Using Shared Procedural Knowledge for Virtual Collaboration Support in Emergency Management

Citation for published version:
Wickler, G, Tate, A & Hansberger, J 2013, 'Using Shared Procedural Knowledge for Virtual Collaboration Support in Emergency Management', *IEEE Intelligent Systems*, vol. 28, no. 4, pp. 9-17. https://doi.org/10.1109/MIS.2013.31

Digital Object Identifier (DOI):
10.1109/MIS.2013.31

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
IEEE Intelligent Systems

Wickler, G., Tate, A., & Hansberger, J. (2013). Using Shared Procedural Knowledge for Virtual Collaboration Support in Emergency Response. IEEE Intelligent Systems, 28(4), 9-17. 10.1109/MIS.2013.31

Using Shared Procedural Knowledge for Virtual Collaboration Support in Emergency Management

Gerhard Wickler, Austin Tate
AIAI, University of Edinburgh, Scotland

Jeffrey Hansberger
US Army Research Laboratory

Abstract

This paper describes a framework that allows the collaborative development and deployment of procedural knowledge for task support in emergency situations. In this framework, procedural knowledge is represented in a wiki using an informal, textual description that is marked up with formal tags based on the <I-N-C-A> representation for hierarchical task networks used in AI planning. Procedural knowledge in the wiki can be used for task support by way of enhanced browsing facilities and the planning capabilities of an HTN planner. The latter supports the automatic composition of procedures to form plans for specific tasks. The tight integration of collaborative editing with deployment is new in this system and advances knowledge engineering for planning domain knowledge, that is, procedural knowledge. An experimental evaluation has shown that the explicit availability of procedural knowledge in emergency situations can reduce procedural uncertainty.

Keywords

Artificial Intelligence Planning, Procedural Knowledge, Knowledge Sharing, Virtual Collaboration

Introduction

Emergency situations usually call for quick and appropriate action to minimize loss of life and property. However, knowing what these actions should be is not always obvious, even if a current, post-disaster situation were known. One way to prepare for emergencies of a given type is to develop so-called Standard Operating Procedures (SOPs), 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, for example.

There is a significant amount of procedural knowledge for emergency response available today, mostly in the form of physical manuals and ranging in size from a few pages to several volumes. While these manuals are considered valuable where they exist, there are a number of problems with such documents in practice:

- Access time: While these manuals are useful for teaching the procedures they contain, they are usually not used during an actual emergency. This is simply because there is no time to search for information in large manuals. Emergency managers may have been through the SOPs, but under stressful conditions options may be forgotten or steps may be omitted.
- Structure: The manuals are often well-structured, but there is no standard way of structuring these documents. An emergency manager who needs to be familiar with different SOPs deriving from different sources may thus find them confusing to use.

- 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 OpenVCE project (Hansberger *et al*., 2010) aimed to develop an open virtual collaboration environment that facilitates collaborative work in a virtual space. This environment could, for example, be used to collaborate on the development of procedural knowledge, or it could be used 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 <I-N-C-A> extension for MediaWiki described in this paper.

The OpenVCE space consists of two linked environments: a dynamic website and 3D space for meetings (Tate *et al*., 2009) as shown in figure 1. This space links aspects of a web-based community portal built on widely available and established open-source software (Drupal and Mediawiki) with publically accessible virtual world 3D spaces in Second Life.

Figure 1. OpenVCE Website and 3D Virtual Meeting Space

To avoid all this technology overwhelming novice users who are attempting to collaborate in this space, the project has also developed the Virtual Collaboration Protocol, which is itself procedural knowledge that describes how this environment is meant to be used to deal with certain types of emergency. This protocol is itself supported by an extension to the website that guides users who are following the protocol.

