contact him at g.wickler@ed.ac.uk.

Austin Tate is a professor of knowledge-based systems and director of the Artificial Intelligence Applications Institute (AIAI) at the University of Edinburgh. His research interests include emergency response using advanced knowledge and planning technologies, and collaborative systems, especially using virtual worlds. Tate has a PhD in machine intelligence from the University of Edinburgh. He's a Fellow of the Royal Academy of Engineering and AAAI, and an *IEEE Intelligent Systems* senior advisory board member. Contact him at a.tate@ed.ac.uk.

Jeffrey Hansberger is a research psychologist for the US Army Research Laboratory and the Human Research and Engineering Directorate. His research focuses on expertise, how it's acquired, how it's shared, and how it's enhanced by technology, design, and the environment. Hansberger has a PhD in human factors and applied cognition from George Mason University. Contact him at: jeff.hansberger@us.army.mil.

2. A. Tate, S. Potter, and J. Dalton, "I-Room: A Virtual Space for Emergency Response for the Multinational Planning Augmentation Team," *Proc. 5th Int'l Conf. Knowledge Systems for Coalition Operations*, 2009; www.aiai.ed.ac.uk/project/ix/documents/2009/2009-ksco-tate-iroom-eresponse.pdf.
3. D.J. Barrett, *MediaWiki*, O'Reilly, 2009.
4. G. Leshed et al., "CoScripter: Automating and Sharing How-To Knowledge in the Enterprise," *Proc. CHI Conf. Human Factors in Computing Systems*, ACM, 2008, pp. 1719–1728.
5. B. Lijnse, J. Jansen, and R. Plasmeijer, "Incidone: A Task-Oriented Incident Coordination Tool," *Proc. 9th Int'l Conf. Information Systems for Crisis Response and Management*, 2012; www.iscramlive.org/ISCRAM2012/proceedings/276.pdf.
6. M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*, Morgan Kaufman, 2004.
7. A. Tate, "<I-N-C-A>: An Ontology for Mixed-Initiative Synthesis Tasks," *Proc. Workshop Mixed-Initiative Intelligent Systems (MIIS), Int'l Joint Conf. Artificial Intelligence*, 2003, pp. 125–130; www.aiai.ed.ac.uk/project/ix/documents/2003/2003-ijcai-miis-tate-inca.pdf.
8. G. Wickler, "Using Planning Domain Features to Facilitate Knowledge Engineering," *Proc. Knowledge Engineering for Planning and Scheduling*, AAAI, 2011, pp. 39–46; http://icaps11.icaps-conference.org/proceedings/keps/keps2011-proceedings.pdf.
9. A. Tate, J. Dalton, and J. Stader, "I-P2- Intelligent Process Panels to Support Coalition Operations," *Proc. 2nd Int'l Conf. Knowledge Systems for Coalition Operations*, 2002; www.aiai.ed.ac.uk/project/ix/documents/2002/2002-ksco-ip2.pdf.
10. G. Wickler, S. Potter, and A. Tate, "Using I-X Process Panels as Intelligent To-Do Lists for Agent Coordination in Emergency Response," *Int'l J. Intelligent Control and Systems*, vol. 11, no. 4, 2006, pp. 248–259.
11. R. Cross and R. Thomas, *Driving Results through Social Networks*, Jossey-Bass, 2009.
12. J. Hansberger and R. Cross, "Understanding Distributed Collaboration within Virtual Worlds," *Proc. Sunbelt XXX Social Network Analysis Conf.*, Int'l Network for Social Network Analysis (INSNA), 2010; www.insna.org/PDF/Sunbelt/4_ProgramPDF.pdf.
13. C.W. Choo, *The Knowing Organization: How Organizations Use Information to Construct Meaning, Create Knowledge, and Make Decisions*, Oxford Univ. Press, 2005.