



E-Learn-VIP Consortium Sample System

Virtual Conference Systems (VCS)

Index

1	Introduction	2
2	Screen Layout	2
2.1	Example of use of separate screens	2
2.2	Windows structure in program.....	3
2.3	Use of zones	3
3	Features of an Accessible VCS.....	3
3.1	Notification Centre.....	3
3.2	Voice Communication	4
3.3	Chat	6
3.4	Application Sharing	7
3.5	Whiteboard.....	8
3.6	Agenda.....	8
3.7	Presentation	9
3.8	Non-verbal/textual communication elements.....	9
3.9	Quick Poll.....	10

1 Introduction

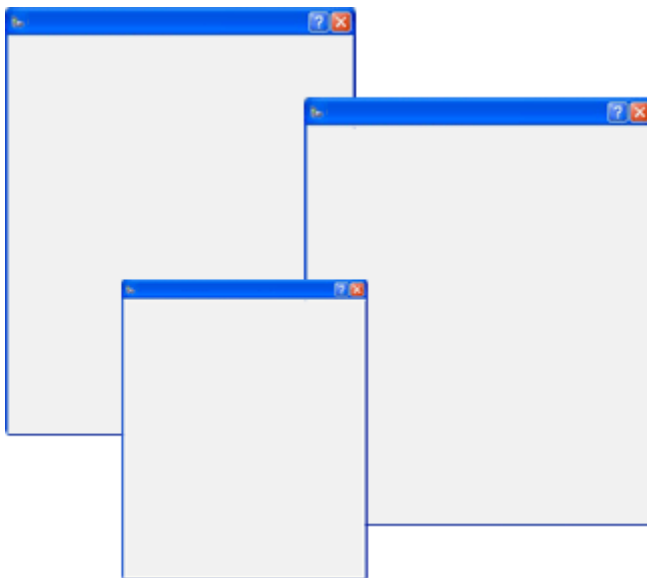
This part of the guidelines points out the technical aspects for accessibility of VCS features. To create accessible VCS it is strictly recommended to also consider the aspects which are pointed out in the part “Common guidelines” and “Specific Guidelines”.

[to the index](#)

2 Screen Layout

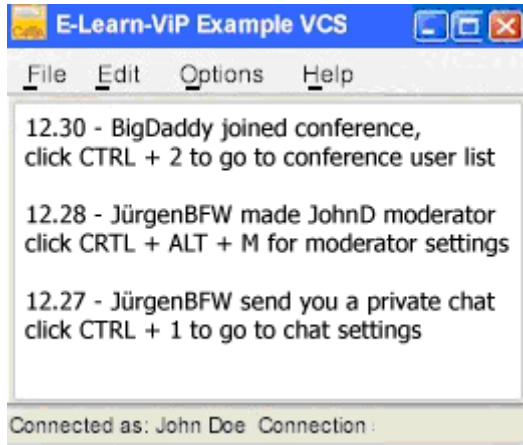
The screen layout of an accessible VCS make is possible to change focus per feature. The VCS should therefore consist of a) separate screens (windows) per feature, b) have a windows structure in the program or 3) the ability to define zones in the program. This structure will ensure a correct focus for assistive technology users.

5.1 Example of use of separate screens



this from happening it is important that the focus of each feature stay there where the user is working. The user needs to manually change focus.

To have access to all information that changed during a period the user has access to the Notification Centre. Here all changes are clearly stated and have tips on how to access these functions.

