Agent Systems Project

and

Use Case Maps

Presenters:
Ray Buhr, Andrew Miga, Daniel Amyot

Non-presenting University participants:
Dorin Petriu, Darcy Quesnel, Kevin Lam, Luigi Logrippo
Outline

OVERVIEW (Ray Buhr)
  • How did we get here? What are we doing? Where are we going?

TOOLS (Andrew Miga)
  • UCMnav tool (Andrew Miga)
  • Agent metamodel tool (Kevin Lam and Darcy Quesnel)

THEORY AND APPLICATIONS (Daniel Amyot)
  • FI contest (Daniel Amyot, Dorin Petriu)
  • UCMs, Agents, and OPI (Dorin Petriu)
  • UCM-based testing (Daniel Amyot)
  • Agent frameworks (Darcy Quesnel)

CONNECTIONS (Ray Buhr)
  • Mitel (Agent project group and a new product group)
  • UCM/Performance (Murray Woodside, Dorina Petriu, Greg Franks, Nortel)
  • Announcements (UCM user’s group, acting PI for CITO agent project)

SUMMARY OF ACCOMPLISHMENTS
  • One summary slide, for reference, not presentation
UCM Navigator Use Case Map Graphical Editor
( A. Miga )

Status

• Developed as a master’s thesis work from a prototype. Development continues until February 1999.
• Supports most of TSE paper except layering and exceptions.
• Being evaluated by various individuals and organizations including Mitel, Nortel WIN project, SEI, Nortel Performance project. Is the object of a study by psychology students involving members of a UCM graduate course and which analyzes the usability of the tool interface
• The tool and the M.Eng. thesis are available on the web at http://www.sce.carleton.ca/rads/agents

Capabilities

• Provides graphical editing environment for UCMs with XML and EPS output
• XML file format allows interoperability with other software engineering tools such as agent tools.
• Extensions for agent system development include support for dynamically plug gable behaviour patterns and specification of goals.
Advantages of Tool Support for UCM Methodology

- Greatly reduces effort required in maintaining and manipulating Use Case Maps, enabling their use in large projects
- Validity checking of tool ensures that all maps are drawn in a syntactically correct fashion
- Support for the concept of stubbing allows complex UCM designs to be navigated in an on-line fashion

Future Extensions

- Addition of advanced editing features such as cut and pasting of paths between maps, extension of drawing area
- Support for multiple root maps and ability to overlay groups of paths on the same map
- Enhanced support for various output files, different EPS formats, MIF files
- Support for features desired for agent work
- Incorporation of feedback from UCM tool usability study
Nature of Tool Interface

Terminating Call Screening (TCS) plug-in for PCR stub. (Figure 7)
Agent Systems Project and Use Case Maps

UCM Navigator Support for Dynamic Behaviour

Plug-in submaps are bound to root level maps by binding input and output points of stubs to those in the submap.

Example root map, Plug-in Map selection dialog and expanded plug-in map.
Towards an industrial-strength UCM tool

Complete implementation requires level of effort several times that of present tool

- Integration with commercial tools. For example, extraction of class relationship diagrams from Rational Rose and ObjectTime for use as the component substrate for UCMs. This means presenting the diagrams as they look in the original tool and providing the ability for the user to select elements in them as full fledged elements of the UCM component substrate to which paths can be bound.

- Present the binding between stubs and plugin maps as well as path continuity through stubs more visually so it can be seen easily in the root map.

- Ability to view flattened maps by expanding stubs into parent map (through numerous levels of stubbing) through nonlinear transformations with automatic layout capability. Includes the ability to highlight routes.

- Automatic generation of documentation reports including text and diagrams should be possible, with the level of detail selectable from the present low level to all detail, and with control over presentation formats and content.

- Much better layout of labels in diagrams.

- Graphical/textual specification of composition for dynamic plugin selection.
Agent Metamodels Tool
(D. Quesnel and K. Lam)

- Based on thesis work by D. Quesnel. Currently being extended by K. Lam.
- Being evaluated by Mitel for use in CITO Agent project. Available on web.
- Java-based tool which extracts agent behaviour information from the XML output of the UCM Navigator to form skeletons of agent metamodels allowing designers to augment them in both tabular form (Conversational and Agent Internal models) and graphical form (Jurisdictional and Usage Rights models).

Agent Internal Model

Agent Usage Rights Model
Feature Interaction Experiment
(D. Amyot and D. Petriu)

Motivation

- Feature interaction (FI) is and will remain a challenging problem.
- Use some of the best ideas of two complementary approaches:
  - UCMs for visual description and integration of causal scenarios.
  - LOTOS for formal specification and V&V.

Our Proposal

1) Avoidance at design time with visual scenarios (*Use Case Maps*).
2) Validation of desirable interactions and detection of remaining undesirable interactions with a *LOTOS* prototype and scenario-based testing.

