

# Lilypad Net: Agents Involving Users in the Future of The Internet

Ben Coburn  
Department of Computer Science  
University of the Pacific  
btcoburn@silicodon.net

## Abstract

Lilypad Net (LPnet) is a distributed, platform neutral, mobile agent system. LPnet's main goal is to promote end-user programming for networked environments. To reduce the knowledge and resource thresholds for writing mobile agents LPnet uses ubiquitous technologies and an abstraction layer to simplify basic agent programming. LPnet supports research and other advanced programming using Python as an agent programming language.

## 1. Introduction

The Lilypad Net (LPnet) project was inspired by the idea of producing fragments of software that could wander across the network and interact with people they encounter. LPnet is also an attempt to use software proxies to recreate the kinds of chance meetings and interactions that people experience in a large city. These ideas could also be attributed to Tron (the movie) and nonlinear programs like BottleMail, or a mild-mannered reinterpretation of computer viruses and worms [LIS82] [BMJ03].

LPnet seeks to “humanize” the network, inspire future programmers, and become a ubiquitous end user application. Therefore the LPnet system must be platform neutral, provide a “sand-box” for running software agents, be relatively light-weight, consume few & readily available resources, and be accessible to “competent” end-users. A “competent” end-user is one who can and may already have taught themselves how to write basic HTML by hand. Due to these requirements, LPnet is being developed in Java with scripting languages such as Jython embedded for writing agents [JPY03]. The LPnet system also uses standard data formats and networking protocols such as XML and HTTP. These requirements do not impose problematic restrictions while making it much simpler for other systems to integrate with LPnet and its agents.

## 2. Analogy and Concept

The naming convention and analogy for this system were chosen to emphasize the mobile and independent nature of these agents. In comparison, Java applets are constrained to waiting for users to load the web page to which they are attached. LPnet agents on the other hand actively travel between computers of their own volition.

The analogy that naturally presents itself is that of frogs hopping around the lilypads of a pond. In this analogy the lilypads are individual computers running the LPnet system. Agents are described as Frogs and Toads, where Frogs are simpler and have fewer requirements. On a larger scale, Ponds and “backwaters” (collections of ponds) are used to organize related sets of individual computers (or lilypads).

In this context individual LPnet hosts (or lilypads) are considered to have a one-to-one relationship with actual people. However this can be generalized to many-to-many relationships without disrupting the analogy or the technology involved. Each instance of the LPnet program behaves independently, so the collection of agents that reside on your work computer may be different from the agents on your home computer.

## 3. Ponds

Although individual lilypads interact in a peer-to-peer manner they still need a way to locate each other. This is the function of ponds. Each pond provides the lilypad with a set of references to lilypads that are in (or subscribed to) that pond, as well as references to related ponds.

Each pond typically represents a particular topic or group of people. These ponds are then integrated into a pseudo-hierarchy by the links between them. These links form a directed graph of nodes reachable from the pond at which the lilypad started requesting information. In Fig. 1 ponds A-E are reachable when starting at pond A. The “parent” ponds z, y, and x are

unreachable from A because neither A nor any pond reachable from A references them. This allows for pseudo-private ponds like z that still connect to the rest of the network. Because ponds are not required to know who links to them, each pond only maintains local data. By maintaining only local information the complexity of individual ponds is greatly reduced.

