# V-CHAT (VERSATILE CHAT)

MARIA CECILIA TEIXEIRA[1]; DAVIDE VACCARI[2];
MARCO SANGALLI[3]; JUERGEN SCHWIETERING[4]

[1]Researcher on Department of Informatics and Communications
Escola Superior de Tecnologia e Gestão (Instituto Politécnico de Bragança),
Qta Sta Apolónia 5300 Bragança, Portugal
mct@ipb.pt

[2]Web-designer, Project Manager Via Ravasi 17, 21100 Varese (VA), Italy
davide@flashpushers.com

[3]Programmer, Software Engeneer Via Eupilio 12, 23842 Bosisio Parini (LC), Italy
mako@flashpushers.com

[4]Senior Software Engineer Avd. J. Orbacieta 13, 30370 Cabo de Palos (Murcia), Spain
juergen@schwietering.com

## INTRODUCTION

We focus our research on text based messaging systems and the impact they have when they are implemented in a work structure and how this kind of software can improve workflow and interaction between firms and clients both in B2B and B2C relations.

In recent years many online text chat systems have been developed and use. Internet Relay Chat or IRC, ICQ, AOL and MSN are some examples of the most popular tools for real-time communications via the Internet and have been used both for doing serious work [4], decision making [5] and for entertainment purpose. Messaging programs tend to imitate each other and they imitate other chat programs either. Each new release simply adds new features that keep the use of this software nothing but a cheaper alternative to phone. Nowadays Instant Messaging, chat system and video conference software must evolve and specialize into something more flexible, sophisticated and secure.

*Organizations that find strategic value in instant messaging should be deploying secure auditable enterprise instant messaging.* [6] Our messaging model is built upon generalizations of the major Internet messaging systems. In asynchronous communication mode minutes or hours may pass between conversational turns. On the other hand, users exchange information relatively rapidly in synchronous communication (see Figure 1). Different models of messaging will be evolving toward an integrated communications model in which people (or applications) choose the best way to communicate.

Increasingly, forms of messaging that are closer to synchronous, “real-time” communication are being brought into the workplace. Previous research [1, 2, 3] has shown that synchronous messaging in the workplace has a number of uses, including opportunistic interactions, broadcasting of information or questions, and a “signaling” function in which people negotiate availability for other interactions. We consider how online messaging and chat can be improved for professional use such as online helpdesk and e-learning. We also considered how the messaging and chat experience can be useful for data marketing analysis and trends analysis. In this context, one of the aspects we investigate was the limits of the protocols used from chat and messaging systems inside Intranets protected by Firewalls.

**OUR GOALS:**

1) Create a powerful core that can lead behind different kinds of chat systems like professional or entertainment targets.

2) Provide analysis tools that extract data from the chat logs using database queries.

3) Provide so called 'nani bots' that perform particular skills such as translation, automatic online assistance like research in phone books, calculators, providing news and many more highly specialized 'users'. This concept is easily integrated as they are external instances providing a standardized interface to the facilities (users do not need to learn how to use many different programs).

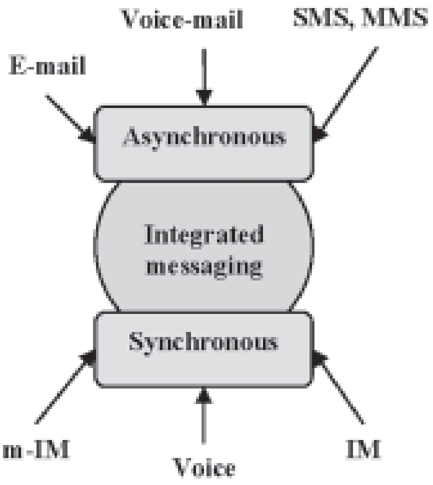
4) "Nani bots" could theoretically serve as a gateway for any service. Also combining different Messenger systems and voice over IP etc. These standalone units can easily be integrated in ICQ, MSN etc.

5) Integration with mobile communication (still in progress).

6) To combine in one tool messaging, chat, and storage of information.

7) Persuade people to have one unique ID which can show different faces to the public. Using the SecureID research [7, 8] to make sensible data hidden.

8) Making each procedure as easy as possible with a fast and understandable access, permitting users to evolve their digital personality (or multiple personalities) according with their expertise with the system.



