GENERAL DYNAMICS
Robotic Systems

Autonomous Command & Control (ACC) of Unmanned Vehicles

Presented by Aaron J. Cahill

GDRS

- Located in Maryland
- Division of General Dynamics Land Systems (Sterling Heights, MI)
- Product development, robotics, automation

Go Places & Look at Things

Need More Autonomy

- The Army, our customer requires that tactical robots need to operate more autonomously

Why More Autonomy

- Reduce operator workload
- Handle the 3 D's; dangerous, dirty, and dull missions
  - Reduce the risk of harm to soldiers
- Increase tactical effectiveness

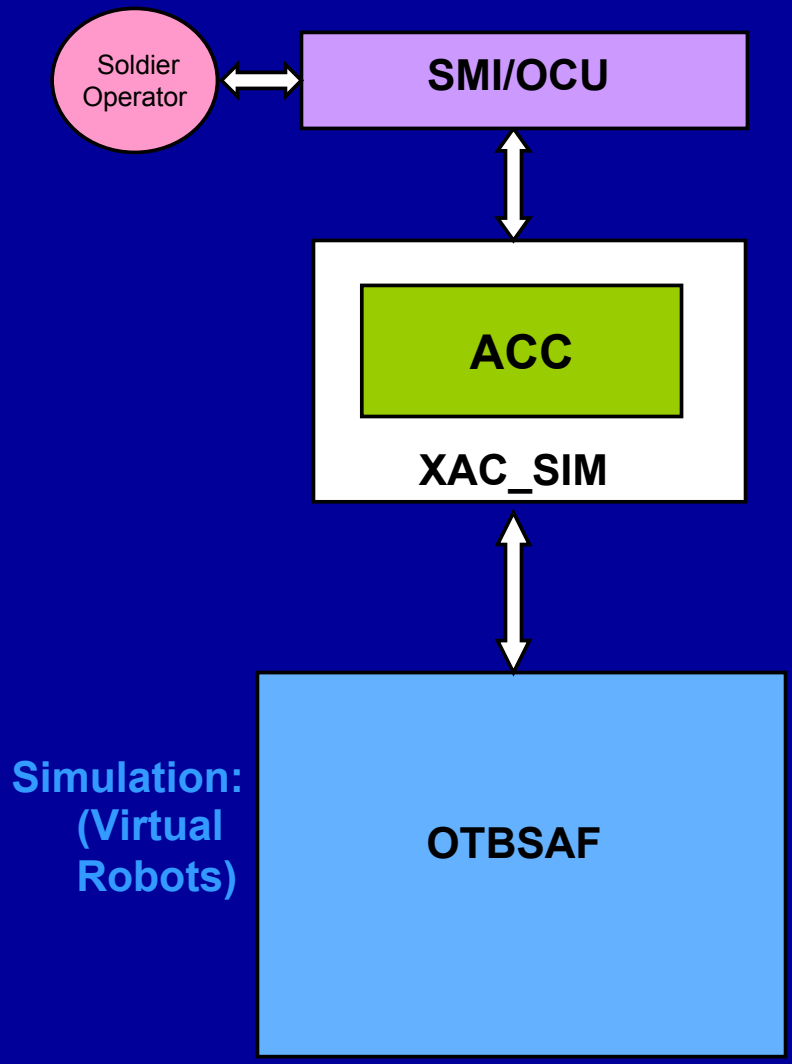
More Autonomy

- Accept commands to carryout missions, make a plan, execute the plan, the only human intervention occurring to approve or disapprove of the robot's plan
- React to the dynamic battlefield environment the way an expert soldier would
- Integrate robots with manned forces