Collaborative Development of Procedural Knowledge

The collaborative development of procedural knowledge can be supported in a number of ways using dynamic web technology. We have based our collaborative document editing facility that can be used to write SOP manuals on MediaWiki (Barrett, 2009). The reasons for this choice are simple: MediaWiki is open-source (a project requirement), scalable (it powers Wikipedia), and there is an active community behind it. However, wiki articles are not structured to support procedural knowledge, which is why we have developed an extension that allows for the structuring of an article according to the principles underlying Hierarchical Task Network (HTN) planning, which provides a ‘natural’ way of decomposing tasks into sub-tasks, and as such is the structure found in many existing SOP manuals.

A system that uses a similar approach, namely, representing procedural knowledge in a Wiki is CoScripter (Leshed *et al.*, 2008). However, their representation is not based on AI planning and thus does not support the automated composition of procedures. The Incidone system (Lijnse *et al.*, 2012) uses Task-Oriented Programming 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 are known as Standard Operating Procedures to domain experts are called methods in HTN planning (Ghallab *et al*., 2004, ch. 11). Methods formally describe how a task can be broken down into sub-tasks. The definition of a method consists of four main parts:

- Name: the symbolic name of the method (there may be several methods for performing the same task);
- Objective: a formal expression describing the task pattern that can be accomplished with this method;
- Constraints: a set of constraints (e.g. on the current state of the world) that must hold for this method to be applicable; and
- Network: a description of the sub-tasks into which this method 'refines' or decomposes the given task.

The name of the method can be used to refer to the method and it usually indicates the way in which the task is to be accomplished. For example, we may 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. The objective of a method is used for matching methods to tasks that must be accomplished, e.g. in a to-do list. For example, a task to be accomplished might be to "provide shelter for 100 people" and the objective of the above method could be "provide shelter for ?x people".

Artificial Intelligence 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 called <I-N-C-A> (Tate, 2003), which stands for *Issues*, *Nodes*, *Constraints* and *Annotations*. In <I-N-C-A>, a *refinement* corresponds to a method and it consists of a set of issues to be addressed in a viable plan, a set of nodes which correspond roughly to the network of a method, a set of constraints, and a set of annotations to hold information about the other elements of a plan.

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

The problem with HTN planning 'domains', the formal computer-processable expressions of SOPs, is that these are rather difficult to write. Domain experts are usually not capable of producing these formal descriptions. Experts in AI planning on the other hand know the formalism, but do not 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 very simple, at least initially, to encourage domain experts to encode their knowledge directly. Only a small number of 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:…}}
```

This must be used if there are multiple methods that accomplish the same objective. If there is 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 only be one objective per method.

The other tag that is provided by the extension allows the explicit specification of subtasks that need to be accomplished by the method:

```
{{#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, i.e. the subtasks are totally ordered: a subtask must be completed before the next is begun.
This simple SOP extension had a number of serious limitations that prohibit the representation of more complex procedural knowledge. For example, tasks and objectives cannot be parameterized, methods are linear only, and no applicability constraints can be expressed.
To address these problems, another extension for MediaWiki based on the <I-N-C-A> representation was developed. The major drawback here is, of course, that domain experts can no longer be expected to formally mark up procedural knowledge with this complex tool. The <I-N-C-A> extension is no longer based on parser functions but uses an XML syntax, 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. 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, as shown in figure 2.