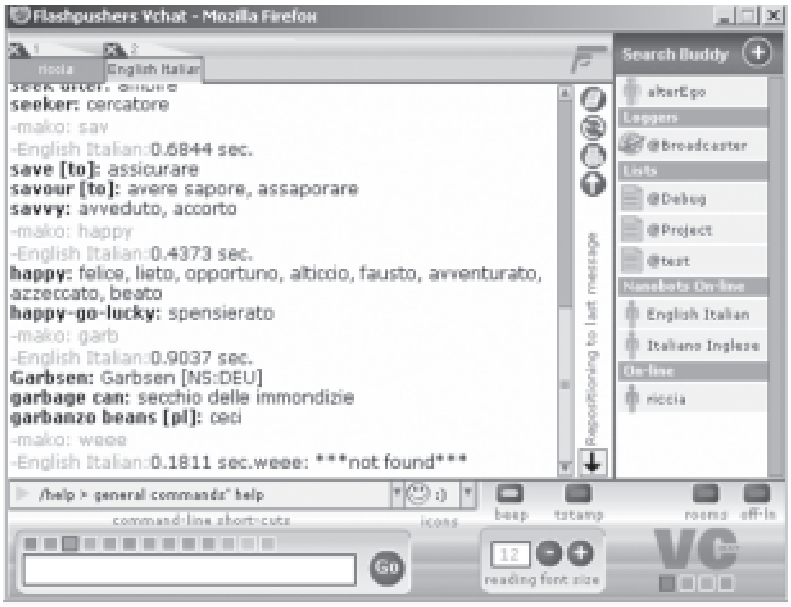
## FIGURE 1. INTEGRATED MESSAGING

### IMPLEMENTATION METHODOLOGY

In the developing process of V-Chat our aim is to overwork the chat experience from the scratch trying to understand limits and problems of the actual available systems and create a complete tool for synchronous and asynchronous communications (see figure 2). We didn't focus only in the instant messaging but we try to develop an integrated and scalable tool able to deal with work-related procedures provide users in one easy understandable layout both a chat/messaging system, a document sharing tool and a platform for third parts developers plug-in. We started to implement the most useful tools IM and chat systems already integrated and we consider to avoid the typical socket connection on UDP/TCP IP protocol and theirs own problems in relation with professional Intranet and Extranet.

We use instead a standard socket connection via HTTP/HTTPS on open source webserver (Apache) to interact between system and users. All the messaging data (as well as the users details) are stored in a open source database (MySQL) and the files shared among users are stored on a secure folder on the Server. We use an open source server side scripting technology (PHP) to store and retrieval data values to the database and perform interface operation between client and server. Users authenticate themselves on our V-Chat trough a HTML login pages; once allowed they must open our client that runs on the free Macromedia Flash Player. All the data exchange among client and server are XML driven with UTF-8 encoding to grant multiple character set compatibility.

**FIGURE 2. V-CHAT INTERFACE**



```
seeker: cercatore
-mako: sav
-English Italian:0.6844 sec.
save [to]: assicurare
savour [to]: avere sapore, assaporare
savvy: avveduto, accorto
-mako: happy
-English Italian:0.4373 sec.
happy: felice, lieto, opportuno, alticcio, fausto, avventurato,
azzeccato, beato
happy-go-lucky: spensierato
-mako: garb
-English Italian:0.9037 sec.
Garbsen: Garbsen [NS:DEU]
garbage can: secchio delle immondizie
garbanzo beans [pl]: ceci
-mako: weee
-English Italian:0.1811 sec.weee: ***not found***
```

The use of XML permit third parts to manipulate incoming and outcoming data to develop custom plug-ins. Some plug-ins, such as word translation, calculator and message history retrieval, has been developed as example, see Figure 3.

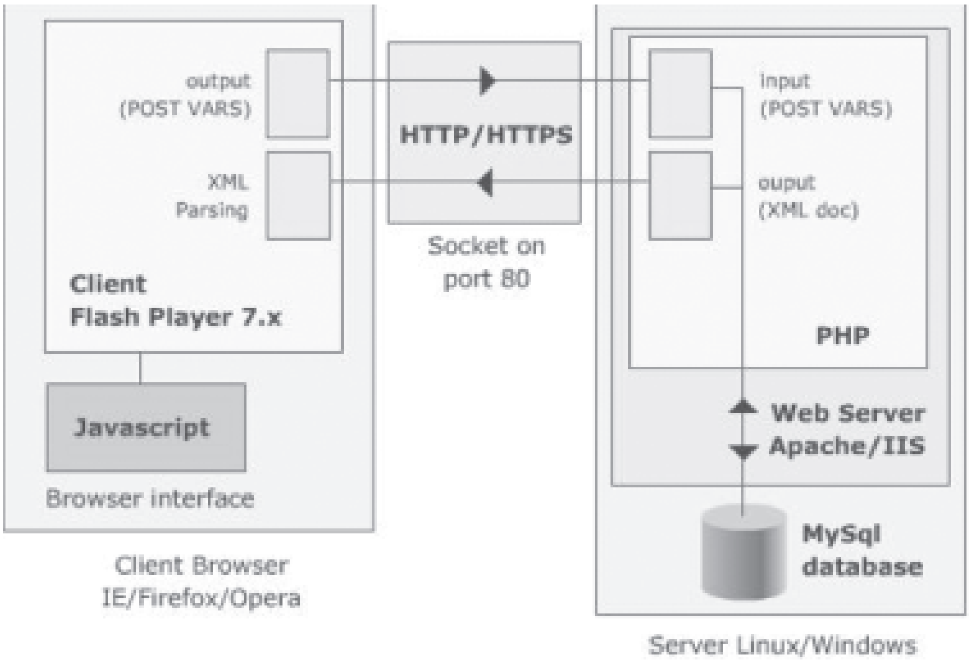


**FIGURE 3: CLIENT SERVER DATA STREAMING**

Developers could implement AI interfaces using AIML scripting language to create Nani bots that can interacts with the database trough XML.

