

```
+-----+
| Google/PSU SoC '09 Project Proposal |
| NAME:          VICTOR IVRI          |
| CONTACT:       victor.ivri@gmail.com |
+-----+
```

```
+-----+
| TITLE:         HAMMOCK              |
+-----+
```

HAMMOCK : Human-AI Multi-Modal Online Communication Kettle. (Brew it yourself!)

```
+-----+
| SYNOPSIS:                                             |
+-----+
```

The idea is to create a flexible framework for multi-modal communication between people using browsers and AI bots sitting on servers. In this context, an AI bot is a rather abstract entity. Basically, it is some logic, perhaps connected to a DB or even other servers, that can intelligibly communicate with users towards some end result. This project aims to considerably ease the development of creative online human-computer interaction systems, with no special preference for any particular modality of communication (e.g. text, game-playing, drawing, etc). The framework will strive to be flexible and modular, to allow maximum freedom for developers in both the client and server-side logic, while retaining a common engine and a global architecture that is optimized for this kind of system. The power of this approach is that it will allow people to develop and share libraries, and quickly develop applications in this particular sphere, without having to repeat all the basic steps.

```
+-----+
| BENEFITS TO COMMUNITY:                             |
+-----+
```

Currently, projects such as PandoraBots and its derivatives (AIML, A.L.I.C.E., etc.) use a restrictive text-based interface, and a pre-defined bot-logic to allow for web-based human-AI interaction. I feel that the potential and the need exist for a much broader type of browser-based communication between humans and AI bots, and many developers would benefit from having such a framework available to them, each for their own particular usage. I haven't decided which of GPL or BSD would be more appropriate for this project, so for now I'll leave it as an open question. However, I'm comfortable following the PSU team's advice on this, whatever it will be.

```
+-----+
| DELIVERABLES:                                       |
+-----+
```

I intend to produce a first release of HAMMOCK by the end of the project, as well as create a several useable libraries and a proof-of-concept prototype with simple multi-modal communication capability. The first deliverable is a 'hard' commitment, while the second is a softer one, which can shrink or expand as necessary. Overall,

I feel that this is a both attainable and non-trivial deliverable.

TIMELINE:

+-----+
NOW -- JUNE 2 ==> Investigation into technology and methodology, and
prototype building.

The deliverable consists of bridging all knowledge-gaps, as well as constructing a
prototype/working-skeleton of the architecture, with a minimal set of features.

+-----+
JUNE 3 -- JULY 10 ==> Programming. The basic framework will be in working
condition, in time for midterm evaluation.

The deliverable consists of having the important features of the architecture (see
'Description' below) operational.

I estimate spending 3 weeks for the core engine, 1 week for supporting libraries
and utilities, and 0.5 weeks for creating a basic console real-time command
interface. In total, 4 weeks of development, including on-the-fly debugging, and a
1.5 week buffer for unexpected delays.

+-----+
JULY 11 -- JULY 21 ==> Advanced features of the framework will be implemented,
and everything will be in bug-free condition.

The deliverable consists of integrating features such as control over number of
browsers per bot, (original design will be one-bot-one-browser), enhanced real-time
control from console, a diverse messaging library, and general code-polish.

+-----+
JULY 22 -- AUGUST 10 ==> Creating a proof-of-concept system, that uses HAMMOCK
to quickly prototype a useable client-server AI application.

This could be a descendant of the original ELIZA, which provides users with
"therapy" of a non-directive flavor. The basic interface would be text-based, but
there would also be a canvas (static or animation), where the "therapist's" facial
expressions would change according to the conversational context. I will try to
package as much as possible into libraries, as well as utilize external, open-
source libraries.

+-----+
AUGUST 11 -- AUGUST 17 ==> Documentation week. Polishing the Javadocs, creating a
developer's manual to HAMMOCK, and documenting libraries.