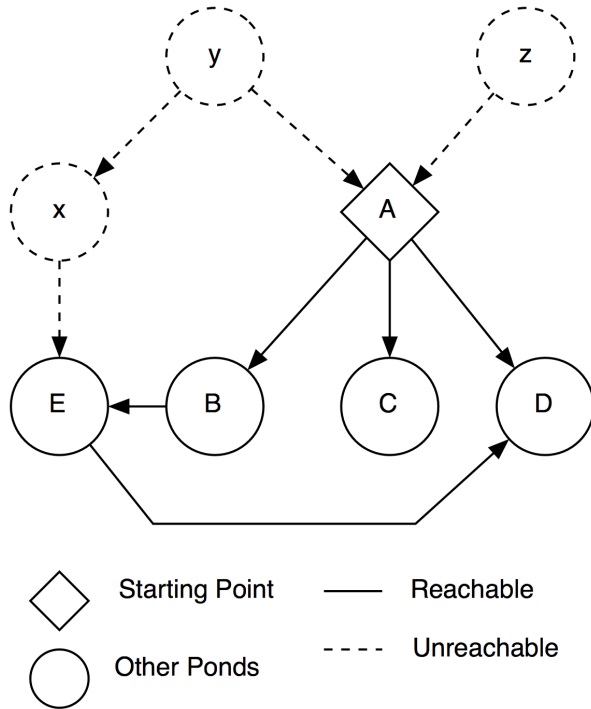


FIGURE 1: A Sample "tree" of LPnet Ponds

The ponds themselves do not participate directly in the Lilypad network. Instead ponds act as a directory for locating other computers running the LPnet application. When each LPnet application starts, it registers its location with the ponds in which the user wants to participate. As an agent finishes running on a host, the system prepares to send it to another LPnet system. This system is randomly chosen from the ponds that the user is registered with and any of the ponds that they reference. Individual ponds may place restrictions on the agents that are allowed to enter them. Individual agents may also request to be sent to a specific LPnet host system.

The simplicity of ponds does not require a dedicated "server" to implement. In fact, the reference implementation for ponds will use a PHP or Perl script that runs inside a normal web server like Apache [PHP03] [PRL03] [APA03]. This is also the way that most users are expected to run their ponds. Anyone who has access to a web server will be able to run a

pond simply by copying the pond script and using their web browser to configure the new pond.

This should make the resource threshold for running a LPnet pond/server significantly less than that of other multi-agent systems. One also does not need to convince the system administrator to allow a "strange" server on their system.

#### 4. LPnet Agents: Frogs & Toads

LPnet agents are logically composed of three sections: the agent information and configuration, the agent code, and the Lily File System (LilyFS) data store. The agent information describes the agent and its author. It is accessible to the user on whose system the agent is running. The agent configuration information is used by the LPnet system to determine the large scale dynamics of the agent's behavior. This includes things such as how many times the agent will run on each computer and which ponds it is allowed to hop to. The LilyFS provides a virtual pseudo-file system where the agent can store data that it wants to take with it from system to system or preserve between runs. Agents can also use the LilyFS to store and organize arbitrary information. Each agent uses its own private LilyFS "partition". These three parts are common among all agents.

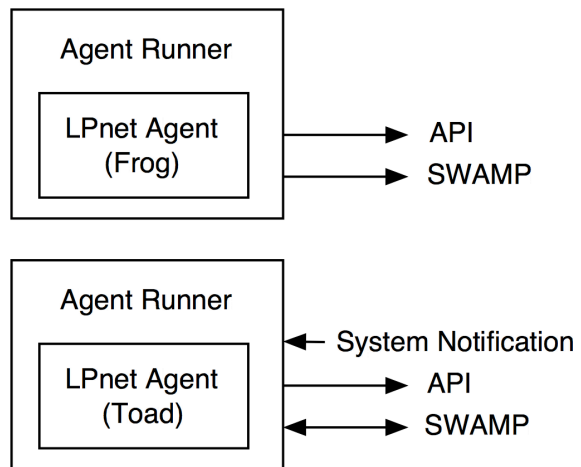


FIGURE 2: Agent Types & System Interactions

The differences that distinguish Frogs and Toads can be found in the requirements for the code section and the way that each agent interacts with the rest of the LPnet system. As you can see from Fig. 2, Toads are separated from Frogs by their ability to handle requests from the system after they have been started. When a Frog is run the LPnet system executes the Frog's code

in a linear manner, as with a script. This code may call methods in the API provided by the LPnet system to simplify certain tasks like using the GUI. For example, the API can be used to facilitate asking the user a question or displaying information. The code may also send Simple light-Weight Agent Messaging Protocol (SWAMP) messages, though the Frog cannot receive messages.

