

VCRI: using shared visualisations for collaboration

Marcel Broeken (m.h.broeken@fss.uu.nl)
Langeveld Research Centre Learning in Interaction
Utrecht University
The Netherlands

Abstract

The VCRI is a Tcl/Tk groupware application developed at the Langeveld Research Centre Learning in Interaction at Utrecht University for research on computer supported collaborative learning. It offers a number of shared visualisations to support collaborative learning. In this paper I will focus on these three important aspects of the VCRI, namely sharing, visualisation and collaboration, and how they relate to each other. The main focus will be on the visualisations as used in three of the VCRI's tools. But first off I'll give a small introduction to the VCRI and its background and I'll finish up with a look into the VCRI's future.

What's the VCRI?

The VCRI (Virtual Collaborative Research Institute) is a multiplatform groupware application with a client-server architecture written entirely in Tcl/Tk. Using the VCRI a community of learners, e.g. secondary school students, can collaborate on an inquiry group task. It provides almost twenty different tools supporting tasks like planning, coordination, communication, data selection and text processing. Different combinations of tools can be used to fit the needs of the task at hand. Users, or members in VCRI terminology, are grouped in groups and groups are grouped in communities. Special users called coaches, e.g. teachers, can monitor users' and groups' progress. Administration of accounts and settings is kept simple thanks to an easy to use administrator interface.

The CRoCiCL (Computerized Representation of Coordination in Collaborative Learning) project focuses on shared visualisations and its effects on collaboration and learning. It uses the VCRI as its main instrument. In September 2006 the project's third experiment will be conducted at two secondary schools in the

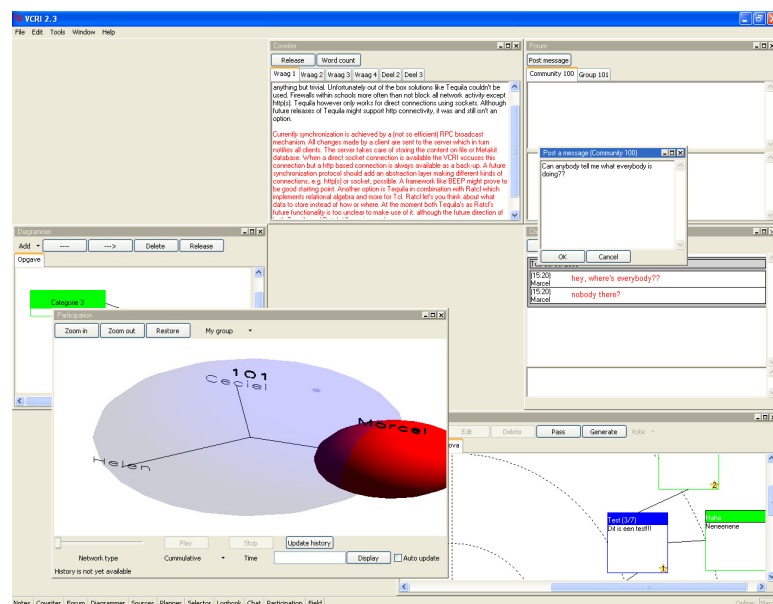


Figure 1: Screenshot of the VCRI

Netherlands. A new tool, the *Commitment Field*, will be introduced. Two earlier experiments, in September 2004 and September 2005, also made use of newly developed tools, the *Participation* and *Shared Space* respectively. All three tools will be dealt with in this paper.

Shared

Collaboration implies working together on shared products. Keeping all members' copies in sync with a master copy on the server and at the same time creating the illusion of working on the same copy (WYSIWIS) are very important aspects of facilitating collaboration. Both will be covered in the next subsections.

Synchronisation

Synchronisation has proven to be one of the biggest challenges in the development of the VCRI. Keeping all copies in sync, including the master copy on the server, preventing or resolving conflicts and persistency of data have turned out to be anything but trivial. Unfortunately out of the box solutions like Tequila couldn't be used. Firewalls within schools more often than not block all network activity except http(s). Tequila however only works for direct connections using sockets. Although future releases of Tequila might support http connectivity, it was and still isn't an option.