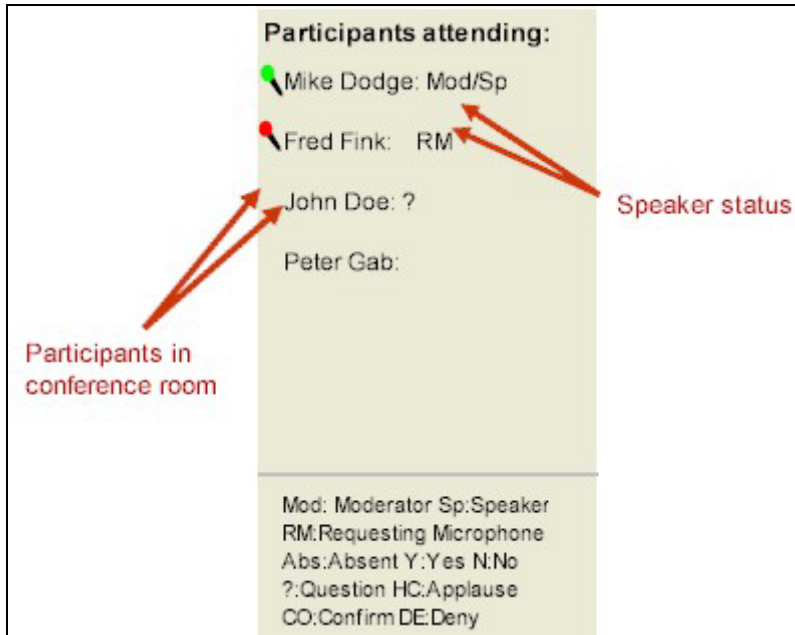


This central point in the VCS would give a line by line change status like: 12.28 – JürgenBFW made JohnD moderator. Click CTRL + ALT + M for moderator settings.

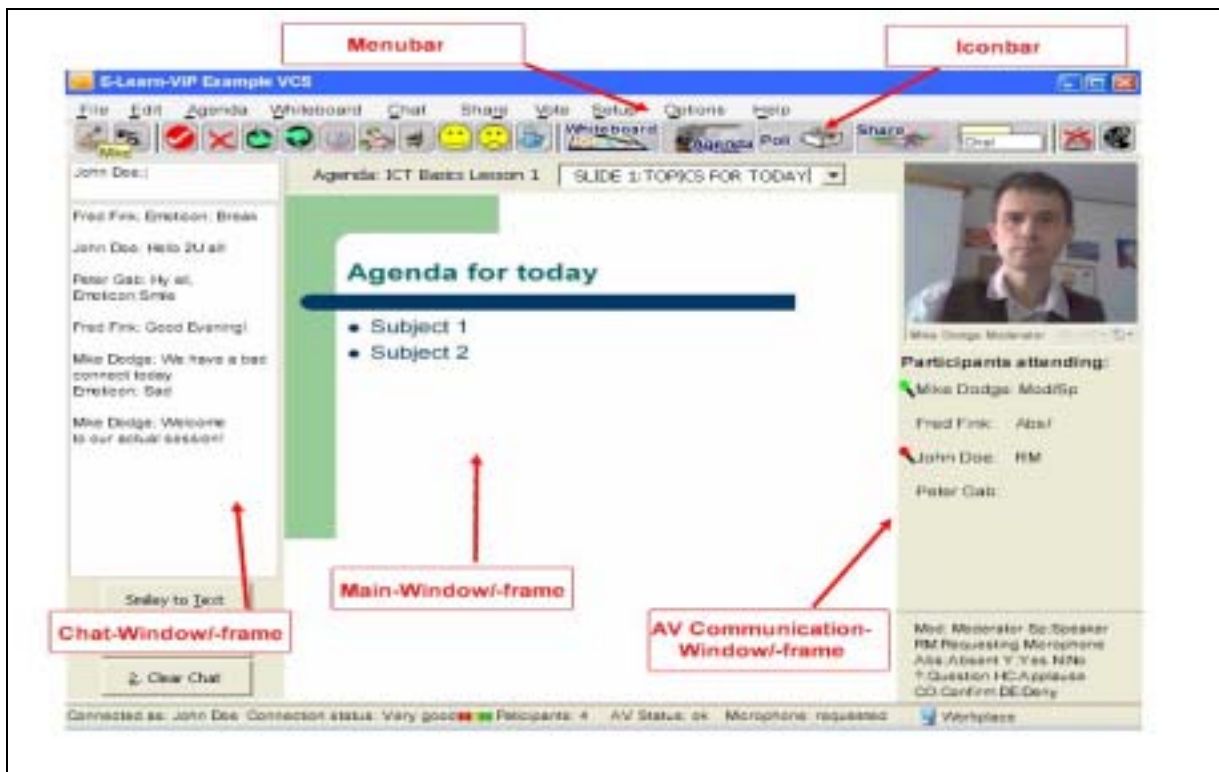
The Notification Centre can have options for sound notifications so each type of change can have a separate or no sound. Also settings for navigation tips (shortcut keys for quick access) could be a valuable option in the Notification Center.