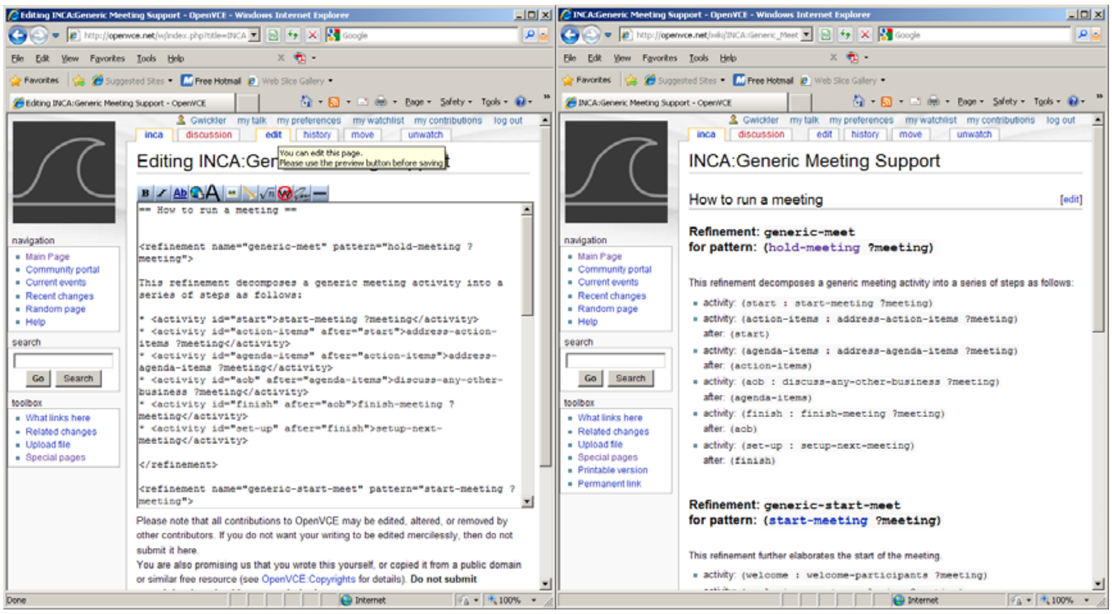

Figure 2. Procedural Knowledge in the edit view (left) and normal view (right)

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 have 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 (Wickler, 2011). However, this work has not yet been integrated with the <I-N-C-A> extension available on OpenVCE.net.

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

While the concurrent editing facilities provided by MediaWiki 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. A so-called Special Page can be used 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 which matches a given task to the objectives associated with each method. The result is formatted and presented to the user as shown in figure 3.

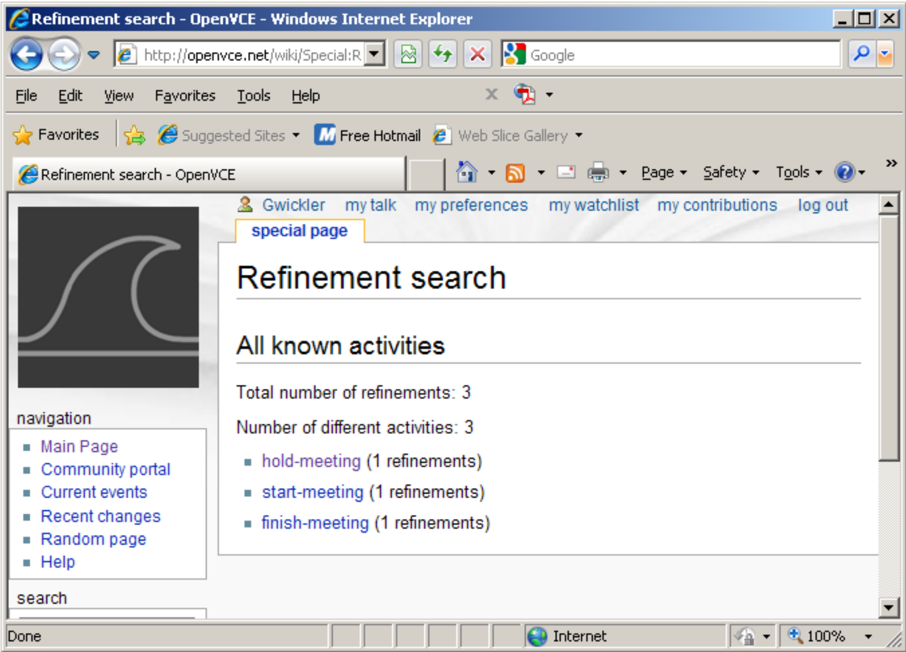

Figure 3. Refinement search in the <I-N-C-A> Extension

In the example shown, 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 is no overlap, i.e. there is exactly one refinement for each task. Users access the refinement by following the link that is provided.

