

Two HICSS98 papers

1. *A High Level, Visual Notation for Understanding and Designing Collaborative, Adaptive Behaviour in Multiagent Systems*
2. *Applying Use Case Maps to Multi-Agent Systems: A Feature-Interaction Example*

Combined slide package with newer view:

www.sce.carleton.ca/ftp/pub/UseCaseMaps/HICSS98slides.ps (and .pdf)

R.J.A. Buhr, D. Amyot, M. Elammari,
Carleton University, Ottawa, Canada

buhr@sce.carleton.ca, damyot@csi.uottawa.ca, elammari@scs.carleton.ca

T. Gray, S. Mankovski
Mitel Corporation, Kanata, Canada
{tom_gray, serge_mankovski}@mitel.com

Slide 1

Key points

- Telecom systems are a fertile field for agents.
- Telephony FI is an example: New features are added, and conflicts must be resolved, often dynamically; agents offer the prospect of incrementally adapting.
- The dynamic nature of agent solutions presents great difficulties for telecom software designers.
- Contribution 1: UCMs are used to express how a system *dynamically modifies its own large scale "behaviour structures"*; high level designs/prototypes of multi-agent systems flow in a systematic way from this.
- Contribution 2: FI resolution approach that can be systematically designed and implemented, *competing feature engines*.
- Part of an intended approach for future commercial systems.
- Papers address approach and FI; slides give newest view.

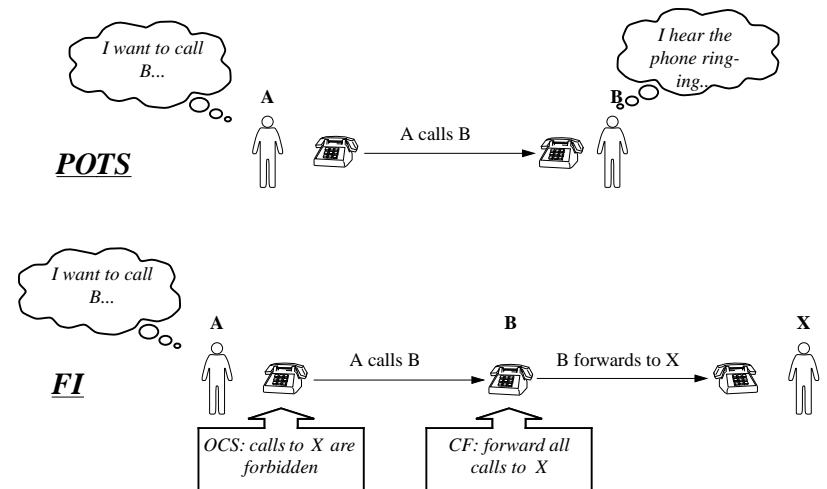
Slide 2

Road Map

- Intertwined intro to UCMs and FI through several FI examples (agents part of the picture, but secondary for the intro)
- Powerful aids to visualization of dynamic situations in agent systems: concept of system "behaviour structures" and of a system modifying its own "behaviour structures"
- Concept of competing feature engines derived from UCMs leads to a promising style of agent solution for conflict resolution problems
- Mapping of these concepts to agent models/prototypes
- Conclusions

