

# ZAKIM – A MULTIMODAL SOFTWARE SYSTEM FOR LARGE-SCALE TELECONFERENCING

*Max Froumentin*

World Wide Web Consortium  
mf@w3.org

## 1. INTRODUCTION

This paper describes Zakim, a multimodal teleconference system used at the World Wide Web Consortium [1]. While the technology presented here does not introduce advanced research work, the it has been developed with robustness in mind, as it is used almost round-the-clock by several hundred people. The context and requirements are introduced first, followed by a description of the system's features. Lastly we describe how the structure of W3C teleconferences has been modified by the use of Zakim, and discuss some issues and possible improvements.

## 2. CONTEXT

The World Wide Web Consortium (W3C) gathers experts both from industry and academia, to work on developing the architecture of the Web. This mostly consists in writing documents: specifications (e.g. HTML, CSS), guidelines or technical reports, which once finalised can be used by software developers to write browsers or by webmasters to design sites, among other usages.

Specifications published by the W3C are written by Working Groups. There are about 50 such groups, each of which comprising between 10 and 90 people. Groups communicate through four means: a mailing list (archived on the web), the group's web site, teleconferences (with up to 50 people attending), and face-to-face meetings. Typically a group will hold one or two one-hour teleconferences every week, hold quarterly face-to-face meetings, and exchange 20 to 150 emails a month. Telephone and face-to-face discussions are lead by the group's *chair*, and are recorded by a *scribe* who is in charge of sending meeting minutes to the group's mailing list (scribes rotate among the group's membership).

One basic principle of W3C groups is to record as much of the discussion as possible on the web, whether they occurred on the mailing list, in teleconferences or at face-to-face meetings.

Running meetings for almost 10 years has empirically made people aware of issues that are well-known to researchers

in CSCW, but which we detail below to highlight those that are addressed by the architecture described here:

- Mailing list discussions can be very verbose, as well as quite inefficient, especially when it comes to making decisions.
- Face-to-face meetings are much more time-efficient, but because of the distributed nature of the W3C membership, their frequency is necessarily limited.
- Teleconferences are next best to face-to-face meetings, but the lack of visual cues (who is speaking, who would like to speak and in what order, etc.) and the occasionally poor sound quality (preventing accurate minuting) can be detrimental to efficiency.

Zakim has mostly improved the situation described by the third point above, although it will be mentioned that means other than teleconferences are influenced by the design as well.

## 3. THE ZAKIM TELEPHONE BRIDGE AND ITS INTERFACES

The W3C runs its own teleconference system (*bridge*), which is usually active 12 hours a day on average, with up to 9 simultaneous teleconferences. The Zakim<sup>1</sup> bridge is a customized Compunetix Context<sup>®</sup> conferencing system [2]. To the core teleconference system has been added a few software components, as well as interfaces between them, described below.

### 3.1. Components

#### 3.1.1. The Telephone Bridge

The functionalities of the bridge itself, upon which all the software described below builds are simple. The bridge can

---

<sup>1</sup>The name Zakim originates from the way W3C names the various teleconference bridges it uses, after the names of the vehicle bridges that cross the Charles River in Boston.

hold several teleconferences simultaneously, it has a programmable schedule that stores all the teleconferences, their time, duration and maximum number of participants. Each teleconference is identified by a pass-code, that the participants dial in through DTMF, after calling the bridge's phone number.

The bridge also retrieves and makes available Automatic Number Identification data (ANI, more familiarly known as caller-ID) when available, and can also react to DTMF tones. The original interface, proprietary software provided by the bridge vendor, lets the operator monitor the activity of each line (noise or voice), and allows various operations such as muting or unmuting a participant, call a number or disconnect a line.

