

## Architectures

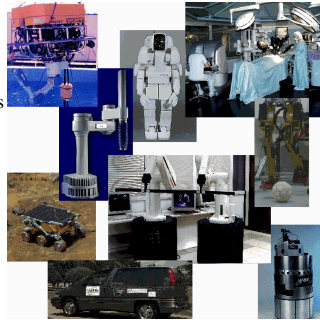
### Organizers & Chairs: Eve Coste-Maniere, Reid Simmons

#### Architecture, the Backbone of Robotic Systems

E. Coste-Maniere<sup>1</sup> and R. Simmons<sup>2</sup>

<sup>1</sup>INRIA Sophia Antipolis and <sup>2</sup>Carnegie Mellon University

- **ARCHITECTURE:** To manage the complexity of robotic systems
- **SPECIFICATION:** Modularity & reuse, programming environments – **VALIDATION:** Test & verification
- **EXECUTION:** Real-time, autonomy/decision, reliability, interactions
- **TRENDS FOR THE FUTURE:** System integration & standards

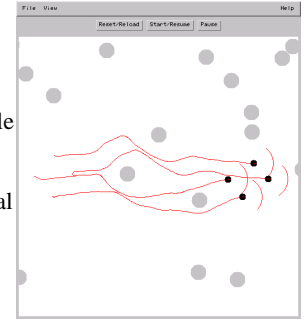


#### Social Potentials for Scalable Multi-Robot Formations

Tucker Balch and Maria Hybinette

Carnegie Mellon University

- New class of potential functions for group navigation
- Enables homogeneous large-scale formations
- Integrated with other navigational behaviors
- Demonstrated in simulation

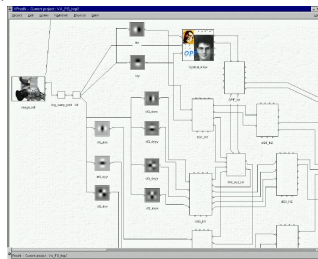


#### PredN : Achieving efficiency and code re-usability in a programming system for complex robotic applications

O. Stasse and Y. Kuniyoshi

Electrotechnical Laboratory, ETL

- Real-time, distributed architectures, portability and code re-usability
- Application model which avoids overspecification, general platform modelization
- Real-time visual attention system. 70 us latency between objects
- Trade-off between software generality and pure modelization



#### Around the lab in 40 days...

Rachid Alami, Raja Chatila, Sara Fleury, Matthieu Herrb, Felix Ingrand, Maher Khatib, Benoit Morisset, Philippe Moutarlier and Thierry Simeon  
LAAS/CNRS

- **LAAS:** LAAS' Architecture for Autonomous System
- A functional and a decisional level
- Example: implementation of a robust navigation system
- Contribution: concepts and software tools



#### Using Model Checking to Guarantee Safety in Automatically-Synthesized Real Time Controllers

David J. Musliner, Robert P. Goldman and Michael J. Pelican  
Honeywell Technology Center

- **Motivation:** Mission-critical autonomous systems (UAVs, spacecraft)
- **Approach:** Automatically build guaranteed real-time controllers, on-line
- **Details:** Plans are verified using model checking techniques
- **Result:** Novel application of formal methods yields reliable, self-adaptive controllers