## MATERIALS AND METHODS

**• Requirements**

PROCESSOR: intel or amd processor >500mhz

RAM: 128 mb ram

HDD: 10Gb hdd

OS: linux or windows server

WEB SERVER: apache or IIS web browser

DATABASE: MySql database

HYPER TEXT PROCESSOR: php 4.3 with recommended settings (session enabled)

SECURE SOCKET LAYER: OpenSSL or others

**• Administration Tools**

Allow to configure server load and clients' performance impact:

1. minimum client's read polling time
2. client's write queue output write time (time between 2 write operations)
3. maximum polling time
4. client's color table (font color) Allow to create new rooms and loggers by a simple interface

1. room name
2. group allowed to speak Allow to create a own front end with replacing graphic resources

Up to this moment our research already gave enough results to permit us to develop a beta version with many features already available:

Server side:

- First Linux/Windows server stable version running with Apache, MySQL, PHP
- PHP scripting interface with Macromedia Flash 7x clients
- Log-in template with different languages settings (UTF8)
- Database structure for messaging support
- Database dictionary for Translation Plugins (demo)

Client side:

- Client Windows Macromedia Flash 7.x stable version
- Adaptive polling messages database
- Queuing management database writing procedure

- Buddies and rooms status
- Command line support

## DISCUSSION

The project was born to create a secure and smart system that brings the most powerful part of commercial messaging systems and the most advanced secure technique to create a smart and usable product without worldwide usable (UTF support).

Server side is based on a database that allows to create a non a volatile message interchange storage system, it means that all the messages that are passing trough are not loss and they are immediately ready to be used. This is the most important goal that allows large corporate and big firms to be ready with newest law restriction on document interchanges, it means that all data are stored and at any time they can be read as the law require.

Server side uses robust open source components such as Apache, PHP, MySql and OpenSSL. All those software had been worldwide tested and they are the most robust and cheap software available for the industry.

The other software required for our system is the client one. We choose it 'cause it works on the most used and well known operative system, like Windows, Linux, Apple, and in conjunction with Flash Player, is the most used plug-in on the net freely distributed by Macromedia.

Flash client has been choose for its easy installation and configuration; that means that no know-how is required to start our client and the capacity to keep the same layout and interaction with different platforms.

The project has been developed with an eye on third part developer, **V-Chat** uses a well know data format to interchange its data. Developers could create custom plug-ins solution on server with an instant feedback from the client side without reprogramming it.

The system can be reassumed as a secure way to interchange simple messages data or complex ones without compromise corporate network's security (no custom socket port needed). It works on open source solid basement and allows programmers to develop custom tools.

## REFERENCES

[1] Nardi, B., Whittaker, S., Bradner, E. Interaction and Outeraction: Instant Messaging in Action. C*SCW 2000*, 79-88, 2000.

[2] Bradner, E., Kellogg, W.A., & Erickson, T. The adoption and use of 'Babble': A field study of chat in the workplace, *ECSCW*, Copenhagen, September 12-16, 139-158, 1999.

[3] Isaacs, E., Walendowski, A., & Ranganthan, D. Hubbub: A sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. In *CHI'02*, Minneapolis, MN, 179-186, 2001.

[4] S. Farnham, H. Chesley, D. McGhee, R. Kawal, and J. Landau. Structured online interactions: improving the decision-making of small discussion groups. In *Proc. of the ACM 2000 Conference on Computer-Supported Cooperative Work*, pages 299–308, Philadelphia, United States, 2000.

[5] J. Herbsleb, D. Atkins, D. Boyer, M. Handel, and T. Finholt. Introducing Instant Messaging and Chat in the Workplace. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 171–178, Minneapolis, USA, 2002.

[6] Daniel W. Rasmus. Evolving Toward Enterprise Instant Messaging. 2002 Giga Information Group, Inc. May 6, 2002

[7] Danah Boyd, Nelson Lai, "SecureId: Security of Identity Through Knowledge", MIT Media Lab, 2001-2002.

[8] Danah Boyd"Faceted Id/entity: Managing Representation in a Digital World." Cambridge, MA: MIT Master's Thesis. August 9, 2002.