### 3.1.2. IRC

The *Internet Relay Chat* [3] is a internet protocol for real-time forum systems. It was chosen by W3C mostly because of the fact that it is an open protocol, and is thus implemented by many pieces of software on nearly all existing operating systems. W3C runs its own IRC server, and people connecting to it can create discussion forums as they require. In particular, Working Groups often have their own discussion channels, usually active during telephone or face-to-face meetings, but which can remain permanently open for casual off-meeting chats. IRC is a text only protocol, and a simple client might display part of a conversation as:

```
<John> what's the agenda of today's meeting?
<Mary> http://www.w3.org/2003/05/12-foo-wg-meeting.html
<John> thanks Mary.
```

By itself, IRC already helps teleconferences in several ways: participants can exchange information such as URLs which would otherwise be difficult and error-prone to spell out on the phone, as in the example above. During discussions, parts of proposed text for a specification can also be copied and refined, without having to be read multiple times on the phone. Finally, the scribe can take minutes directly on the channel, which provides a way for other people to follow the discussion as well as giving them an opportunity to correct or complete the notes that were taken.

```
<Al-scribe> John: I see two solutions to this problem
<Al-scribe> John: one is to remove section 1.
<John> I actually said 'section 2', Al.
<Al-scribe> Sorry, thanks for the correction, John.
```

### 3.1.3. The Web

The Web is W3C's means of archiving documents: draft specifications, meeting minutes, miscellaneous notes and memos, etc.

Scribes are responsible for making meeting minutes available on the W3C's site. There are no particular rules that

scribes have to follow, either in the way minutes are synthesized (from literal transcript to short synthesis of the decisions made) or how they are published on the web (possibilities include: sending the IRC text log directly to the mailing list's web archive, or publishing a properly formatted HTML page, with colour codes for speakers, separations of comments made on IRC or on the phone, etc.) This lack of coherence in style was never deemed critical, since a consequence of publishing minutes on the web is that they can be corrected or superseded in place if the group decides to.

The web site can also be used to display and send commands to the bridge, as described below.

## 3.2. Interfaces

Interfaces between the bridge, IRC and the web have been developed in order to make the complete system more efficient than if the components were used separately.

### 3.2.1. IRC/Bridge: Zakim-bot

*Zakim-bot* is a software component that was developed to allow users to send commands to the bridge using IRC. The interface is a *bot*, a software-controlled IRC pseudo-user that responds to specific commands and can write in channels when some events occur. For example:

```
<scribe> John: one solution is to remove section 1.
<Zakim> +Mike
<John> Zakim, who's on the call?
<Zakim> On the phone I see John, Mike, Mary (muted),
        Alexei
<Alexei> Zakim, mute me
<Zakim> Alexei should now be muted
```

*Zakim-bot* warns when a participant connects to the teleconference (second line above) or when one hangs up – callers are recognised through caller-ID lookup. The bot also warns when a teleconference starts. Other features are accessible through commands, as shown on the third line above: people present on the bridge, as well as people whose line is muted.

Teleconference control commands include displaying the list of all the teleconferences scheduled at this time, or associating a meeting to the IRC channel *Zakim-bot* is in (the bot can be in multiple channels for simultaneous teleconferences):

```
<John> Zakim, list teleconferences
<Zakim> I see VB_VBWG(ssml)12:00PM, XML_QueryWG()12:00PM
        active.
<John> Zakim, this is VBWG
<Zakim> ok, John
<John> Zakim, what's the passcode?
<Zakim> the conference code is 8294, John
<Zakim> +John
```

As shown above, *Zakim-bot* also accepts commands to retrieve the pass-code is for a given meeting, and can also be told to disconnect a participant, or to ask who is currently speaking:

```
<John> Zakim, who's speaking?
<Zakim> John, listening for 10 seconds I heard sound from
the following: Mary (10%), Robert (45%)
```

Another functionality that Zakim-bot provides is floor control. Although this feature is not directly connected to bridge functionality, it has proven very useful for palliating the lack of visual information and to prevent speaker collisions and interruptions. The bot maintains a queue of speakers, which people on IRC can add themselves to, or leave:

```
<John> queue+ to ask if we should remove section 1
<Zakim> I see Mary, John on the speaker queue
<Bob-chair> ack John
<Zakim> John, you wanted to ask if we should remove
section 1
<Zakim> I see Mary on the speaker queue
<Al-scribe> John: Yes, I really think we should remove
the section
...
<Bob-chair> queue?
<Zakim> I see Mary on the speaker queue
```

With the meeting chair monitoring the queue by asking Zakim-bot, people who wish to actively participate to the meeting can do so without having to interrupt the current speaker.