5.5 Voice Communication

Voice communication is one of the main features of a VCS and like all other features accessible control features are important. Options like taking turns, inviting a moderator or sending a question proposal should be accessible via shortcut keys and clearly visible on the screen. Easy access to the status of participants and all other status features should be accessible and get a focus when needed.



Voice is the most important features for our target group and should therefore get the highest quality of transportation on the network and in the program. A presentation, video should not interfere with the quality of voice.



Some features of the Voice Communicator are:

- Status of users
- Moderator visibility
- Participant question feature
- Applause function

- A confirm and deny option to respond to questions easily.

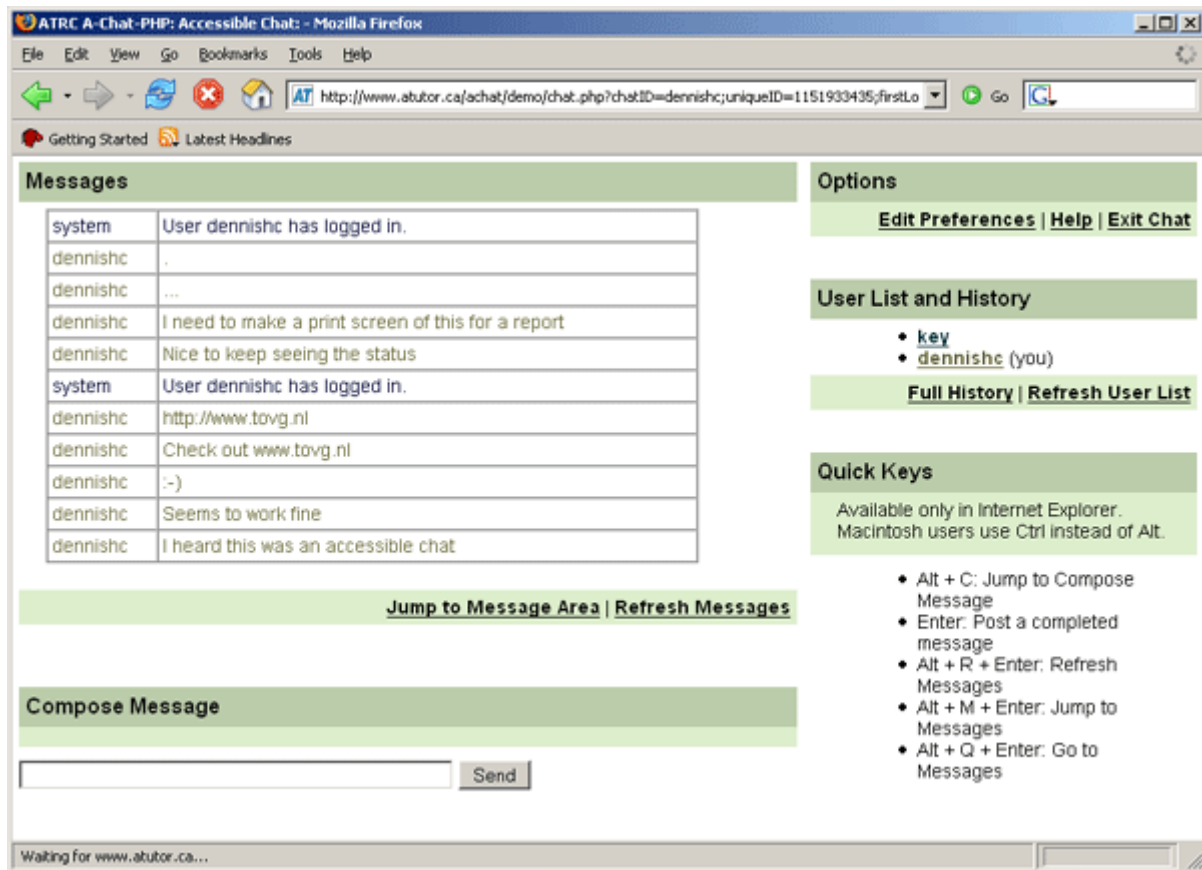
5.6 Chat

Second most important feature of a VCS after voice communication is the chat function. This one presents no difficulties on accessibility if the following features are built in:

- Complete access and focus control to all areas and objects
- Private chat options for participants
- Emoticons (also known as “smileys”) that are automatically translated to text (☺ should become the word ‘smile’)
- Page refresh control (manual or automated with time settings)
- New message pop-up (on and off switch)
- Sorting option for new messages (on top or at the bottom)
- The option ‘show new messages only’.
- Option for inclusion of quick jumps (internal pagelinks).



An example of a accessible chat is the A-Chat-PHP feature of Atutor at: <http://www.atutor.ca/achat/demo/login.php> of which a screenshot is present under this paragraph.