Exporting Procedural Knowledge to I-X

I-X is a framework for writing applications that support distributed task-centered work. Its principal interface is the process panel described in (Tate *et al*. 2002; Wickler *et al*. 2006). Procedural knowledge written using the <I-N-C-A> extension described above 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. The syntax transformation is straight-forward (compare fig. 2 and 4). The result is shown in figure 4.

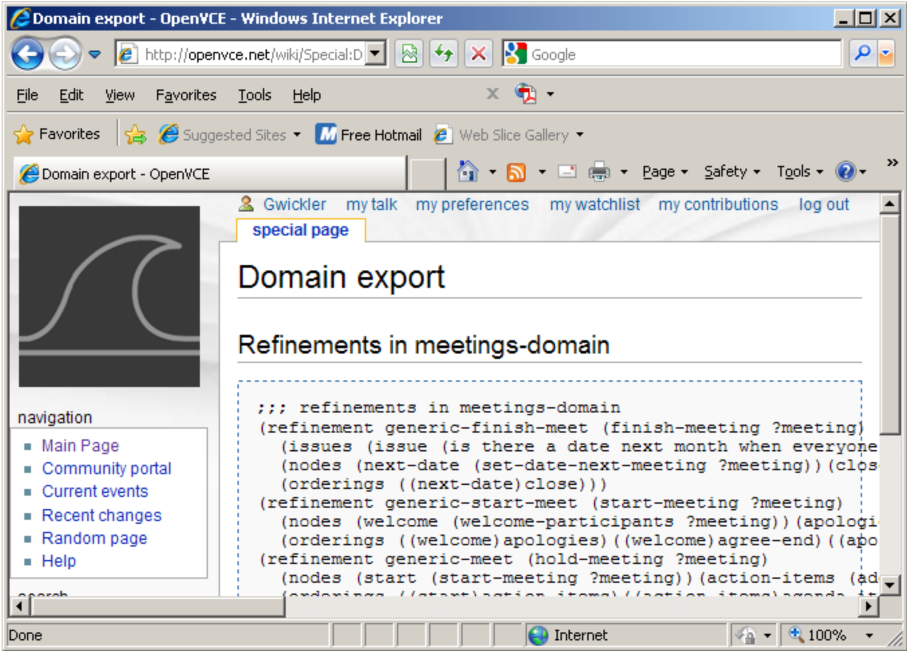

Figure 4. Domain export with the <I-N-C-A> Extension

The content of the shown page is not 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 using mixed initiative planning, where the user defines the high-level strategy and the planner attempts to fill in the detail, or fully automatically.

Task Support in the OpenVCE Environment

The third level of task support is part of the OpenVCE website and shown in figure 5. The idea is again to provide 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 has to be provided, e.g. by the I-X system. This plan can then be displayed as part of the OpenVCE website as a hierarchical set of tasks that need to be accomplished.