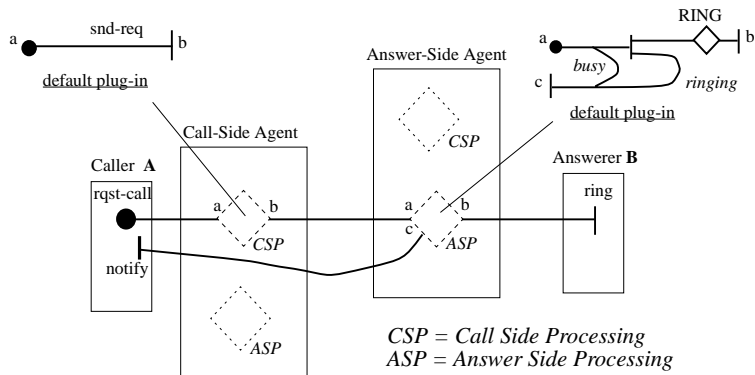
Slide 3

Case study explaining both FI and UCMs



Slide 4

POTS with agents and UCMs

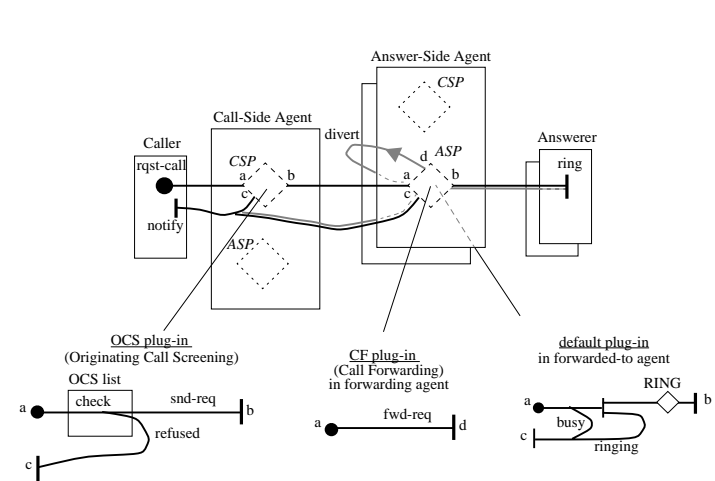


UCMs express "behaviour structures"

- embody causal sequences without specifying details
- express dynamic modification directly ...

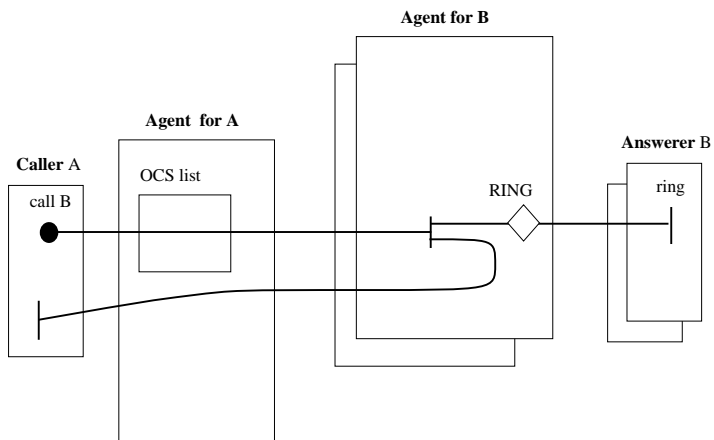
Slide 5

Possible Feature Interaction



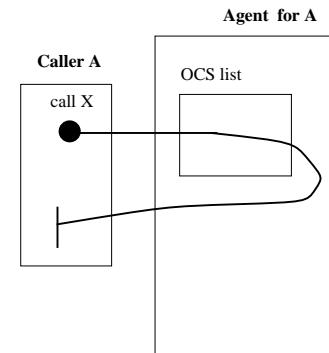
Slide 6

Correct "behaviour structure"



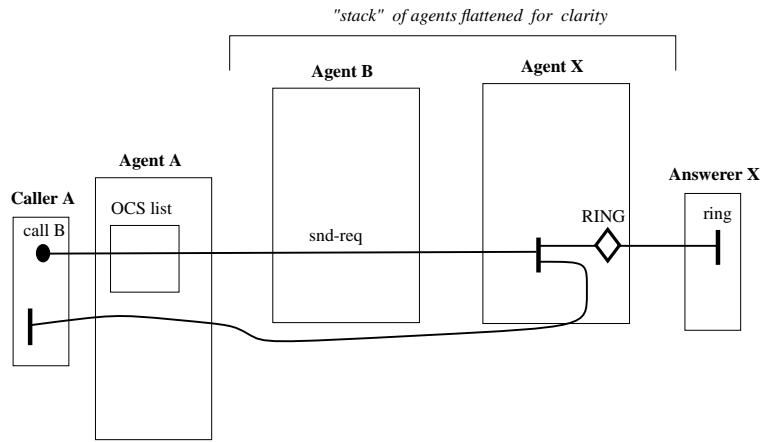
Slide 7

Correct "behaviour structure"