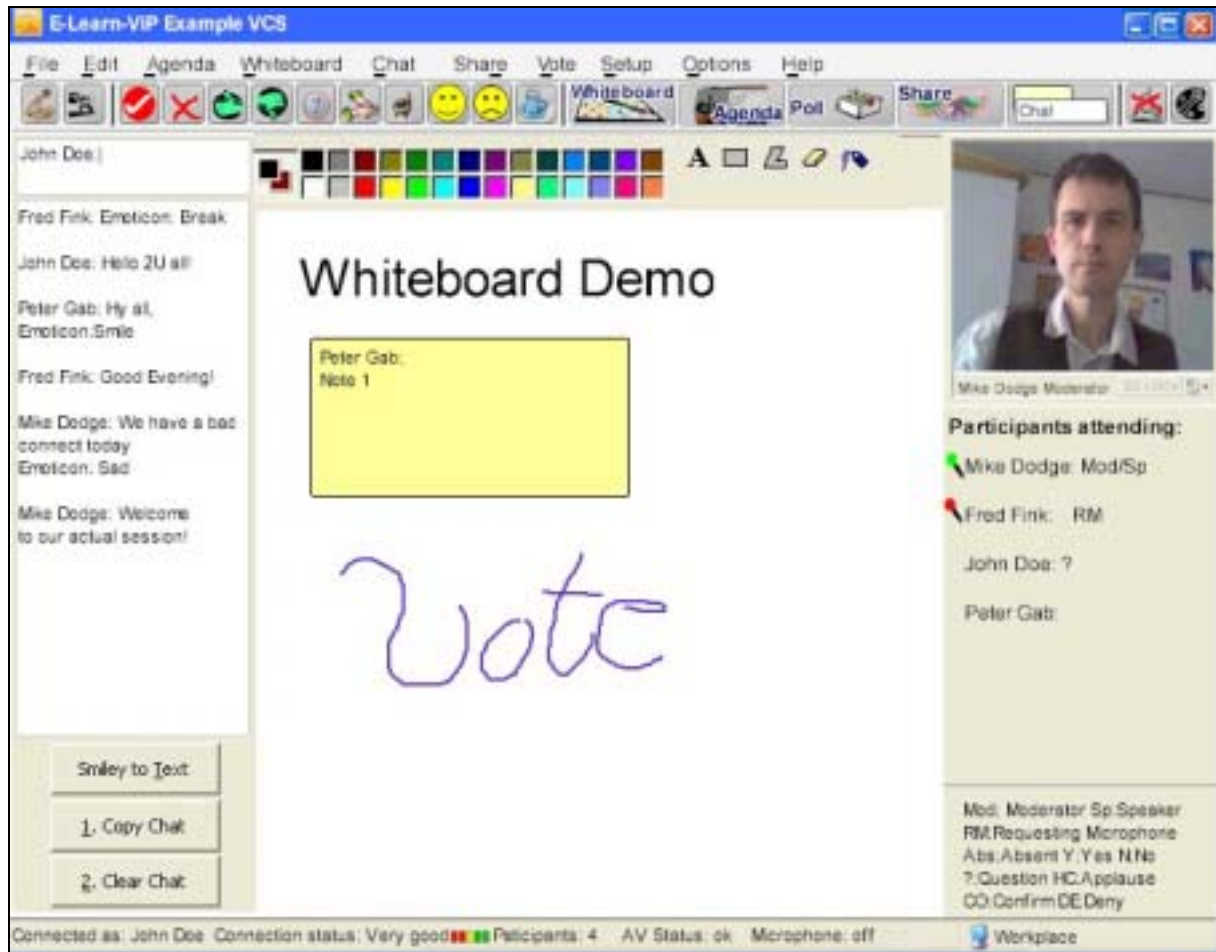


5.7 Application Sharing

We have seen working examples of active application sharing in example systems like Citrix and windows terminal server that are barely usable for magnification users, but have very little working accessible versions all over the world. This kind of application sharing in a VCS will be very hard to program with the current state of technology. Assistive technologies (AT) with speech need high bandwidths and AT-software at both ends.

All accessibility guidelines for access to software apply to the application sharing feature (Guidelines http://trace.wisc.edu/docs/software_guidelines/toc.htm). In general