OverSwarm

http://syscall.org/doku.php/overswarm

A Framework for the Development and Evaluation of Distributed Swarm Intelligence Algorithms on the OverSim Simulation Platform

What's the Goal?

Enabling a comprehensive experimental validation of network oriented swarm solutions. OverSwarm simplifies the development and evaluation of distributed swarm Intelligence algorithms based on the concept of mobile ant agents.

How does it work?

By leveraging an existing peer-to-peer overlay simulation framework, namely OverSim. OverSwarm includes a domain specific language for describing agent's behavior. Our solution supports the development of new P2P overlays, extension of existing solutions, and execution of swarm algorithms on top of existing P2P overlays.

About OverSim

OverSim is an open-source overlay and peer-to-peer network simulation framefork based on OMNeT++ (an event based simulation environment). Several models for structured and unstructured P2P systems, such as Chord, Pastry, Kademlia, and GIA, are available. Oversim is developed by the Institute of Telematics at the Karlsruhe Institute of Technology. http://www.oversim.org

Faster prototyping

OverSwarm allows the programmer to **focus on the collaborative aspect of the algorithm** rather than on network related issues such as data serialization or the details of agents' execution. Distributed swarm intelligence solutions can be deployed and evaluated on top of peer-to-peer overlays already implemented in OverSim. Algorithms can be easily modified and tested, and can **exploit existing OverSim modules** or user defined functions written in C++.

- **Works like a scripting language**
- **Automatic memory management**
- **Strong transparent migration**
- **Access to user-defined functions**
- **Compiles to C++ code**

```
(var previous nil)
(if (< (rand) 0.5) (begin
    (set! previous (getThisNode))
    (migrate (getPredecessor))
    (var result (doSomething))
    (migrate previous)
    (if (> result 0)
        (doThis)
    else
        (doThat)))
else
    (migrate (getSuccessor)))
```

Ant Agent's Behavior

With probability 50% either migrate to the successor, or: migrate to predecessor, doSomething, then migrate back to the previous node and if the result of doSomething was greater than 0 doThis, otherwise doThat.

High Level

```
switch(packet->getType()) {
case 0:
    if (rand() < 0.5) {
        packet->setPrevious(this->getAddress());
        packet->setType(1);
        sendMessageToUDP(this->predecessor,packet);
    } else {
        sendMessageToUDP(this->successor,packet);
    }
    break;
case 1:
    int result = doSomething();
    packet->setResult(result);
    packet->setType(2);
    sendMessageToUDP(this->getPrevious(),packet);
    break;
case 2:
    if (packet->getResult() > 0) {
        doThis();
    } else {
        doThat();
    }
    break;
default:
    // Handle unknown message
}
```

Network Protocol

```
(var previous nil)
(if (< (rand) 0.5) (begin
    (set! previous (getThisNode))
    (migrate (getPredecessor))
    (var result (doSomething))
    (migrate previous)
    (if (> result 0)
        (doThis)
    else
        (doThat)))
else
    (migrate (getSuccessor)))
```

