RETSINA

Katia Sycara, PI

The Intelligent Software Agents Lab
http://www.softagents.ri.cmu.edu
http://www.cs.cmu.edu/~softagents

Description:

The RETSINA multi-agent infrastructure is a system of heterogeneous agents that adaptively interoperate to address a variety of domain-specific problems. Each RETSINA agent draws on a sophisticated planning and reasoning architecture to interleave distributed planning, information gathering and execution within dynamic and uncertain environments. RETSINA Agents and Infrastructure components:

1. **Interact in open dynamic environments**
2. **Retrieve and fuse information**
3. **Dynamically form teams to plan and re-plan**
4. **Interoperate with non-RETSINA systems**
5. **Find agents through middle agents**
6. **Interoperate with non-RETSINA systems**
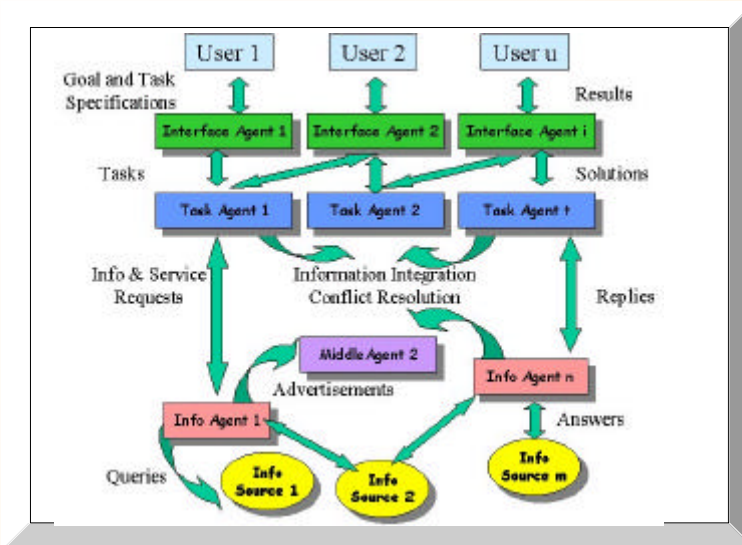
Results:

Demonstrated

1. **Dynamic and scalable middle agent infrastructure (including multiple ANSs, Matchmakers, Brokers, Facilitators, etc. through use of Discovery protocol (see opposite).**
2. **Location based Discovery and services.**
3. **Peer-to-peer agent interactions on mobile devices and *just-in-time* GUIs.**
4. **Algorithms for multi-agent reinforcement learning with limited sensory information.**
5. **Collaboration with other participants, including ALP, MIATA, CoAX and Teamcore.**
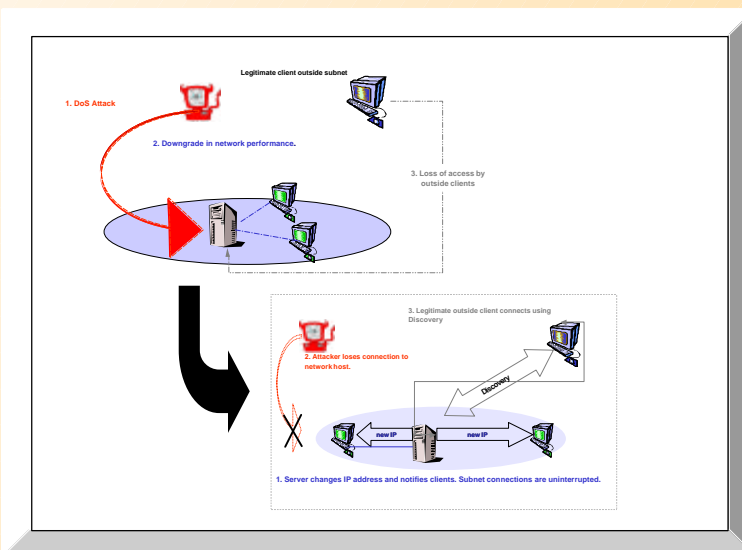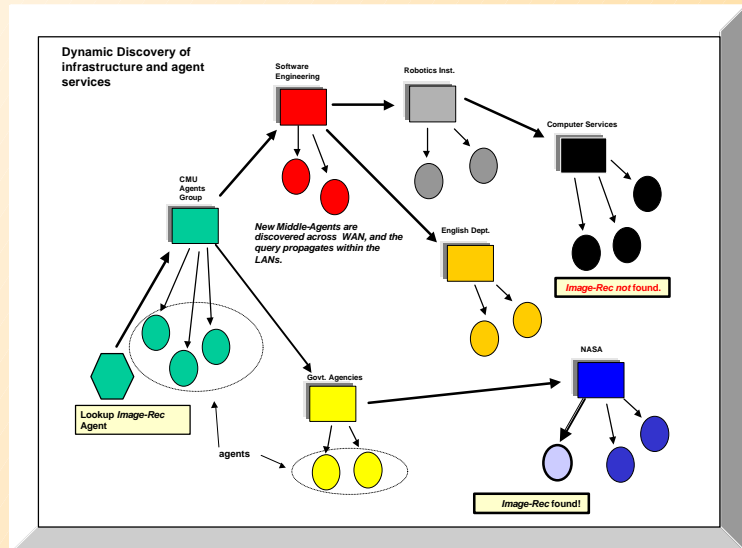
Recent Developments:

Socket Handoff Defense to DoS Attack

With the **Socket Handoff Defense**, a targeted organization maintains the operation of the networked infrastructure during a DoS attack. A lightweight socket handoff technology allows network computers to relocate out of harm's way by renumbering their IP network addresses.

Benefits:

1. The termination of the DoS attack. Spoofed or inaccurate IP addresses will be dropped and "left behind" in the "move."

2. Continuity of service. Relocation accomplished without interrupting services and network connections that are valid and already active. These connections will be automatically updated with the new and renumbered addresses.

3. Application transparency. Socket Handoff is implemented in the OS kernel. All network applications can benefit from it without needing to be rewritten.

4. Gradual phase-in. Allows connection to network of non-Socket-Migration-enabled Applications under normal conditions. Applications need a Socket-Handoff-established kernel to maintain a connection when the server relocates and hands off its new IP address.