Currently synchronisation is achieved by a (not so efficient) RPC (remote procedure call) broadcast mechanism. All changes made by a client are sent to the server which in turn notifies all clients. The server takes care of storing the content on file or in a Metakit database. When a direct socket connection can be established the VCRI uses this connection but a http based connection is always available as a back-up. A future synchronization protocol should add an abstraction layer making different kinds of connections, e.g. http(s) or sockets, possible. A framework like BEEP might prove to be a good starting point. Another option is Tequila in combination with Ratcl which "implements relational algebra and more for Tcl". Ratcl let's you think about what data to store instead of how or where. At the moment both Tequila's as Ratcl's future functionality are too unclear to make use of it.

WYSIWIS

One of the VCRI's most important features is WYSIWIS (what you see is what I see). Group members working on a task share all their tools (except Notes for personal notes). Tool content is synchronised between all members creating the illusion of working on one product together instead of working individually on a personal virtual copy. Examples of tools with shared content are the *Cowriter* (text), the *Diagrammer* (diagram/schema) and *Selector* (annotated bookmarks). WYSIWIS not only makes this collaboration easier, it also increases the awareness of the group processes and activities taking place. Several researchers have suggested that awareness can play an important role in facilitating CSCL (computer supported collaborative learning). When members are collaborating, they have to be aware of the activities of their group members, because this awareness allows members to decide which activities they have to engage in, and it enables them to anticipate group members' actions. In short, awareness is expected to facilitate smooth and natural collaboration.

Awareness is not only stimulated by the tools but also by the taskbar. Members use the taskbar to navigate between all different tools but it also provides visual feedback about

the group members' activities. First off it shows which members are online. Secondly it shows which tool is being used by which member. When a member uses a tool, the label on the other members' taskbars corresponding to the tool blinks in the member's colour.

Collaboration and user experience

Studying the effects of shared visualisations on collaboration is of little use when collaboration is cumbersome. I have tried to make collaboration as easy as possible. Lacking any good human interface guidelines for groupware I have used feedback from users and information sciences students to improve the VCRI's user experience. Some of these improvements will be discussed in the next section. First I'll focus on the overall VCRI user experience. Secondly I'll take a look at the *Participation*. Finally the evolution of the *Cowriter*'s user interface will be looked at.

VCRI's user experience

The VCRI is basically a collection of tools within one application. One of the biggest user interface problems is to create a distinct and recognizable look and feel for the VCRI and at the same time respect the conventions of the operating system. Because the VCRI runs on multiple platforms (Windows, Mac OS X and Linux) there's not one set of conventions, making this even harder. With the help of students at information sciences at Utrecht University we've tried to make the VCRI consistent with the operating system, consistent within itself and yet still recognizable.

Tile, a re-implementation of many of the core Tk widgets, has been used to give the VCRI a native look and feel. Tile is highly customizable by defining themes, leaving the option of a VCRI theme open. Configuring and tweaking buttons, frames, notebooks, etc is no longer necessary and yet the VCRI looks slicker than ever with less effort, especially on Windows XP. Configuration options outside Tile's scope, like button placement, colour usage and terminology, have been set as consistently as possible. Furthermore the msgcat package has been used to make the VCRI multilingual. Currently multilingual means bilingual, namely English and Dutch.

Participation's usability mistakes

The *Participation* was one of the tools to receive most criticism from the usability survey conducted by the information sciences students and therefore one of the tools to benefit most. Although most changes seem insignificant they have improved the user experience a great deal. I'll illustrate this with a couple of examples.

Participation features a playback function allowing the member to view the development of the participation visualisation through time. In the first version users could replay the participation visualisation by pushing a 'Replay' button. During playback this button changed into a 'Cancel' button which could be used to cancel the playback. Although this seems logical it isn't. Users are used to terminology like 'play', 'pause' and 'stop' when playing something.

There's also a progress bar which can be used to jump to a specific time in the participation history. In earlier versions this progress bar didn't move when replaying the participation although this is to be expected. Both mistakes are fixed in the latest version, making this tool more intuitive to use.

Another usability mistake was the interface to the zoom and view functionality. Members can zoom in and out on the visualisation or change views between their group and the

community. This functionality could be accessed only through a contextual menu. It turned out members found it difficult to locate this menu. This was remedied by adding a toolbar with comparable functionality to the *Participation*.