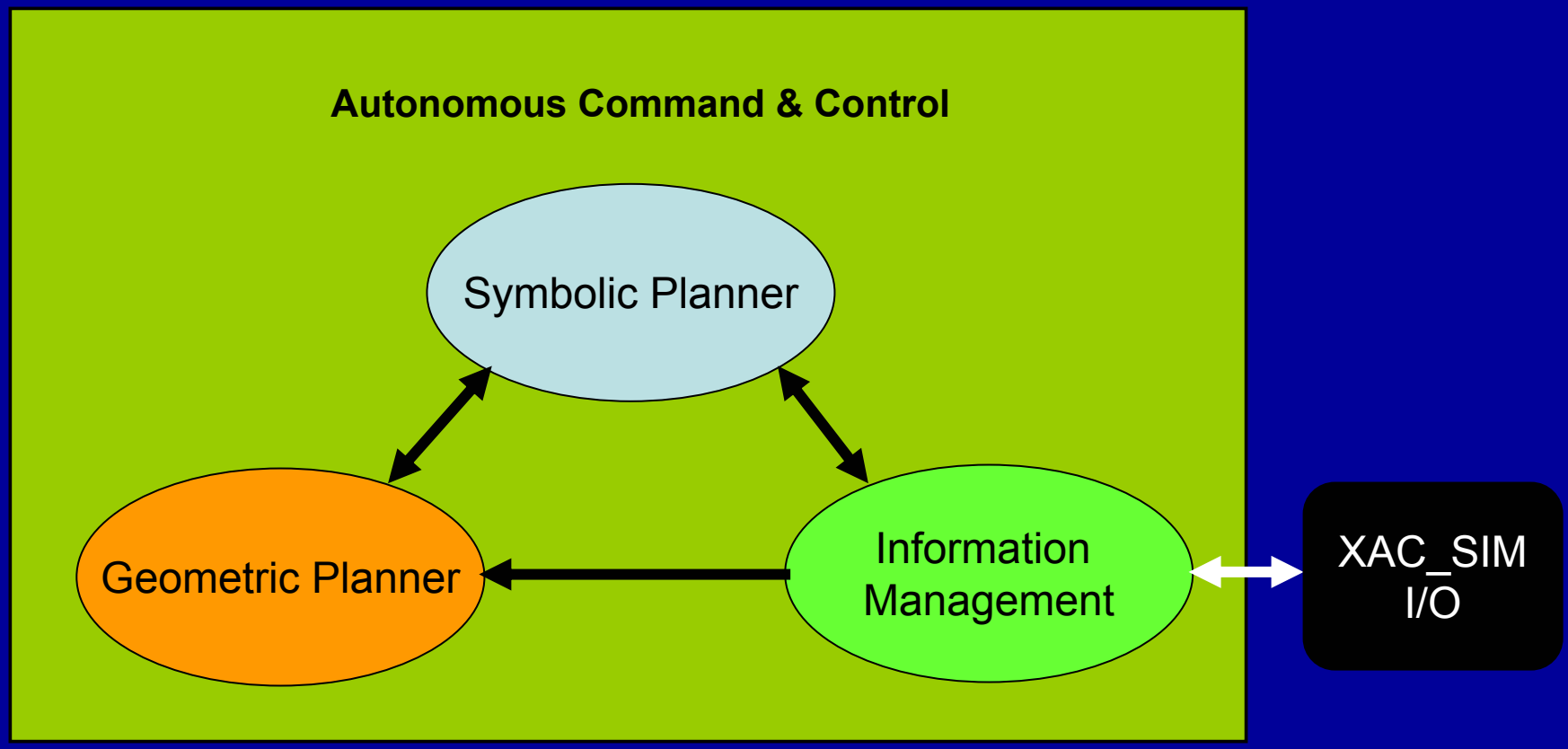
The ACC Approach

- Provide platform level (vehicle level) intelligent control for UGVs.
- Operators command and control ACC using Operational Command Language (OCL)
- Close loops with platform subsystems (e.g. autonomous mobility, RSTA suites, technical fires control)
- Communicate with other ACC nodes via a P2P comms link

ACC Build 0.X Architecture



ACC Components



Tactical Aspects

- Hybrid deliberate/reactive behaviors
- Individual Tasks, Collective Tasks
- Robotic Tactics, Techniques, and Procedures compatible with evolving Unit of Action (UoA) doctrine
- Seamless combined manned/unmanned operations

Why Soar?

- ✓ Goal driven
- ✓ And reactive
- ✓ Solves problems the way an expert would
  - – We would like to test a robot like the Army tests a soldier
- ✓ Traceable goal stack
- ✓ Agent to agent and agent to human communication

UV-Soar

- OCL Parser
- Phase System Planner
- Reactive behavior
  - React to Phase Lines
- Communicate manned and unmanned entities with a simulated radio
- Send reports

So Far, So Good

- ACC consisting of Soar, Geometric Planner, and IM functionality along with OTBSAF install in Redhat 9

?