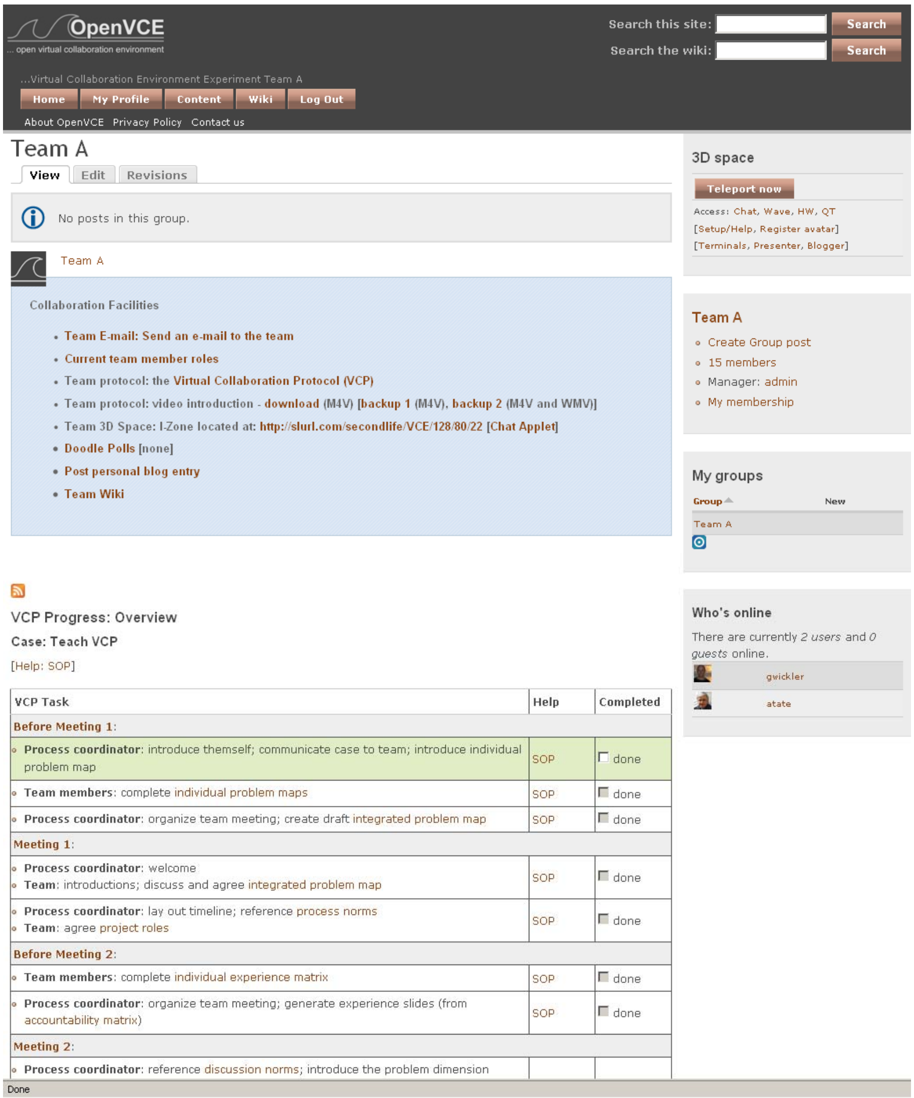

Figure 5. OpenVCE Website with VCP Progress Overview (To-Do List)

An example of such a hierarchical task list is shown in figure 5. The to-do list is shown as a table with three columns. The first column is a summary of the task. This may include links to other web pages including forms that need to be completed for this task to be accomplished. 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 tick-box for this to-do list item.

Each row represents a task to be accomplished. Indentation is used to visualize the hierarchical structure. Only lowest-level tasks have an associated tick-box to indicate completion of the task. 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 with only the very first sub-task being highlighted as ready.

Case Study: The Virtual Collaboration Protocol

One of the outputs of the OpenVCE project is the Virtual Collaboration Protocol (VCP) by Robert Cross (Cross and Thomas, 2009). 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. Since the VCP can be seen as an SOP for collaborating using OpenVCE technology, we have chosen to use it as a test case for the MediaWiki extension described above.

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 consolidated problem dimensions
- before second team meeting: individuals describe experience with respect to problem dimensions
- second team meeting: discuss experience and assign subteams to address each dimension
- before third meeting: develop solutions for each dimension
- third meeting: present and discuss solutions for each dimension
- after third meeting: integrate solutions into coherent solution document

A key concept is the *problem dimension*, which describes an aspect of the problem that needs to be addressed by the team. Each of the phases is described by 1-2 pages of text explaining what needs to 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.

Encoding the VCP in the Extension

The first step towards encoding the VCP using the MediaWiki extension was to import the text. This was done by creating seven articles on the wiki for each of the phases described in the original VCP document (which was written using MS Word). In addition, another 'overview' article was created from the table of contents. Each of the section titles was marked up as a subtask to enable the extension to find relevant methods, namely 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. Since for performing a number of these subtasks there was no further advice provided by the document (= no suggested methods), 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 where 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 Management using OpenVCE

Two experiments were conducted in 2010 to examine the impact the VCE had on crisis planning and collaboration when compared to traditional means of distributed collaboration among crisis response organizations and individuals. Results and conclusions from the second and more comprehensive of the two experiments will be discussed further.