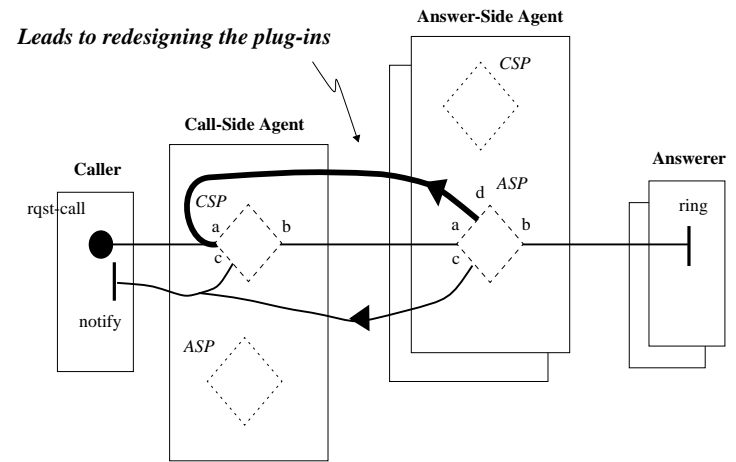
Slide 8

Incorrect "behaviour structure": Feature Interaction



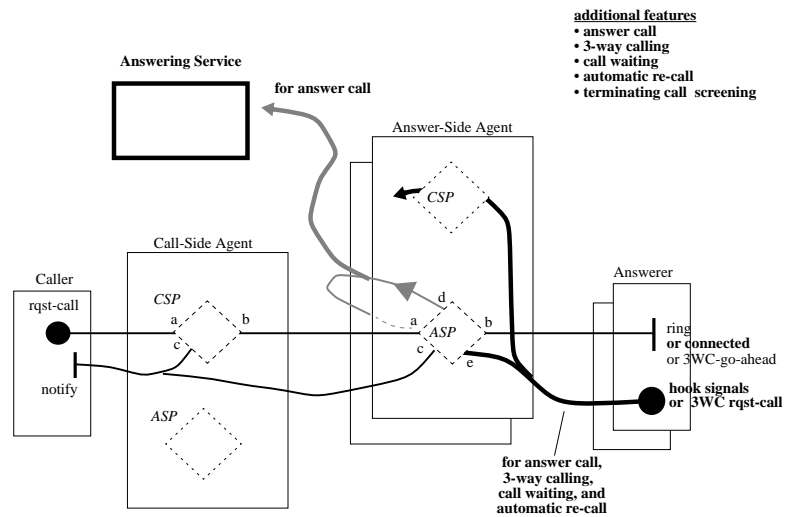
Slide 9

Solution



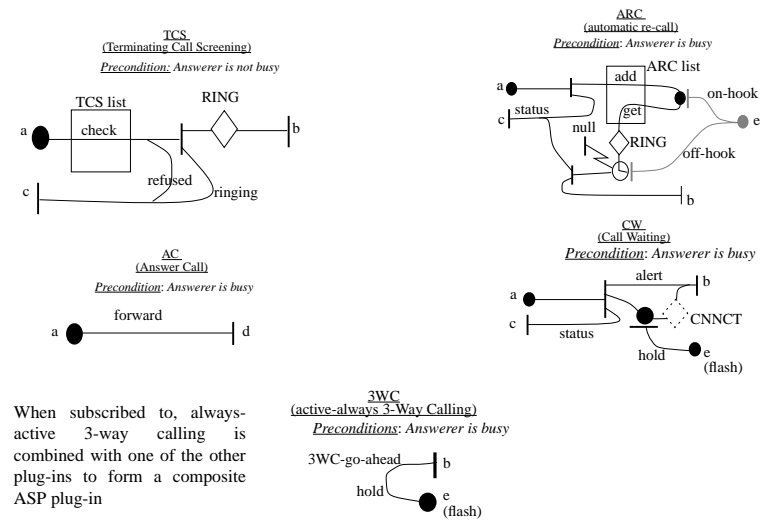
Slide 10

More features



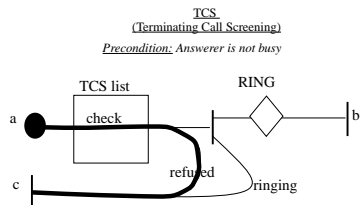
Slide 11

More features

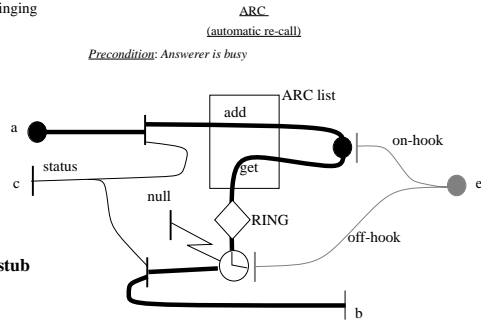


Slide 12

More FI



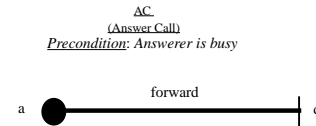
- **inconsistent behaviour between alternative plug-ins that may be selected for the same stub**