Yet another functionality is agenda management. Similarly to the speaker queue, Zakim maintains a list of agenda topics which the chair can go through, possibly adding or removing items:

```
<John> agenda+ recommending changing section 3
* Zakim notes agendum 5 added
<John> agenda?
* Zakim sees 5 items remaining on the agenda:
* Zakim 4. solving issue #42
* Zakim 5. recommending changing section 3 [from John]
...
<Bob-chair> Zakim, take up agendum 5
<Zakim> agendum 5. "recommending changing section 3"
taken [from John]
```

Again this functionality makes it easier to track what is happening on the teleconference and helps produce accurate meeting minutes.

Finally, Zakim-bot accepts commands from privileged users to set up ad-hoc teleconferences which are not in the normal schedule. If enough resources are available on the bridge at the requested time and for the requested number of participants, the bot will schedule the requested teleconference.

### 3.2.2. IRC/Web: RRSAgent

*RRSagent* is another IRC bot that automatically publishes the log of the current channel on the web, updating the page regularly. It chooses the URL automatically according to a given scheme, and also lets the participants set the access controls of the page (some W3C meeting records are member-restricted or staff-restricted). For instance:

```
<John> RRSAgent, record
* RRSAgent is logging to
http://www.w3.org/2004/03/18-vbwg-irc
...
<John> RRSAgent, please make log member-visible
<RRSAgent> I have made the request, John
```

### 3.2.3. Bridge/Web Interface

IRC is a linear medium and the only way for a participant to know information such as who is on the call, is to ask Zakim-bot explicitly. If many people on the call request the information at different times, this can add a lot to the IRC channel, which can be detrimental to the readability of the meeting record. Therefore a dynamic web page is more adapted to this task, as it can be accessed individually<sup>2</sup>. Such a page exists on the W3C's site and displays the names of participants to all the running teleconferences in real time (Fig. 1).

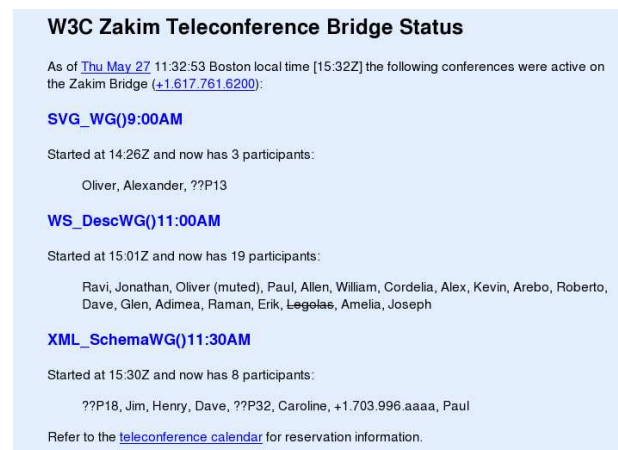


Fig. 1. Screenshot of Zakim's Web monitor

Another web form also provides a way to have the bridge dial out to a number among a list. One could imagine the speaking queue and agenda be displayed similarly on the web, but this has not been implemented so far.

### 3.2.4. Phone/Bridge interface

In situations where the user cannot connect to IRC or to the web (e.g. if the user is attending the teleconference with their mobile phone), the Zakim interface can also recognise DTMF sequences as commands: 61# is used for muting, 60# for unmuting, one can add themselves to the speaker queue by dialing 41#, or be removed with 40#. The result of these commands are reflected on IRC:

```
...
* Zakim hears Susan's hand up
...
```

<sup>2</sup>Note that commands can also be sent to Zakim-bot in a private IRC chat, avoiding interference with the group's channel.