The VCE experiment introduced a biological agent outbreak scenario to two teams comprising crisis expert volunteers distributed across the U.S., U.K., Canada and Italy. The traditional group (control condition) used technology and means that would normally be used for distributed collaboration across these types of organizations (government, industry, non-government, military, and academia) during a crisis, including email for asynchronous and telephone and teleconferencing for synchronous collaboration. The virtual group (experimental condition) used the full capability of the VCE. The traditional group consisted of 7 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; and no members of either group had any prior experience of working together. Each group was given the same scenario and asked 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, division of labor and roles for planning activities, and identify 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 in whatever way they thought fit. As such, their success would depend on their existing abilities to coordinate and collaborate.

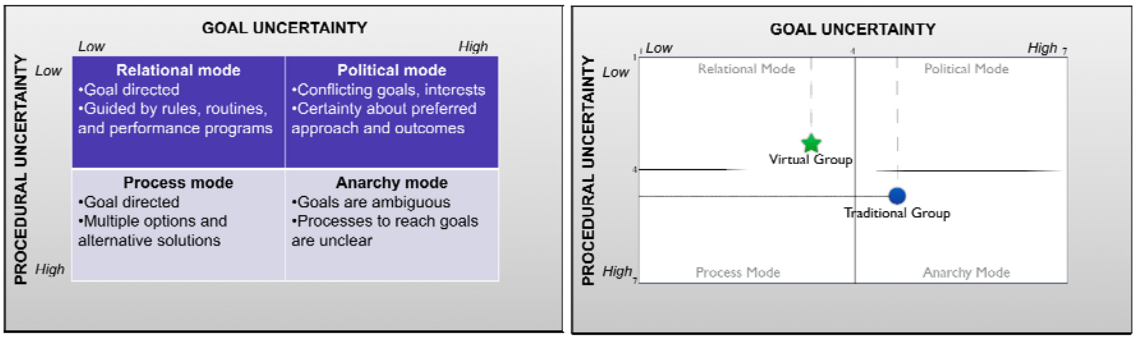

Figure 6. Goal and Procedural Uncertainty Results

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 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 (Hansberger, 2010). In order 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 a measure of uncertainty for each participant. Uncertainty was evaluated along two dimensions, namely *goal* and *procedural* uncertainty. Goal uncertainty is defined as 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. Choo (2005) has defined these uncertainty dimensions in terms of their interactions with each other. The amount of goal and procedural uncertainty possessed by an individual and group will dictate the mode (see figure 6, left) of interactions and ultimately the success of the group. Placing the results for goal and procedural uncertainty along the uncertainty dimensions presents a clear picture of how much uncertainty was involved for each group (figure 6, right). The traditional group finds themselves interacting in the "anarchy mode" where there is ambiguity with both goals and procedures. Group and individual feedback after the experiment confirms this finding. There was considerable effort needed by this group to establish a common ground and understanding within the group 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 have not worked together before. The virtual group using the VCE and collaboration protocol fared much better and found themselves working within the “relational mode” where goals and procedures are clear and understood. The overall difference between the two groups was statistically examined using repeated measures analysis of variance (ANOVA) and there was a significant difference between the two groups as suggested in figure 6 (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.

Conclusions

This paper describes a framework for developing and deploying procedural knowledge in emergency situations where collaboration is needed. Collaborative development of procedural knowledge is supported through an extension to a wiki that allows the marking up of informal SOP knowledge with formal tags that can be processed by an AI planning framework, I-X. Collaborative deployment of procedural knowledge is supported in the wiki itself through enhanced browsing capabilities, 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 has been 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 the user’s actions in an emergency management setting are 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.

