Integration and State Summarization

Prof. Holly Yanco
Munjal Desai
Dan Hestand
Pragmatics Integration

Pragmatics (Alchemy)

GUI

Planner

Robot

State Summarization

World Database
Query:
Where are you?

Constraints:
- Time

Submit Query

Response Detail:
2

Communication Channel

Task:
Find

Morphine & Cash & (Ammo & Cell Phone)
Pragmatics Integration

- Planner
  - GUI
  - Find (Assault Rifle)
- Pragmatics
- Robot
Pragmatics Integration

- Planner
  - Find(Assault Rifle)
  - Search()

- GUI
  - Search()

- Pragmatics
  - Planner

- Robot
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Found(Water)
Pragmatics Integration

- Planner
- Found(Water)
- Pragmatics
- GUI
- Robot
Pragmatics Integration

- Planner
  - Task not complete
  - GUI
  - Robot
  - Pragmatics
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Found (Assault Rifle)
Pragmatics Integration

Planner → Found(Assault Rifle) → Pragmatics

GUI

Robot
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Task complete
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Task Complete
Stop()
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Find(Assault Rifle)
Pragmatics Integration

- Planner
  - Find(Assault Rifle)
- Pragmatics
- GUI
  - Search()
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Found(Water)
Pragmatics Integration

- Planner
  - Found(Water)
  - GUI
  - Robot
  - Pragmatics
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Task not complete
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Found(Hand Gun)
Pragmatics Integration

- Planner
  - Found(Hand Gun)
  - GUI
  - Robot

Pragmatics
Pragmatics Integration

- Planner
  - Task complete
  - Pragmatics
- GUI
- Robot
Pragmatics Integration

- Planner
- Pragmatics
- GUI
- Robot

Task Complete
Stop()
Query
Where are you?

Constraints
- Time

Submit Query

Response Detail
2

Communication Channel

Task
Find
TCP or SOA?

- TCP requires prior knowledge of server/client locations (IP address, port numbers)
- Relocation may invalidate those locations
- Requires in-depth understanding of socket programming, connection management, threading, ...
- Addition of new servers or new clients requires significant rework
- TCP only readily allows point-to-point type communication (i.e., specific client to specific server)
TCP or SOA?

- Service Oriented Architecture (SOA) views everything as a “service” to be produced or consumed
- Services are the central abstraction mechanism
- Services represent a piece of functionality made available to potential consumers
- Services are NOT applications!
- Types of services include simple/basic and composed
- Services provide a collaboration mechanism that is highly decoupled and interoperable
SOA-based Pragmatics Integration

- Commander GUI (Java)
  - GUI to Planner
  - GUI from Planner

- Message Broker (IceStorm)

- Planner (C++)
  - Planner from GUI
  - Planner to GUI
  - Planner to Robot
  - Planner from Robot

- Robot (C++)
  - Robot to Planner
  - Robot from Planner

- Pragmatics/Alchemy (Ruby)
Pragmatics Integration

- Based on the “enterprise service bus” (ESB) concept
- Initially, Publish/Subscribe messaging is used over TCP
- Request/Response implemented as paired messages
- Specific mechanisms hidden in a protocol adapter
- Component interaction defined via an interface definition language
- Interface represents the service call
- Interface methods represent the Request/Response message pairs
- Interaction is mediated by either local libraries or standalone processes
Definition

• State Summarization: A system that improves situation awareness by providing a condensed and coherent explanation of the actions of the robot along with the rationale for those actions.
Features

• Queries: questions that the commander can ask

• Temporal modifiers: add time based constraints

• Spatial modifiers: add location based constraints

• Verbosity: controls amount of information being reported
Queries

- Where are you?
- Where have you been?
- What is the task?
- What have you done?
- Do you see anything?
- What have you seen?
- Is the room clear?
- Did you find ‘x’?
- How many ‘x’ did you find?
Temporal Modifiers

- **None**
  > What have you seen?

- **Now**
  > What have you seen now?

- **Recently**
  > What have you seen recently?

- **Time based**
  > What have you seen between t1 and t2?

- **Since event**
  > What have you seen since the start of task?

- **Since Seen**
  > What have you seen since seeing the bomb?
Spatial Modifiers

- None
  > What have you seen?
- Coordinate
  > What have you seen at (x,y)?
- Rooms
  > What have you seen in room A?
- Proximity
  > What have you seen in proximity of (x,y)?
- Within
  > What have you seen within m meters of (x,y)?
- Relative to me
  > What have you seen within m meters of commander?
SOA-based State Summarization

Commander GUI (Java)
- GUI to Robot
- GUI from Robot

Message Broker (IceStorm)

State Summarization Adapter (C++)
- State Summarization from Robot
- State Summarization To Robot

Robot (C++)
- Robot to State Summarization
- Robot from State Summarization
- Robot to GUI
- Robot from GUI

State Summarization (Python scripts)
State Summarization Integration

- Based on the “enterprise service bus” (ESB) concept
- Paired message adapters provided for each language and interface; one adapter mediates incoming messages, the other mediates outgoing messages
- Robot has 4 adapters because it sits between the GUI and the State Summarization scripts
- State Summarization scripts are handled by an adapter process to mediate messaging and call the appropriate script(s)
Which Is Better?

- Given multiple architectural approaches to integration, how do we make a decision on which approach is “best?”

- Architectural assessment methodologies exist
  - Only “notionally” quantitative
  - Not designed with consideration for selection from a set

- Self-similarity has been identified as a feature of component based architectures
  - No available, extant work on quantifying self-similarity for components
  - Some work on complexity metrics based loosely on self-similarity
Integration Wrap-up

- TCP or other point-to-point mechanisms work efficiently on a small scale but do not scale
- SOA approach decouples the client and server
  - Easier relocation of servers/clients
  - Easier server replication
  - Easier addition of new servers/clients
- ESB concept allows service providers and consumers to be added at will (well, almost)
Test Scenario
Year 3 Plans

- Queries
  - How do I get to you?
  - Do you need help?
  - Time estimation based questions
  - Questions based on robot’s ability to complete tasks

- More temporal and spatial modifiers
- Dynamic importance based on entropy
Year 3 Plans

- Finish Pragmatics to SOA integration (95%)
- Integration with LTL - PAR
- Integration of state summarization with parsing
- Validating state summarization through user testing
- Generative XML based SOA integration