Experiment

- Illustration with the First Feature Interaction Contest example.
- 12+ features captured and integrated with UCMs. Some FI avoided.
- 4 features specified and tested with LOTOS. Three FI detected and resolved.
Methodology

Verdicts

- At least one test case from the individual feature set has failed.
- At least one test case from the feature pairs (FI) set has failed.
- At least one probe has not been visited by the entire test suite.
- The test suite has passed successfully, and all probes have been covered.
Some Conclusions

- Design decisions are necessary, although the burden of the integration is mostly taken care of at the UCM level.
- Sequences of stubs used in the integration of scenarios at the level of UCMs help to avoid some interactions between features.
- Interactions between features in one stub (e.g., INFB and TCS) are still possible.
- Some interactions can be avoided with deterministic and complete preconditions and by composing plugins in stubs according to the intent of the features.
- Test suites for detecting interactions between pairs of features are constructed on top of existing test cases, hence promoting reuse and consistency among tests.
- Good tool support for the UCM integration (UCM Navigator) and for the validation and coverage measurement of the LOTOS specification (LOLA) suggests that this approach can be used in an iterative and incremental design process.

Future Work

- Comparison with other LOTOS-based techniques.
- Linkage of the OPI model to the UCM notation.
- Adapt to agent prototypes derived from Use Case Maps.
UCM-Based Testing (D. Amyot)

- Concerned with the existence (or the absence) of traces or scenarios in the specification. Such scenarios can also represent interactions between features.
- Test selection strategies based on the coverage of UCM paths. We already have 19 testing patterns for the derivation of test purposes from UCMs.
- Acceptance and/or rejection test cases for abstract sequences.
- Theory based on LOTOS testing on the way.

**Coverage**: Pattern Alternative - All paths

- Abstract sequence 1: <R, V, F, S>
- Abstract sequence 2: <R, V, O, M>

Generation of acceptance and rejection test cases for each abstract sequence.

---

**Figure 2. Derivation of Validation Test Cases from UCMs**
UCMs and OPI (D. Petriu)

- Larger topic is migration from UCMs to OPI models and (hopefully) vice versa.
- Major point of interest: Can graph grammars be used on resulting OPI models to aggregate and compose different parts of a model?
- This would mean that integration of UCMs could possibly be automated or semi-automated at a level other than UCMs (interesting point to investigate).
- Another point: Can OPI concepts be integrated to the UCM notation? This would help the generation of rejection test cases from UCMs.
- The FI work from this summer will serve as the model on which we will base the work to move from UCMs to OPI (and hopefully vice versa).
- Good starting point because it is a reasonably complex system, but one which we understand fairly well at this point.

Status and Future Work

- First contact established with Mihai Barbuceanu.
- Two days a week on thesis starting in January, full-time starting in May.
- Investigate the generation of OPI models from the UCM Navigator XML files using the Agent metamodel tool.
Agent Frameworks (D. Quesnel)

Work accomplished

- Contributed to the development of an experimental agent prototyping environment (each agent in this environment is an object with an associated thread of control, a rule engine per feature, and access to tuple-space communication common to all agents). Contributed as coauthor to a paper presented at PAAM98 that described an application of this environment to feature interaction.

- Contributed to the development of the agent metamodel tool described earlier that is being developed to help provide part of a bridge between UCMs and prototypes of the kind supported by the above environment.

Current and Future Work

- Currently studying the framework literature and some existing non-agent frameworks and tools (e.g., JavaBeans)

- Objective is to develop an approach for agent prototyping that combines framework ideas with what was learned from the development of the agent prototyping environment mentioned above.

- This work is still in the studying and thinking stage.
Connections

Mitel

- Agent project with Tom Grey and Serge Mankovski.
  - Joint papers. UCMNav tool and agent metamodel tool evaluations.
  - Other possible Mitel applications.
    - UCM use within a product group starting a new product

From UCMs to communicating-state-machine models

- Methodology developed by Francis Bordeleau (Phd thesis in progress)
- Being applied in industry in a local company (CML)
- Real-Time-UML connection

UCMs/Performance

- Part of Woodside’s CITO project (acting PI Dorina Petriu, researcher Greg Franks).
- Nortel participation.
- UCMNav tool extended for entry of performance-related info.

Wireless Intelligent Network (WIN) project

- For documentation and formalization of standards (Logrippo’s CITO project)
Agent Systems Project and Use Case Maps

Connections (continued)

UCM use in SEI

- Jeromy Carriere (developer of the first quickie prototype of UCMNav) working with a company on an aerospace application. UCMs being used, including UCMNav tool.

UCM User’s Group being formed

- Enough interest and use in various companies that a forum is needed for exchange of ideas, championing wider use of UCMs, encouraging the development of tool support, etc.
- Point person is Daniel Amyot. Daniel is already in the process of registering a web site name and creating a site.
- Lobbying to be conducted in the UML/Rational/Objectime community for integration, support of UCMs. Ammunition is solicited from key user companies. Interactions already started. Meeting with Jacobson in January. Invitation to present to UML real time committee chaired by Selic.
- Future of the UCMNav tool is an issue.

New acting PI for Agent Project Dec 98 through Jun 99

- Luigi Logrippo
Accomplishments

Two tools, one relatively mature soon, one still immature. Penetration of tools and ideas into industry.

Presentations and demos at PAAM’98, FIW’98, CITO (Ott. & Tor.), Carleton U., U. of O., and Mitel.

Publications


Upcoming
