Agent Steve

Khaled Blah

Embodied Agents Research Group

July 2nd, 2009
Who is Agent Steve?

- **Steve:** SOAR Training Expert For Virtual Environments
- helps students learn physical, procedural tasks in a virtual world
- performs tasks as example, monitors students, provides assistance
Introduction

is this part redundant?

- impractical to provide training on real equipment —> mock up virtual world of the actual student’s working environment
- primary research areas: intelligent tutoring systems, computer graphics, agent architectures
- Steve has pedagogical capabilities —> can answer questions like ”what next?” and ”why?”
- animated body —> more human like assistance
- Steve monitors the virtual world, maintains a plan to complete his current task, revises this plan unexpected events
- course author provides domain knowledge for Steve, course author is not a programmer
- Steve can coexist with other agents and people —> goal: team training
- this paper: single students on one-person tasks
Working with students

- Student has virtual world interface: head-mounted display with microphone
- Steve introduces tasks — begins demonstration
- Steve explains what he does verbally while demonstrating it, uses gaze to address the student — quality of natural speech?
Working with students - Example

**Speaker** conversation

Steve: "I will now check the oil level"
Steve: "First, pull out the dipstick."
Steve: "Now we can check the oil level on the dipstick. As you can see, the oil level is normal."
Steve: "Next, insert the dipstick"
Steve: "Make sure all the cut-out valves are open."
Steve: "Open cut-out valve three"
Steve: "I will now perform a functional test of the drain alarm light. First, check that the drain monitor is on. As you can see, the power light is illuminated, so the monitor is on"
Student: "Let me finish."
Student: "What should I do next?"
Steve: "I suggest that you press the function test button."
Student: "Why?"
Steve: "That action is relevant because we want the drain monitor in test mode."
Student: "Why?"
Steve: "That goal is relevant because it will allow us to check the alarm light."
Student: "Show me how to do it"
Student: "What should I do next?"
Steve: "I suggest that you press the reset button on the temperature monitor."
Steve: "That completes the task. Any questions?"
Steve: "That action was relevant because I wanted to dampen oscillation of the stage three gauge"

**Virtual world action**

moves over to the dipstick
looks down at the dipstick, points at it, looks back at the student —>Steve pulls out the dipstick
Pointing at the level indicator
Steve pushes the dipstick back in
Steve looks at the cut-out valves, one is open
pointing at the open valve
Steve points to the alarm light
student takes over, Steve steps aside to monitor
looks at Steve, he replies
conversation
conversation
conversation
conversation
Student understands, but is unsure how to do it
Steve moves to button and pushes it —>alarm comes up, indicating it is functioning properly
the student recalls that she must extinguish the alarm light, pushes the wrong button, different alarm light to illuminate
Student presses the reset button, presses the correct button, Steve looks at her
Student asks why he opened the cut-out valve 1

task completed
Steve’s capabilities

- generate and recognize speech
- demonstrate actions
- use gaze and gesture
- answer questions
- adapt domain procedures to unexpected events
- remember past actions
Virtual Worlds for People And Agents

- separate components in separate processes
- communication via message exchange
- Simulator: controls virtual world behavior
- Visual Interface: head-mounted display, data glove, pinch glove, 3D mouse for clicking on objects — all treated the same by visual interface component
- Visual Interface: receives messages from the other components (mainly simulator) and outputs a 3D world
- Visual Interface: informs the other components of human interaction what about other agents?
Audio: for humans, receives messages from the simulator, location and audible radius, broadcasts appropriate (context) sounds to the headphones.

Speech Generation: for humans, receives text messages from other components (mainly agents), converts the speech to text, broadcasts the speech on the human’s headphones.

Speech Recognition: for humans, recognizes speech, outputs semantic tokens to other components, Steve does not understand natural language.

Agent: each agent runs as separate component.
- Message Dispatcher: all messages are sent to MD
- Message Dispatcher: component registers for messages in which it is "interested"
- Message Dispatcher: dispatcher forwards messages to all interested components
- Message Dispatcher: example - visual interface components register for virtual world change messages
- Message Dispatcher: increases modularity (components need not know interface to other components, thus extensibility (adding new ones without affecting existing ones)
Steve’s Architecture

Steve consists of three main modules:

- **perception**: coherent state of the world, monitors messages from MD, identifies relevant events in the virtual world (Example: Student takes action)
- **cognition**: interpretation of messages from the perception module, chooses goals, executes plans to achieve goals, sends motor controls messages
- **motor control**: decomposes messages into a sequence of lower-level commands that are sent to the other relevant components via MD, provides locomotion/gaze/gesture and

---

**Domain knowledge**:

- **course author** gives Steve domain knowledge in a declarative language
- **declarative language**: for domain experts and not programmers
- **perceptual knowledge**: important objects, relevant object attributes, spatial information (more later)
- **task knowledge**: procedures for accomplishing domain tasks, text fragments
Perception - The State Of The World

- The simulator maintains most information on the state of the world (in attribute-value pairs).
- Perceptual knowledge provided by course author includes a list of relevant attributes.
- Perception module registers for messages indicating changes in these attributes.
- Perception must satisfy cognition module’s need for a coherent state of these attributes.
- Some messages from the simulator indicate simultaneous changes (Example: light should be illuminated — message: button depressed, light is on).
- To avoid inconsistent state of the simulation snapshot: either see the effects of all simultaneous changes or none at all (— start & end messages).
Steve needs to know spatial properties (position, orientation, spatial extent) of objects if he is to handle them.