if the application on the server side is accessible the same accessibility features must be offered on the client side.

5.8 Whiteboard



Since drawings are not always accessible via mainstream software solutions like a screenreader we recommend vector images are recommended since they provide a sharp image even after magnification. to include a text editor for this user group. For people that use magnification.

In addition to this an accessible whiteboard should include:

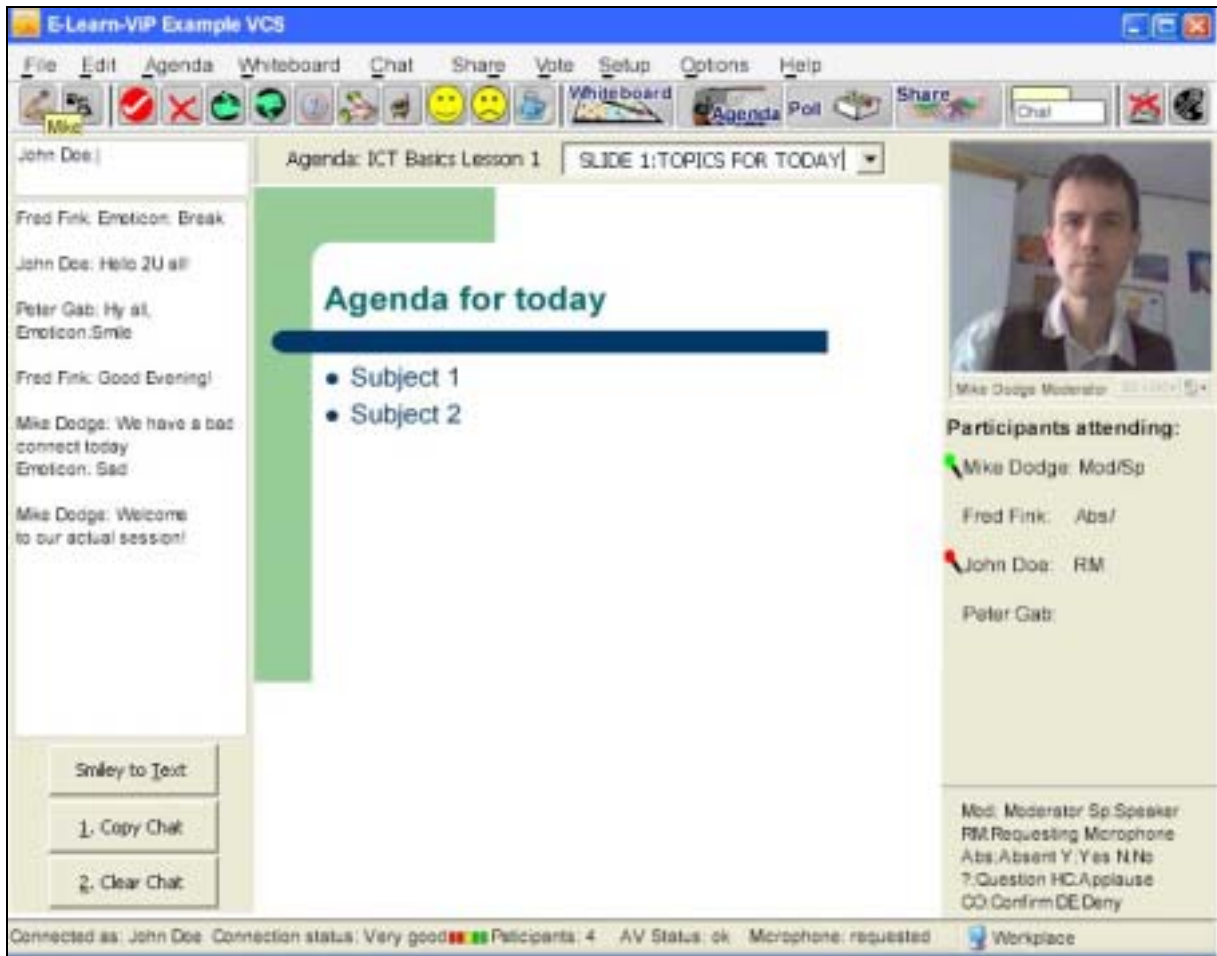
- Full stylesheet support
- If SVG is used: labels and hierarchical grouping of objects.
- Keyboard support on all features (including drawing and colouring).