***Cowriter* evolutions**

The *Cowriter* is a collaborative text editor allowing users to simultaneously edit a text. By using this tool it becomes a lot easier to work on a shared text. It's no longer necessary to send different versions of the text back and forth. All changes are instantly visible and users can react on each other's edits. A technical problem arises due to the fact that simultaneous editing can cause conflicts when two users edit the same piece of a text differently possibly resulting in a conflict. To prevent problems like inconsistent versions conflicts must be prevented or resolved. The *Cowriter* enforces conflict prevention through a locking mechanism. This locking mechanism and its user interface have evolved through all the VCRI's incarnations.

An earlier version of the *Cowriter* was called TC3 (Text Composer, Collaborative & Computer-supported). In TC3 conflicts were prevented by using a locking mechanism which used the complete text as a locking unit. This means the entire text is either locked or 'available'. The state was visualised using a traffic light. If the light is green, the text is available. If the light is red, it is locked. Although this mechanism is very easy to understand and use it also caused frustration because group members had to take turns editing their shared text. This turn taking made it seem less of a group effort. This problem was tackled in the next version.

The next version had the ability to use the *Cowriter* for more than one text. This made it possible for members to work simultaneously on different texts making things a little bit better. To further increase the idea of real-time collaborative text editing the locking unit was decreased to pieces of text. The same locking mechanism was used but instead of (un)locking the entire text it became possible to (un)lock a selection. The traffic light visualisation was replaced by a visualisation using background colouring of the locked pieces of text. The colour of the member locking the text is used as background colour. This change in the *Cowriter* fixed most of the first version's problems. However feedback from users indicated that locking didn't work intuitive. The *Cowriter* was one of the few tools where users had to read the help files before use, indicating it was not as straightforward as it could be.

The current version still uses the same locking mechanism behind the scenes. However, to the user it seems as if editing is possible at any time and at any place in the text. Before each edit the *Cowriter* automatically (un)locks the current paragraph, allowing the user to edit as he pleases and meanwhile preventing conflicts. Use in real life will have to prove if this solution works as intuitive, easy and transparent as hoped.

Future versions of the *Cowriter* might use optimistic concurrency control like SubEthaEdit does. This means that conflicts aren't prevented but are resolved using an intelligent set of heuristics.

Visualisations

The first two letters in CROCiCL stand for computerized representation. These representations or visualisations (combined with sharing and collaborating) are one of the most important aspects of the CROCiCL project, and quite unique in CSCLE environments. They might (positively) influence (the coordination of) collaboration by providing group members with insight in their collaboration. Three of these visualisation

tools, *Participation*, *Shared Space* and *Commitment Field*, have been used as 'treatment' in three different studies and will be discussed below.

Participation

Participation is an automatically generated visualisation of participation rates. Participation rate is a measure of how much each group member contributes to his group's online communication. Using the *Participation*, a member can compare his own participation rate to the participation rates of his group members. Each member is represented by a sphere; group members' spheres are grouped. The distance of a sphere to the group centre indicates the number of messages sent by the member compared to the other group members. A message is a contribution to either the *Chat* (instant messaging) or the *Forum* (bulletin board). If a sphere is close to the centre, the member has sent more messages compared to a member who is farther away from the centre. The size of a sphere indicates the length of the contributions by a member compared to the other group members. A contribution is measured in keystrokes. Keystrokes in all tools are taken into account. If a sphere is smaller, the member has sent shorter contributions compared to a member whose sphere is bigger. Participation of groups can be compared by examining the grey circles. These indicate a group's total participation, compared to other groups. A bigger grey circle indicates more participation and the closer this grey circle is positioned to the centre of the community the more messages this group has sent.

The *Participation* visualises for the most part the quantity of the communication between group members, and to a lesser extent the quality of communication. Nonetheless, quantity of communication is important for successful collaboration. When unequal participation exists between group members, this is an indication some of the group members might be a free rider. Quality and quantity of participation are both important for collaboration. If a group member only types a few messages, he cannot be regarded a full-fledged group member, even though his messages may be of high quality.