## 4. RESULTS

### 4.1. How meeting efficiency has increased

As the functionality presented above shows, many casual teleconference operations that would have had to be managed by hand are now handled by the bridge and its interfaces. In fact, meeting administration is now almost entirely detached from the contents of the meeting. All the participants (not only the operator, as was originally available) can now enquire and operate bridge controls without interfering with the teleconference. This has dramatically reduced the time some parts of the meetings previously took, such as the roll call formerly carried through by the chair at the beginning of each meeting: now the list is provided automatically, and thus avoids the lengthy process of asking who is on the call. Another formerly inconvenience that Zakim has remedied is line noise: often some connections are burdensome, either because the participant is in a noisy environment, or because the line itself produces static or echoes. The 'offending' participant or line are sometimes hard to determine, as people are not always aware of the effect problems on their side cause. Zakim makes it easier to detect noisy lines through the "who is speaking?" command and allows other users to mute them.

The queue management mechanism has also improved efficiency, not only because it provides equal opportunity for participants to express themselves on the teleconference, but also because time is no longer lost by interruptions on the call by people notifying that they would like to say something.

RRSagent provides automatic publishing of the IRC log of the meeting, while the agenda feature of Zakim-bot adds information to the log about the progression of the meeting. As is outlined above, IRC discussions has two main benefits. The first is that minutes can be taken collaboratively, people other than the scribe can complete the record, or take over when the scribe themselves is speaking. This usually greatly improves the quality of meeting records, and this use of IRC has thus been successfully generalised to face-to-face meetings. The other benefit is the accessibility IRC provides: because notes are taken in near real-time, people who have trouble listening to the teleconference, often because they are not native English speakers, can follow the meeting much more easily.

### 4.2. Problems and Improvements

Although Zakim has made working group meetings easier, some problems remain. This section summarises them, as well as lists planned improvements.

#### 4.2.1. Shortcomings and limitations

The first category of limitations, making it difficult to make the meetings as efficient and comfortable as participants would hope, comes from the phone system itself, as well as the teleconference bridge hardware and interface. Line noise originates independently of the teleconference bridge, and while one could envisage noise recognition and cancelling algorithms applied within the bridge, it is a difficult system to implement and necessitates dedicated hardware to process the signal in real-time.

Similarly, DTMF tones can be heard on the bridge, and since some telephones send them at a somewhat high volume, they can end up being a nuisance if too many people use them. This is why the set of DTMF commands has been deliberately kept small. However DTMF tone cancelling could also be achieved with techniques similar as above. The bridge vendor has announced they will work on an upgrade to the bridge's firmware that would mute all DTMF from the teleconferences. This would then allow the implementation of more DTMF commands without the current drawback.

Another issue originating either from the phone system or privacy concerns is with the ANI: some calls, often going through relays in different countries do not provide phone number information. Or, it can happen that the the participant is not calling from a registered number. Another problematic situation is when callers are behind a corporate switchboard (PBX). It then becomes difficult to know who is on the call, and also to map the list of people on the bridge to the list of people on IRC. In those cases, it has to be done manually:

```
<Zakim> +??P14
<John> Zakim, P14 is me
<Zakim> +John, got it
```

Zakim also has a DTMF sequence permitting callers to provide a personal identifier: callers can dial a personal code which the bridge can detect to associate that caller's Zakim identifier to the phone line they are dialing from.

A limitation, this time coming from IRC as well as the web, is the lack of real-time monitoring of the lines, in order to detect who is speaking or which line is making noise. While the bridge's original interface shows the information in real time, Zakim-bot only displays it after listening to the line a few seconds. It is imaginable that Zakim-bot would just dump the information on IRC as fast as it samples the line noise, but it would quickly drown the IRC channel with hundreds of lines of text. This necessary sampling over a few seconds has the consequence that it is hard to identify a speaker if more than one person has been speaking, or to identify a noisy line if a buzz is intermittent.

Yet another category of problem that occurs with this setup originates from the participants: discipline, for one,

becomes more critical than usual, as people are expected to respect the speaker order. The chair, in particular, has the additional duty of keeping the speaker queue managed by Zakim-bot synchronous with the way the actual meeting is happening. People who for some reason cannot use IRC (corporate firewalls being the most common cause) might also feel frustrated as some aspects of how the meeting is run, visible on IRC, could escape them.

Moreover there is often a tendency to see the meeting discussions split: conversations arising from a side comment that someone typed on IRC can diverge from the main telephone conversation, leading to confusion about what to record of the meeting, and in what order. Here also, this behaviour can be a concern if not all participants use IRC.