- Perception module queries visual interface components for such information.
- Perception module can also ask other agents.
- Perception module keeps track of position of Steve relative to objects.
- The course author needs to provide: front vector, grasp vector, press vector, agent location.
Track of humans:
- visual interface component uses sensor on human’s head-mounted display to track them
- perception module keeps track of humans field of view when working with them, perception module asks student’s visual component interface
- subsequently: visual interface sends message to perception module when object leaves or enters student’s field of view

Path Planning:
- Steve carves virtual world in a graph, nodes between objects that Steve can move to without colliding with anything
- course author can manipulate the graph if necessary
- result: Steve has perceptual knowledge for navigation in the virtual world
cognition module might miss something if it could only ”see” periodic changes —> events

cognition module receives a snapshot of the world and a list of important events since the last snapshot

state changes: perception module passes along a few important events like feedback on Steve’s object manipulations

actions on objects by humans, human’s visual component sends messages

human’s speech: start message and end message with semantic token

agent’s speech: agents can listen (register) for when other agents start speaking, agents send separate messages with semantic representation
Cognition

- authors main task: building a set of domain-independent pedagogical capabilities (on top of SOAR)
- course author must provide domain knowledge written in SOAR?
Cognition - Domain Task Knowledge

Task: functional-test

Steps: press-function-test, check-alarm-light, extinguish-alarm

Causal Links:
- press-function-test achieves test-mode for check-alarm-light
- check-alarm-light achieves know-whether-alarm-functional for end-task
- extinguish-alarm achieves alarm-off for end-task

Ordering constraints:
- press-function-test before check-alarm-light
- check-alarm-light before extinguish-alarm

- Steve represents domain task as hierarchal plans
- task: a set of steps, steps: a set of primitive or composite actions, composite actions are tasks?
- ordering constraints between tasks partial order?
- causal links: steps achieve goals which might be preconditions for other steps
- course author must define goals and primitive actions
- goals: attribute-value pairs, two kinds of goals: attributes of virtual world (e.g. push button, light is on) or his own mental state
- sensing actions: learn about the state of attributes of the virtual world at a particular point in a task (e.g. check-alarm-light)
Cognition - Decision cycle

- input phase: get perceptual information
- goal assessment: using the perceptual information to determine which goals of the currents are satisfied
- plan construction: construct a plan to complete the task
- select and execute operator: a set of rules that represent Steve’s capabilities
Cognition - Goal assessment

- course author defines a goal, a SOAR production rule is generated
- rule checks current perceptual input or mental state
- goal becomes satisfied → rule ”fires”, goal becomes unsatisfied → rule ”retracts” possible looping?
Cognition - Plan Construction

- constraints: explain actions, handle unexpected results, follow standard procedures whenever possible
- for each decision cycle: construct plan
- plan construction: find relevant, unsatisfied goals and primitive steps that achieves it, find preconditions
- execute plan: determine which steps to do next
Choosing a task:
- There may be several possible steps to take at any point during a task
- But: human conventions may suggest a certain task to go first
- Discourse focus: humans prefer one subtask at a time
- Discourse focus represented as a stack, Steve pushes a step onto the stack, top element in focus of demonstration, the bottom element of the stack is main task
Demonstrating a task step:

1. Move to the relevant object
2. Explain what he is going to do (text fragment, repetition acknowledgment, Cue phrases: ”First, ...” - ”Now we can...” - ”Next, ...”)
3. Perform the task (send motor commands, monitor perception module)
4. If appropriate, explain results

Let me finish: if the student says ”let me finish.”, Steve switches to monitoring.
Demonstration - Monitoring a Student

- Use own plan to complete tasks and assess students, answer questions
- follow plan but: recognise achieved goals of sensing actions, monitor the student’s view field —> assume the student caused the changes
- evaluate a student by matching his actions with his own task assessment, give feedback
- ”What should I do next?”: Steve suggests the next step in his own plan, Steve can adopt to student’s deviation
- ”Show me what to do.”: demonstrate the next step, if there are multiple next steps —> chose one randomly
- Explain the causal structure of a task if the student asks for it, reply to follow-up questions until the original task is reached

After-action review: Student may ask questions about certain steps during the demonstration, Steve has no ”active” plan, uses SOAR debriefing system ”to go back in time”
Motor Control

- receives commands from the cognition module: speak, move objects, look at objects, nod, point
- there is a motor command for each primitive action
- lower-level body control capabilities: press objects, flip switches, turn valves, move objects short distances, pull and push objects
- motor control module sends these commands to the message dispatcher: actions, speech, body animation
- communication with the student: head (gaze) and hand (pointing, moving objects)
Motor Control - Locomotion

- cognition module sends motor commands
- Steve uses a graph to move around in the virtual world without colliding with objects or people (see: Perception - Path Planning)
- Steve determines his target using objects radius and front vector (provided by course author)
Motor Control - Gaze & Hand Control

- Steve looks where he’s going, at objects (also: when pointing at them), at humans (waiting, monitoring, nods when informed of something)
- Motor control seeks bounding sphere of objects, sends follow-up messages to starting moving towards the object
Thank you for your attention!