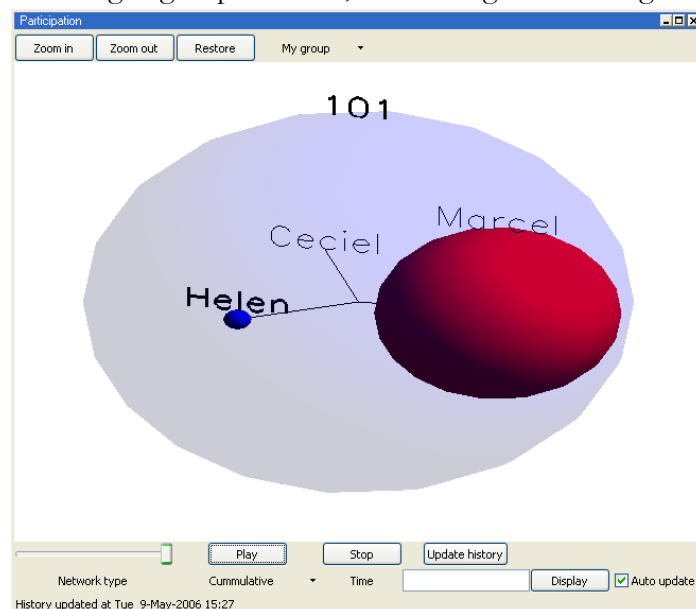


Figure 2: Screenshot of the *Participation*

The *Participation* can be opened and examined by members at any time. Furthermore, the *Participation* can be continually updated in order to display the most recent situation.

Members can zoom in to examine a part of the visualisation more closely, while the visualisation can also be rotated using the mouse to examine the visualisation from a different perspective. The *Participation* can display members' cumulative participation rates (i.e. total number of messages sent), but can also display a moving average.

The moving average displays members' participation rates during the past twenty minutes. It also allows for replaying the 'participation history' or jumping to a specific moment in this history. Furthermore members can switch between a view which shows only their group and a view which

shows the entire community. Finally, it is important to note that members are not forced to use the *Participation*. In other words, it is available and members can use it whenever they want, but they can also choose to ignore (and close) it whenever they want. As described above the *Participation* is a visualisation of the participation rates of groups and members. It uses tcl3d v. 0.3 (formerly known as tclogl) to create a three-dimensional representation of the participation. Earlier versions of the VCRI used TkOGL. Both tclogl and TkOGL provide access to OpenGL, a 3D graphics library, which allows the use of OpenGL commands in Tcl code without almost any modifications. TkOGL turned out to have some limitations which rendered it less suitable. The most important limitations were the lack of support for stubs, which makes it no longer necessary to recompile binary extension for each new Tcl version, and support for other platforms than Windows or Debian Linux. Tcl3d has been stubs-enabled from the start and by using SWIG (simplified wrapper and interface generator) it's made easy to compile tcl3d on almost any platform. Currently prebuilt binaries for Windows, Linux and Mac OS X are available at tcl3d's website. Switching from TkOGL to tcl3d turned out to be a piece of cake and has proven to be a smart move. Tcl3d doesn't suffer from TkOGL's drawbacks, but it's also regularly updated with excellent backwards compatibility.

The results from the research using the *Participation* turned out to be interesting. The tool was used fairly intensively. On average, students displayed the *Participation* on their screen 18% of the total time they were online. Students with access to the tool sent more long messages (more than five words) compared to students with no access. These findings indicate that the *Participation* stimulated students to invest more effort into collaboration and to type more substantial messages. Furthermore the students reported they knew better when a group member was not working hard.

Shared Space

Shared Space is another automatically generated visualisation. It's a special version of the *Chat*, providing the same functionality but extended with the visualisation. It analyses all messages sent. First, the *Shared Space* discerns discourse topics based on time intervals. When members do not send messages for more than 59 seconds, a new topic begins. Secondly, the *Shared Space* analyses the content of each chat message in order to determine whether a chat message indicates discussion or agreement. For this purpose, the *Shared Space* determines the communicative function of the message. This is done using the DAC (dialogue act coding) filter. This filter uses over 1300 rules based on discourse markers to determine the communicative function of a chat message. Discourse markers are characteristic words or phrases signalling the communicative function of a message. In total, five main categories of communicative functions are distinguished: argumentative, responsive, informative, elicitive and imperative. Each category consists of several subcategories. For example, the elicitive category consists of verification questions, open questions, set questions, and proposals for action. In total, 29 different dialogue acts are distinguished. Of these, confirmations, acknowledgements, and positive evaluations are considered indications of agreement, while denials, verification questions, negative evaluations and counterarguments are considered indications of discussion or debate.