Reciprocally it has happened that people attended a meeting only through IRC. This has proven to be quite inconvenient as, even though they could follow the meeting, the time delay between phone conversation and their record on IRC made it very difficult for them to react to a statement, as the meeting may have moved on quickly.

#### 4.2.2. Improvements

Other technical shortcomings of Zakim are presumably easier to address and a few of them are under study either by the maintainers of the system or by individual working groups.

The linear nature of IRC makes it difficult to perform some tasks that are idiosyncratic to the activity of a working group. As mentioned above, portions of specifications being discussed can be copied on the channel so that the participants can review and correct them. However those sections must remain short, and IRC does not provide a view of the effect of the changes in the whole specification – in particular people are not necessarily up to date with the location of the text in the document. One way to palliate this deficiency is to have the document's editor amend the text as decided and, as often as possible, publish it on the web during the meeting. While W3C provides this facility and makes it simple, it suffers from delays inherent to the W3C web site, mostly caused by mirroring the site. Using a real-time CSCW system would possibly overcome this limitation as well as allow simultaneous editing of the document. An example of such a system is SubEthaEdit [4], a distributed document editing tool, which would seem quite valuable to the type of work done at W3C. Nevertheless it remains to be tested if such systems are able to scale to up to 50 participants.

A similar well-known problem is that of the distributed white-board, which many research project and commercial products already address. Unfortunately the lack of a common and open protocol, resulting in too few cross-platform implementations has prevented its use so far. The scalability concern mentioned above applies here as well.

Video conferencing has also been investigated, without a definite outcome yet: technical difficulties (choice of protocol, bandwidth required) compared to the apparent lack of obvious benefits has made it a secondary improvement. However, ongoing testing is being performed and may lead to production use for some meetings in the future.

Recording the audio channel of the meetings is also not considered critical, given that the burden of storing and accessing the recordings as well as annotating the sound segments and correlating them with the written record does not appear very advantageous over the current architecture. However, ongoing research such as that carried out by EU project AMI [5] could provide interesting solutions.

Voice recognition for bridge commands seems a facility that, even if it appears feasible, would probably not provide many advantages over the existing system. While accessibility would be increased for participants not on IRC or who cannot use DTMF, it might add to the confusion as bridge commands are intermixed with meeting discussions. Voice recognition for automatically recording the meeting minutes is a much farther goal, especially because of the unreliable voice quality of some callers, as well as the necessity to synthesize minutes, as opposed to the limited usefulness of a literal transcript.

## 5. CONCLUSION

This paper introduced the Zakim teleconference bridge used at the World Wide Web Consortium. While academic research describes much more advanced multimodal meeting systems, the requirement that the system be robust as well as accessible to the greatest possible number of participants, has made the feature set seemingly limited, compared to today's advanced experiments. However it has radically changed the way W3C holds meetings, improving time efficiency as well as accessibility for most participants, and provides valuable insight into large-scale augmented distributed meetings.

## 6. ACKNOWLEDGEMENTS

Work on this article was supported by EU FP6 projects MWeb and AMI. Development of Zakim-bot was supported in part by funding from US Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0593, "Semantic Web Development". The views are those of the authors, and do not represent the views of the funding agencies.

The author would like to acknowledge Ralph Swick, of W3C, for developing the software presented in this paper. RRSAgent is based on an irc logger tool written by Dave

Beckett of Institute for Learning, Research and Technology,  
UK.

## 7. REFERENCES

- [1] World Wide Web Consortium  
(<http://www.w3.org/Consortium/>)
- [2] Compunetix Contex<sup>®</sup> teleconference system  
(<http://www.compunetix.com/ix/csd/prod/index-sp.html>)
- [3] J. Oikarinen, D. Reed. *Internet Relay Chat Protocol*. IETF RFC1459. 1993 (<ftp://ftp.rfc-editor.org/in-notes/rfc1459.txt>)
- [4] SubEthaEdit - collaborative editor  
(<http://www.codingmonkeys.de/subethaedit/>)
- [5] Augmented Multi-party Interaction - EU IST FP6 Project - (<http://www.amiproject.org/>)