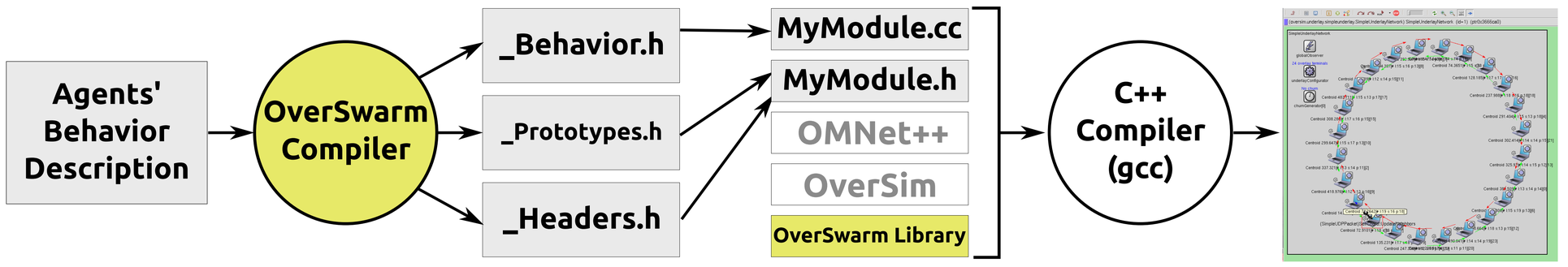
OverSwarm

More natural way of defining agent's behavior

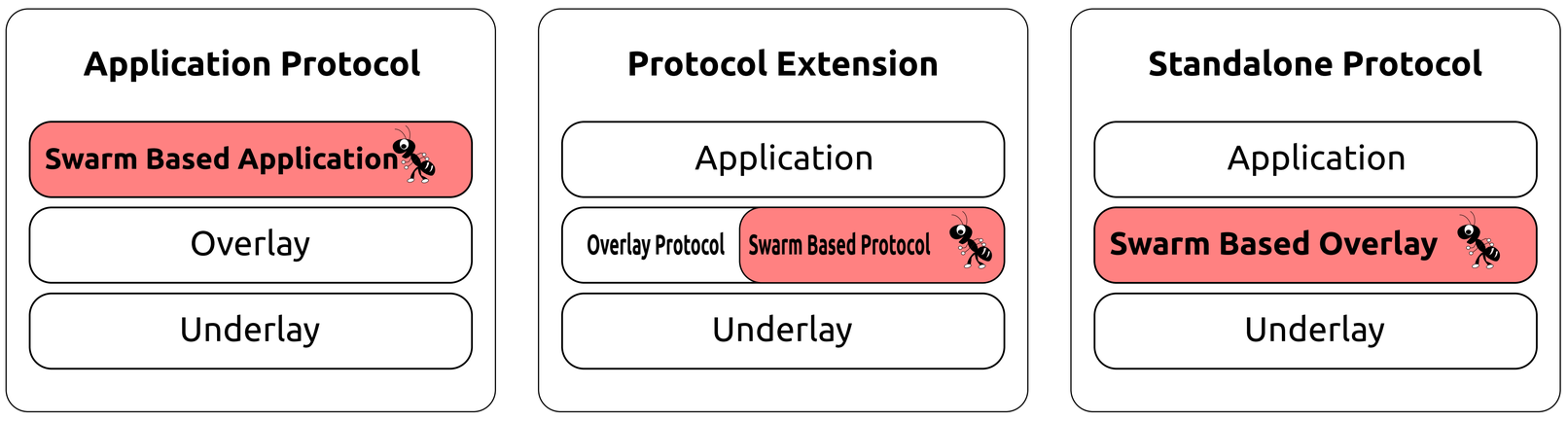
Swarm intelligence algorithms are often detailed by describing the behavior of each agent. Accordingly, it can be awkward to translate this into a message based network protocol, with the flow of operations being more difficult to follow. Conversely, thanks to OverSwarm's programming language agent's behavior is easier to describe and understand.

Toolchain

OverSwarm includes a compiler that takes the agent's behavior and generates the corresponding C++ code. This code can be integrated within an OverSim module and compiled to create a complete package ready for testing with the OverSim platform.



Multiple targets



With OverSwarm it is possible to target different levels of the OverSim framework. The swarm algorithm can either be run as application on top of an existing overlay, as an extension of an existing protocol (for example Chord), or as a standalone overlay protocol. This flexibility enables a wide range of experimentation and testing.

FNSNF Fellowship n° 134285
Fonds national suisse
Schweizerischer Nationalfonds
Fondo nazionale svizzero
Swiss National Science Foundation

Dr. Amos Brocco (amos.brocco@gmail.com)
Institut für Telematik, Karlsruher Institut für Technologie (KIT)