Finally, after establishing whether the chat message indicates discussion or agreement, the *Shared Space* moves the whole topic to the left or to the right in small steps. When the chat message indicates discussion, the *Shared Space* moves the topic to the left; when it indicates agreement, it moves the topic to the right. The movement of the topics

corresponds to the discussion and agreement headings underneath the chat history. The lines above the topics visualise the development of the online discussion.

Both *Shared Space* and its super class *Chat* are implemented as normal Canvas widgets. Topics are rectangles and messages are Text widgets positioned on the Canvas as window items. This way default functionality for Text widgets like copy and selection is instantly available. It also simplifies substituting emoticons with images using Text's

search command. Using tags to identify all items

belonging to a topic makes manipulating (moving, adding, deleting) topics relatively easy. The only differences between the *Shared Space* and *Chat* user interface are the topics, the horizontal movement and the lines above the topics visualising the development of the online discussion. The analysis of the chat messages is the only difference between both tools as far as business logic is concerned.

Because of these small differences in functionality between *Shared Space* and *Chat* it's proven a smart

move to implement *Shared Space* as a subclass of *Chat* thus reducing the complexity and simplifying code maintenance.

The study using the *Shared Space* also produced some interesting results. They show that students with access to the *Shared Space* visualisation tend to perceive the CSCL environment as more media rich, although this effect only approached significance. Secondly, the results show an effect of the *Shared Space* on group norms. Students with access to the *Shared Space* visualisation report more critical group norms. Third, students with access to the *Shared Space* visualisation reported more positive group behaviour and perceived their group's task strategies as more effective. The *Shared Space* visualisation only had a small effect on students' collaborative activities. Students with access to the *Shared Space* visualisation asked less task-related questions, were less busy maintaining and indicating shared understanding, and made less negative remarks about the CSCL environment.

In all, these results show that the *Shared Space* had a positive effect. First, the *Shared Space* seemed to facilitate communication and discussion. Students perceived the medium as marginally more media rich. Furthermore, students needed to devote less effort to maintaining and signalling shared understanding. Second, the *Shared Space* stimulated a more critical, exploratory group norm perception. Treatment group students perceived their group as more engaged in a constructive but critical online discussion. Finally, the *Shared Space* influenced students to their group's behaviour as more positive and their group's task strategies as more effective.