The SWAMP protocol is a light-weight communication protocol used by LPnet systems and their agents to exchange information. The SWAMP protocol itself is an HTTP compliant extension of the HTTP protocol. It uses extra required headers to specify LPnet specific addressing information. Messages are delivered with confirmation that the other lilypad has received them and are placed in the appropriate mailbox/queue. For an agent to send a reply, a new message must be created. This allows agents to exchange messages without inherently blocking on replies from each other. Agents may be written so that they wait for a reply message, though this should be done carefully with a timeout to guarantee that the agent does not wait indefinitely.

Toads are markedly different from Frogs in that they are required to implement a set of methods that will be called by the LPnet system. This interface gives the LPnet system a set of “call-backs” that it can use to notify the Toad of important events. These events include the starting and stopping of the Toad and the arrival of SWAMP messages. Each Toad interface method is called through a separate thread. This allows the Toad to respond independently to an incoming message while managing long-term actions. Because Toads have full access to SWAMP messaging it is expected that they will make up the bulk of the serious agents in the LPnet system. Frogs on the other hand will provide an easier entry point for novice users and a good medium for quick one-off agents.

## 5. Discussion & Future Directions

As the LPnet system has been developed it has focused on the following goals and principals.

**Light-Weight:** Unlike most other multi-agent systems LPnet is designed to operate without specialized servers. This allows it to be used as a “normal” application.

**User-Centric:** Unlike many agent systems LPnet focuses on human interaction through agents and the chance encounters that make life interesting.

**Accessible:** LPnet is designed in the hope that it will bring networking and other experimental “Internet technologies” within the reach of untrained computer users.

**Open-Source:** The LPnet project will be released under the GPL license. All agents are also inherently open-source [GPL03]. This allows LPnet users to learn from each other's code.

To summarize these goals one could say that LPnet is designed to be a democratizing force on the direction of future Internet technologies. By making a multi-agent system accessible to normal users it is hoped that a community of “non-technical” people will be formed that will dream up uses for this technology that would otherwise not be developed. LPnet is also designed to provide “instant gratification” to novice programmers, thus inspiring more people to take up the art of programming. As LPnet is developed and made more user friendly, the reward of seeing a simple agent wandering around the network will become even more significant. This effect may also make LPnet a valuable tool for teaching middle and high school students about programming. The benefit of teaching a general purpose language like Python in the LPnet environment is that a program not much larger than “Hello World” can exhibit very complex behavior that would otherwise require pages of sophisticated code. As the LPnet community grows the popular uses of the system may even drive the commercial development of new Internet “technologies”.

## 6. Conclusion

A prototype of the LPnet system has already been developed and tested. Currently this prototype is being redesigned and implemented in a more extendable and robust form. As this incarnation of the LPnet system becomes usable, alpha versions suitable for programmers will be released [LPP03]. It is expected that this version will provide a solid base from which to build and refine the LPnet system. As this happens and documentation is produced the LPnet system should rapidly become accessible to a greater cross-section of people.

## References

[APA03] Apache, <http://httpd.apache.org/>, 2003.

[BMJ03] BottleMail, <http://www.bottlemail.jp/>, 2003.

[GPL03] GNU General Public License,  
<http://www.opensource.org/licenses/gpl-license.php>,  
2003.

[JPY03] Jython (java implementation of python),  
<http://www.jython.org/>, 2003.

[LIS82] Lisberger, Steven. "Tron", 96min, 1982.

[LPP03] LPnet Project, <http://www.lilypadnet.org/>,  
2003.

[PHP03] PHP, <http://www.php.net/>, 2003.

[PRL03] Perl, <http://www.perl.org/>.

[PYT03] Python, <http://www.python.org/>, 2003.