Coordinated Behavior
- What type of emergent behaviors arise from groups of agents?
  - Individual agent frailty requires groups of coordinating agents
  - Malleable agent adaptation
  - Autonomous talent-balancing – agent drops some of its functionality because too many other agents can do it
  - Application survivability – what is the minimum number of agents required to ensure the clique survives?

Exploratory Behavior
- How does an agent explore a network?
  - Content-based remotely accessible shared memory
  - Remote communication vs. agent migration
  - Agent sleep schedule

Malleable Behavior
- How does an agent adapt to a changing context?
  - Agent instruction morphing
  - Dynamic instruction set
  - Agent reflection – so an agent can determine what code it has and what it needs

Clique Behavior
- What type of emergent behaviors arise from groups of agents?
  - Individual agent frailty requires groups of coordinating agents
  - Malleable agent adaptation
  - Autonomous talent-balancing – agent drops some of its functionality because too many other agents can do it
  - Application survivability – what is the minimum number of agents required to ensure the clique survives?

Coordinated Behavior
- How do agents coordinate?
  - Primitive abstractions supporting high-level decoupled interactions
  - E.g., shared tuple spaces
  - The choice of a coordination model for sensor networks is an open question

Research Implications
- Spatiotemporal Behavior
  - What type of spatiotemporal behaviors do applications exhibit?
  - Agents must be present at a certain time & place to perform
  - Applications tentatively categorized into 3 spatial structures:
    1. Sparse - Agents coordinate over large distances
    2. Compact - Agents are all located in the same general area
    3. Geometric - Agents must maintain a geometric shape, e.g., a circle around an intruder

- Security
  - How do we deal with malicious agents and privacy?
  - Lightweight mechanisms required
  - Virtual machine "sandbox"
  - Coordinated message scattering and collection

- Integration with IP Networks

- Container Security and Tracking

- Fire Tracking

- Habitat Monitoring

- IP Network

Experimental Test Bed
- Implemented on MicaZ, ported to MicaZ, and MIRACL Days25 motes
- TinyOS 1.1.14, NesC 1.2
- 4 Agents/mote
- 100 byte tuple space
- 46K ROM, 3.3K RAM
- Integrated with the Cricket Indoor Localization System

Performance Evaluation
- Migration instructions are more reliable because of hop-by-hop acknowledgements...
- …but remote tuple space operations have less overhead

This research is supported by the NSF under NOSS contract CNS-0520220