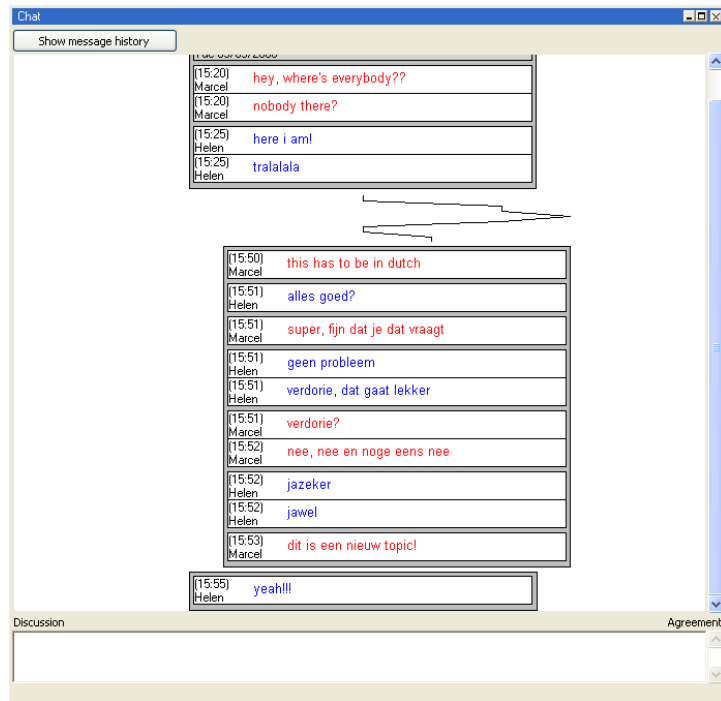


Figure 3: Screenshot of the *Shared Space*

Commitment Field

The *Commitment Field* is the latest addition to the VCRI. This tool allows members to construct a diagram on a topic with different positions. Making use of the available sources a member can assign arguments, pros and cons to the different positions. The *Commitment Field* visualises real-time the complexity and 'argumentative power' of each position. This visualisation can aid the member in determining which position requires more investigation or is better supported. This tool will be the next experiment's 'treatment'. The experiment will try to answer whether or not the *Commitment Field* influences the argumentation.

Members can add new arguments, including supports and refutations, to different positions on a given topic. The arguments can be linked to the Sources (extensible collection of websites) or any website. All items can be edited or deleted. Members can add (or remove) votes to show which items they value more. Whereas the calculated scores are primarily quantitative the votes add a qualitative dimension to the tool. The *Commitment Field* uses rounds to evoke equal participation of all members. To prevent one member from doing all the work or free-riding all online members can perform one action per round. An action is adding, editing or deleting an item, voting or passing. The next round starts when all members have performed an action thus making sure that everyone at least does something.

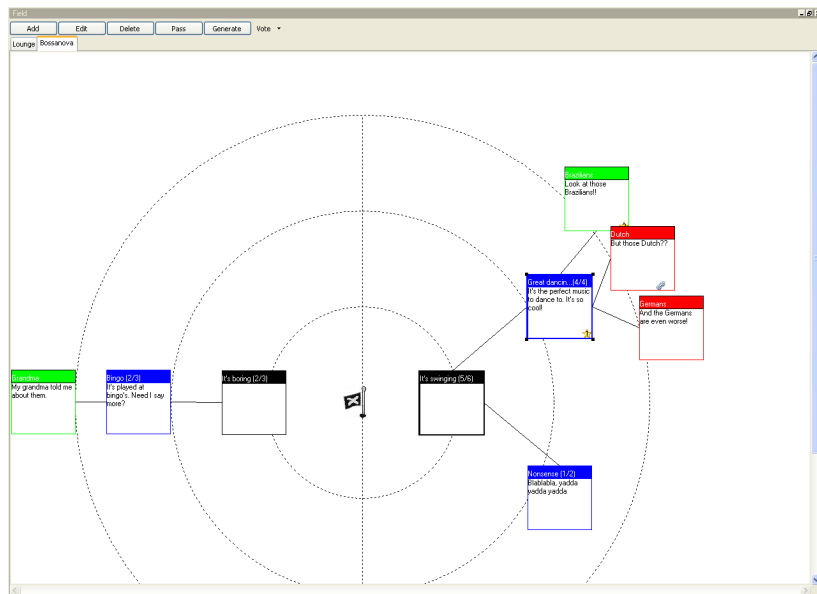


Figure 4: Screenshot of the Commitment Field

The *Commitment Field* has a Model-View-Controller architecture. Using this design, different views could be reviewed while the model (the data) remained the same. The chosen view is a circular representation on a canvas, see Figure 4. The scores for complexity and 'argumentative power' and the hierarchy of

arguments and positions determine the positions on the canvas. Complexity is visualised by the width of the border around arguments and positions. The wider the border is, the more complex this item is. Argumentative power is visualised by the position of the positions in the field. The greater the power, the more centred the position is. Changes to the items, like edits or votes, influence the field representation instantly and all changes are visible to all group members. The *Commitment Field* is still under development but should reach its final version mid-summer.

The *Commitment Field* will be used in a study on two secondary schools in September 2006.

Looking forward

In the near future the development of the VCRI will head in a different direction. The code will go open source, it will be released for the educational market and it will be used in a large European research project.

KP-Lab

Utrecht University is one of the partners in KP-Lab (Knowledge Practices Laboratory), an European research project which focuses on creating a learning system aimed at facilitating innovative practices of sharing, creating and working with knowledge in education and workplaces. These objectives have a great deal in common with the VCRI's functionality. The VCRI itself and its ideas might provide inspiration for this project which in turn could benefit the future development of the VCRI.