Acknowledgements

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

REFERENCES

Barrett, D.J. *MediaWiki* (2009) O’Reilley.

Choo, C. H. (2005). The Knowing Organization: How Organizations Use Information to Construct Meaning, Create Knowledge, and Make Decisions. Oxford University Press, New York, NY.

Cross, R. and Thomas, R. (2009) Driving Results through Social Networks, Jossey-Bass.

Ghallab, M., Nau, D. and Traverso P. (2004) *Automated Planning: Theory and Practice*. Morgan Kaufman.

Hansberger, J. (2010). Understanding distributed collaboration within virtual worlds. *Proc. Sunbelt XXX Social Network Analysis Conference.* Trento, Italy.

Hansberger, J., Tate, A., Moon, B. and Cross, R. (2010) Cognitively Engineering a Virtual Collaboration Environment for Crisis Response, *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Working,* Savannah, Georgia, USA.

Leshed, G., Haber, E., Matthews, T. and Lau, T. (2008) CoScripter: Automating and Sharing How-To Knowledge in the Enterprise. *Proc. CHI 2008,* Florence, Italy.

Lijnse, B., Jansen, J. and Plasmeijer, R. (2012) Incidone: A Task-Oriented Incident Coordination Tool. *Proc. ISCRAM 2012*, Vancouver, Canada.
Tate, A., Dalton, J. and Stader, J. (2002) I-P2- Intelligent Process Panels to Support Coalition Operations. In Proc. of the *Second International Conference on Knowledge Systems for Coalition Operations* (KSCO-2002). Toulouse, France.
Tate A. (2003) <I-N-C-A>: an Ontology for Mixed-initiative Synthesis Tasks. Proceedings of the Workshop on Mixed-Initiative Intelligent Systems (MIIS) at the International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, pp 125-130.
Tate, A., Potter, S. and Dalton, J. (2009), I-Room: a Virtual Space for Emergency Response for the Multinational Planning Augmentation Team, Proc. of the *Fifth International Conference on Knowledge Systems for Coalition Operations* (KSCO-2009) (Lawton, J., Patel, J. and Tate. A. eds.), Chilworth Manor, Southampton, UK.
Wickler, G., Potter, S., Tate, A. (2006) Using I-X Process Panels as Intelligent To-Do Lists for Agent Coordination in Emergency Response, International Journal of Intelligent Control and Systems (IJICS), Special Issue on Emergency Management Systems, Vol. 11, No. 4.
Wickler, G. (2011) Using Planning Domain Features to Facilitate Knowledge Engineering, Proceedings of Knowledge Engineering for Planning and Scheduling (KEPS 2011), Freiburg, Germany.

	Gerhard Wickler is a senior research fellow at the School of Informatics at the University of Edinburgh. He obtained his PhD in Artificial Intelligence in 1999 and held research positions in Europe since. His research interests include AI planning and intelligent agents. Dr. Wickler is 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 the director of the Artificial Intelligence Applications Institute (AIAI) and holds the Personal Chair of Knowledge-Based Systems at the University of Edinburgh. His research interests include emergency response using advanced knowledge and planning technologies, and collaborative systems, especially using virtual worlds. Dr. Tate has a PhD in machine intelligence from the University of Edinburgh. He's a Fellow of the Royal Academy of Engineering and a fellow of AAAI, and an IEEE Intelligent Systems senior advisory board member. Contact him at a.tate@ed.ac.uk.

	Dr. Hansberger is a research psychologist for the US Army Research Laboratory and the Human Research and Engineering Directorate and joined ARL as a Distinguished Scholar in 2003. He received his M.S. and Ph.D. in Human Factors and Applied Cognition from George Mason University. Dr. Hansberger currently is assigned to the ARL, AMCOM Field Element at the Redstone Arsenal, AL where he supports research on expertise, how it is acquired, how it is shared, and how it is enhanced by technology, design, and the environment. Contact him at: jeff.hansberger@us.army.mil.