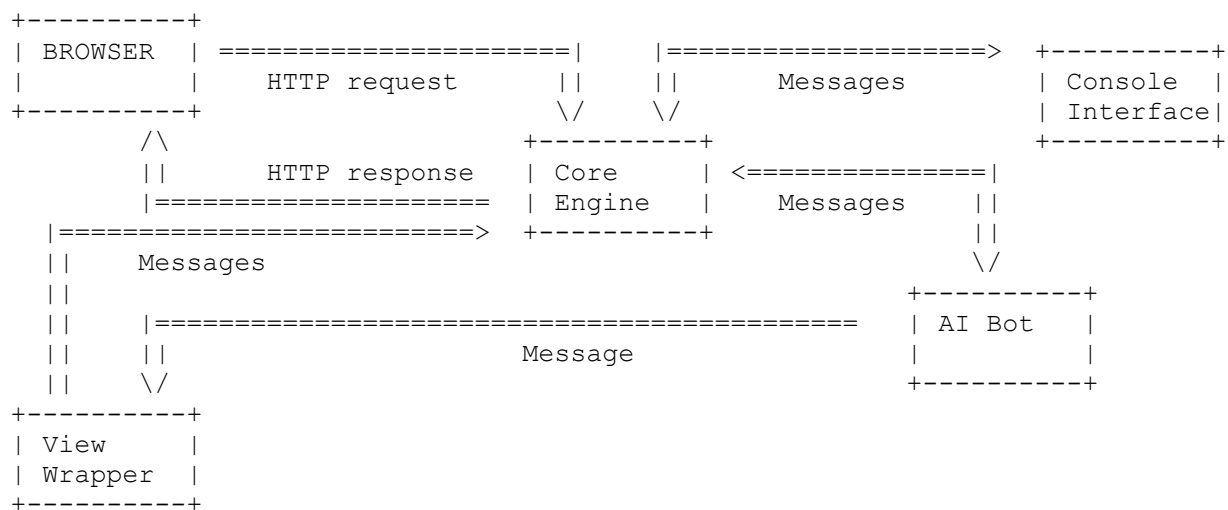
+-----+
FIN. Spend next few years building cool AI bots that communicate with people in
different ways, as well as help others to develop for HAMMOCK.

+-----+
| DESCRIPTION: |
+-----+

The project would be implemented using Java and Tomcat client-server technologies. An MVC architecture will be roughly followed. The controller (core engine) will be implemented as part of HAMMOCK, while leaving the model (AI-logic) and view (Java-logic wrapper, inside of which JSP + HTML + JavaScript pages will exist.) as abstract classes. Thus, the global logic would be set, while leaving important implementations to developers. The bot and view will run as separate threads for increased flexibility and performance, and communicate with each other via a messaging service. To leave most power with the developers, I intend to 'cut the middleman', and have the model send messages directly to the view, instead of the traditional approach where the controller has to be in the middle. Additionally, I think it would be a sound design decision if the View nothing of the communication with the browser, and use the Engine to forward all HTTP responses to it. In a sense, this can be interpreted that the Model encapsulates the view, but it seems the best decision in the circumstances. The system should work in two regimes: both as a one-bot-per-browser, and as a one-bot-many-browsers, with tweaking capacities (launch a new bot per 20 browsers, limit the system to 100 bots, etc.) Additionally, the developer/maintainer of the system would always remain in control, through a real-time console interface.

The intent is for libraries to grow around the core framework, so that in some time developers could easily pick-and-choose from them to customize this framework for their own needs. I intend to start developing the messaging library, to support some variety of modes of communication, such as canvas drawing, text-based, radio buttons, etc.

PROPOSED ARCHITECTURE:



```

+-----+
| RELATED WORK and SHORT BIO: |
+-----+

```

Here's a bit about my background, to help you decide whether or not I'm qualified for this kind of project. Currently, I'm an undergraduate student at York University in Toronto, ON, double-majoring in the cognitive and computer sciences. I consider myself a good Java programmer, a mediocre C/C++ programmer (currently

employed as a p/t C++ junior dev.), and I know bits and pieces from other languages (such as MIPS assembly, Lisp, Prolog). I have also completed many courses in the social sciences in psychology, linguistics and philosophy as part of my program. I've created an advanced AI client for the game of RISK for last year's GSoC ([1] is a link to the public GIT repository), as well as several genetic algorithms and artificial-life simulations simply for fun ([2] is a link to my private GIT repo).

[1] <http://git.thousandparsec.net/gitweb/gitweb.cgi?p=gencon-rfts-ai.git>

[2] git://github.com/vivri/small_projects.git