Educational release

After the third and last experiment of the CRoCiCL project it's the intention to release the VCRI for educational use. Not only is an educational release required by NWO (Netherlands Organisation for Scientific Research), there is also a fair amount of interest of practitioners to use the VCRI in schools. As of yet setting up the VCRI environment isn't as easy as it could be. Before release VCRI development will focus on easy installation and administration. Most data and settings for the VCRI are stored in a Metakit database on the server. Administration for this database is partly accomplished using Metakit@Web. Metakit@Web is a web-based admin-interface for Metakit databases. Although this works fine for (system) administrators with basic knowledge of the VCRI's inner workings it's too intricate for the 'average' user. Administration has already been greatly improved by implementing an *Administrator* tool. This tool is a VCRI tool like all others providing access to most of the VCRI settings and data. Administration functionality will move away from Metakit@Web to the *Administrator* thus making it more suitable for use in a normal school environment.

Open source

Besides an educational release there are also plans for opening up the VCRI's source code allowing third party developers to extend and improve the VCRI and its tools. More work will be spent on refactoring and modularisation of the code to make this easier. This should result in a framework which focuses on extensibility. Adding a new tool should be as easy as adding plug-ins to your web browser. The VCRI already is designed to make updating easy. Most changes in the VCRI code are server-side. Members can keep using their versions of the client while the server offers new functionality. The need for updates at client-side can be even further reduced by using Starsync, a technology to fetch and update Starkits over the net. It sends only changed files, in compressed form. This way the VCRI can be continually updated without having to force people to 'install' a new client. Making the VCRI open source in combination with an educational release could make the VCRI an interesting environment for online collaboration in primary and secondary schools as well as higher education.

Summary

The key features of the VCRI are the shared visualisations which add support for collaborative learning. The three visualisations in this paper, Participation, Shared Space and Commitment Field, support different aspects of collaborative learning, participation, discussion and argumentation respectively.

These tools combined with all other tools make the VCRI well suitable for use in primary and secondary schools as well as universities when working on collaborative tasks. It looks like the VCRI has a hopeful future!

References

- Tequila remote arrays for Tcl, <http://www.equi4.com/tequila.html>
- Metakit by Equi4 Software, <http://www.equi4.com/metakit.html>
- BEEP Core protocol central, RFC 3080, RFC 3081, and related BEEP profiles, BEEP implementation, BEEP Protocol, BEEP, Blocks Extensible Exchange Protocol, <http://www.beepcore.org>
- Vlerq: Ratcl, <http://www.vlerq.org/vlerq/ratcl>
- Tile: an improved themeing engine for Tk, <http://tktable.sourceforge.net/tile>
- Tcl Bundled Packages - msgcat manual, <http://www.tcl.tk/man/tcl8.4/TclCmd/msgcat.htm>
- SubEthaEdit, <http://www.codingmonkeys.de/subethaedit/index.html>
- Tcl3D: Overview, <http://www.tcl3d.org>
- Welcome to KP-Lab & KP-Lab Web Site, <http://www.kp-lab.org>
- Vogel-Nest - MetakitAtWeb, <http://vogel-nest.de/wiki/Main/MetakitAtWeb>
- Starsync, <http://www.equi4.com/264>

- Janssen, J., Erkens, G, Jaspers, J., & Broeken, M. (2005). *Effects of visualizing participation in computer-supported collaborative learning (CSCL)*. In C.P. Constantinou, D. Demetriou, A. Evagorou, M. Evagorou, A. Kofteros, M. Michael, Chr. Nicolaou, D. Papademetriou & N. Papadouris (Eds.), *11th European Conference for Research on Learning and Instruction: Multiple perspectives on effective learning environments* (pp. 484-485). Nicosia, Cyprus: University of Cyprus.
- Janssen, J., Erkens, G, Jaspers, J., & Broeken, M. (2006). *Visualisation of Agreement and Discussion Processes during Online Collaborative Learning*. To be presented at *CIP&T SIG: Avoiding simplicity, confronting complexity: Advances in studying and designing powerful (computer-based) learning environments*, Leuven, Belgium: Katholieke Universiteit Leuven.