Some best practises on accessible whiteboards are:

- A-Communicator (ACom is included in ATutor)
<http://www.atutor.ca/acomm/index.php>

5.9 Agenda

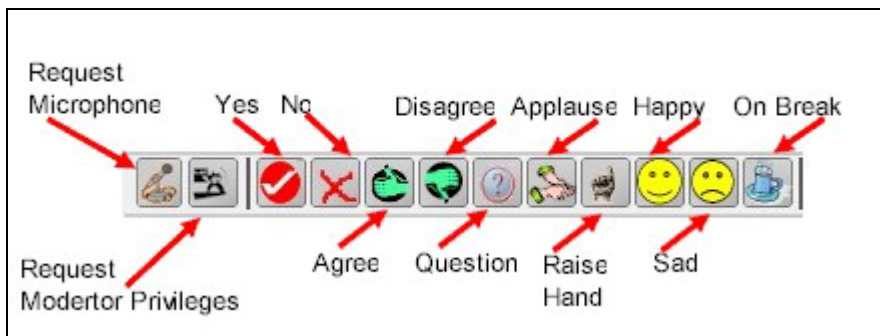
The agenda brings structure to the meeting and has the option to include attachments to the agenda points. These attachments can be documents, images, presentations and links to VCS tools or features. The agenda should be in a list, and each text accessible and links / attachments well described.



5.10 Presentation

The moderator should be able to offer a presentation. If PowerPoint is used this presentation will be a shared application. Other means of linear presenting should be included in the VCS. Text should be readable with screenreaders and individual grouped or zones in a presentation must be keyboard accessible.

5.11 Non-verbal/textual communication elements



All Non-Verbal communication elements like buttons for: yes, no, agree, disagree, question, applause, raise hand, happy, sad, on break etc. Should result in a readable text in either the chat or in other parts of the VCS. These non-verbal elements should be treated exactly like emoticons.

5.12 Quick Poll

Besides the use of non-verbal elements like yes and no to answer a question a poll should be implemented to pick the best option between participants. A poll consists of two or more answers to a question and have options to vote for one or more options.

The question, answers and results should be accessible. The best way to get multiple choice questions accessible is putting the button (checkbox) behind the answer and presenting the results in percentages so that there accessible for all users. All elements of selection in the poll should be keyboard accessible.

[to the index](#)