Organization

- Data communication is mesh
- Control communication is tree
- Synchronization is tree

Communication: Overview

Communication is preformed via:

- XML Formatted TCP/IP Streams
- NanoXML Library Used for XML Parsing
  - http://nanoxml.sourceforge.net/

Job Dispersal

- When a node requires a new partner, it sends a request to it's master.
- The master tracks overload levels for each slave node
- the least overloaded node creates a new instance

Job Dispersal

- If none of the slaves have a load of 0 (unloaded) the master sends a request for a fresh node to it's master.
- It's master either assigns it a better node, or tells it that it can't get any better.

Communications: Overview

RMI

Pro	Con
Easy To Code	To Difficult To integrate Non-Java Code

Communications: Overview

XML -> Multicast TCP/UDP

Pro

- Efficient Network usage

Con

- Overly Complex Code
  -lost packets
  -unordered data
  -small packet size

Communication: Example

```
<node action=“request”>
<load value=“3” />
<target>
<system host=“name” port=“num” />
</target>
</node>
```

Communication: Example

```
<node action=“new”>
<master >
<system host=“name” port=“num” />
</master>
<target>
<system host=“name” port=“num” />
</target>
</node>
```

Communication: Example

```
<node action=“ready” >
<system host=“name” port=“num” />
<load value=“0” />
</node>
```

Communication: Example

```
<node action=“assign” >
<master>
<system host=“name” port=“num” />
</master>
<target>
<system host=“name” port=“num” />
</target>
</node>
```

Communication: Example

```
<node action=“ready” >
<target>
<system host=“name” port=“num” />
</target>
<load value=“0” />
</node>
```

Communication: Example

```
<node action=“start” >
<task id=“5” />
<data>
<int name=“x” value=“5” />
.
.
.
</data>
</node>
```

Communication: Example

```
<node action=“change” >
<from>
<system host=“name” port=“num” />
</from>
<to>
<system host=“name” port=“num” />
</to>
</node>
```

Communication: Example

```
<output>
<data>
<int name=“x” value=“5” />
<string name=“y” value=“test” />
</data>
</output>
```

Organization

- Overall structure is similar to the pyramid network organization.
- Control is conducted along vertical connections
- data transfer between nodes are preformed along grid at base.
- Nodes above base are controlling hierarchy.

Control

- Whatever can be handled at a local level will be.
- If additional resources are needed, send a request up to the next higher level.
- New resources (processors) are added to apex of pyramid.

Similar Projects

- JTeD : http://www.cs.dartmouth.edu/~nicol/S3/jted.html
  - Written in Java
  - Self Organizing
  - Uses individual class files for individual simulation elements
  - Time Warp Synchronization
  - Designed for simulating network structure/activity

Similar Projects

- JSSF
  http://www.ssfnet.org/SSFdocs/ssfSimulators.html
  - Java Scalable Simulation Framework

Similar Projects

- Warped
  http://www.ececs.uc.edu/~paw/warped/
- Time Warp based simulation kernel
- Written in C++
- Runs under MPI for parallelism