

## Rencontre: a Rebol Agent-Oriented Middleware

François Jouen

CHArt Cognitions Humaine et Artificielle (Human and Artificial Cognitions Lab)

EPHE/Paris 8

41 rue Gay-Lussac

F-75005, Paris

In many of experiments done in the lab we have to record sensors that generate electrical signals to measure biophysical phenomena, such as temperature, force or brain electrophysiological activity. Some commonly used sensors are strain gauges, thermocouples, thermistors, angular encoders, linear encoders, electrodes ... To measure signals from these various transducers, we have first to convert them into a form that an A/N device can accept for recording. But we also have to analyze « on line » the sampled data in order to modify different parameters of the subject's environment according to the behavioral reactions that are analyzed by the recording of sensors. This implies the use of a multitasking development. Unfortunately Rebol doesn't offer such opportunities for multitasking experiments. This is the reason why we have developed **Rencontre**. **Rencontre** is a middleware elaborated for the collaboration between interactive applications developed with different languages and toolkits and running on different operating systems. Since Rebol is a messaging language using internet protocols it was very easy to conceive **Rencontre** as a model of communication between active "agents" which are dynamically stored in a list: agents connect to the bus, send messages and leave the bus without interfering with the other agents. Another characteristic of this middleware is the very simple exchange data protocol between agents we adopted. Each agent, whatever the operating system or the language used by the developer, must be able to send and receive a text message (a string of chars) including only information about the name of the agent and the content of the message organized as follows "Application Name: Message". Moreover each agent is responsible of the data treatment sent by the others agents. If the agent has to process the messages sent by another agent, works is done locally. On the other case, the message is purely ignored.

Whatever the language used, the functioning process is similar. A supervisor creates the bus by opening a TCPIP port. Each agent must first verify that the bus is created and then connect to the bus. In other words agents give appointments to each other on the bus. When connected agents can send messages that are collected and then broadcasted by the supervisor to each connected agent. When receiving a message, each agent can locally process or ignore the process.

In the Devcon 07, principle and implementation of Rencontre will be illustrated by a real time data acquisition software.