Alternative plug-ins for ASP stub in Answer Side Agent

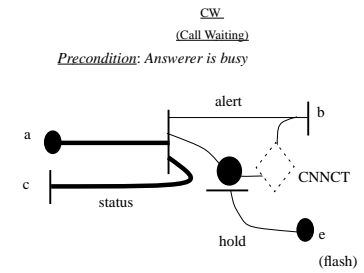
Slide 13

More FI



- **inconsistent behaviour between alternative plug-ins that may be selected for the same stub**

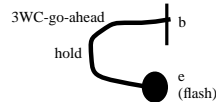
Alternative plug-ins for ASP stub in Answer Side Agent



Slide 14

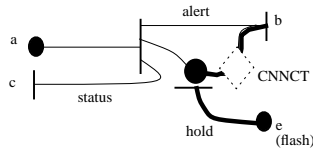
More FI

3WC (active-always 3-Way Calling)
Preconditions: Answerer is busy



- **conflicting interpretation of events between elements of a composite plug-in for the same stub**

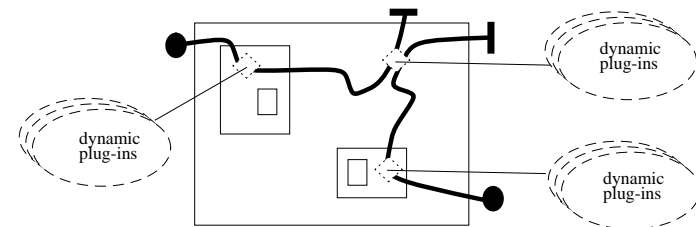
CW (Call Waiting)
Precondition: Answerer is busy



elements of a composite plug-in for ASP stub in Answer Side Agent

Slide 15

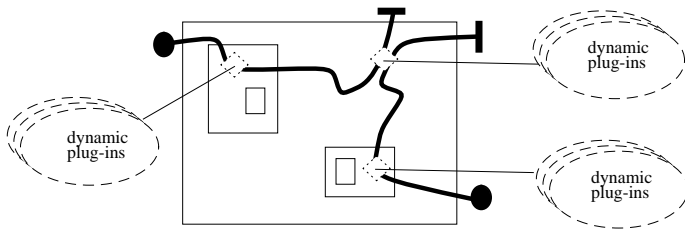
Concept of a system modifying its own "behaviour structures"



- **expressed in a first-class manner**
- **people can understand it**
- **leads to a systematic approach for high level design: "architecting behaviour"**
- **has been exercised on numerous examples of industrial scale and complexity**
- **flows naturally to agent solutions ...**

Slide 16

Visual "behaviour structure" helps visualize problems

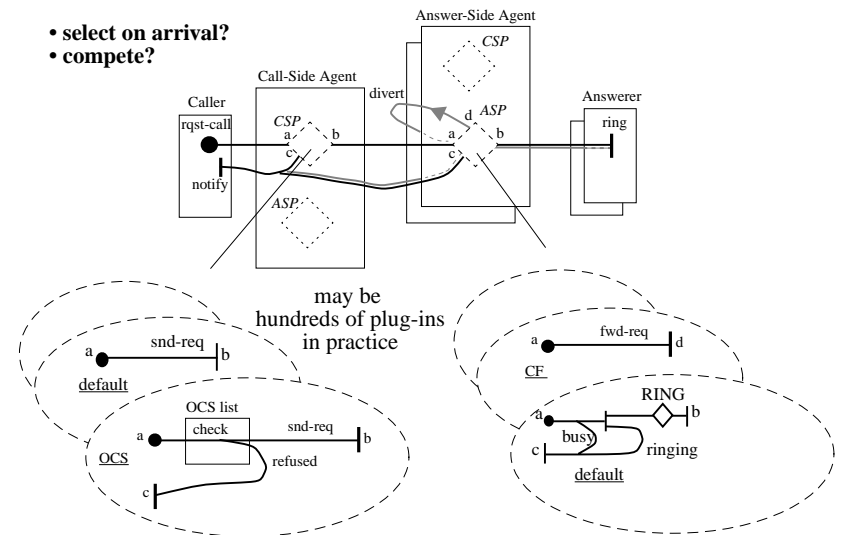


- conflicting interpretations of the same event in different places
- inconsistent behaviour between plug-ins
- unexpected effects of combinations of plug-ins
- ...

Slide 17

Concept of competing feature engines

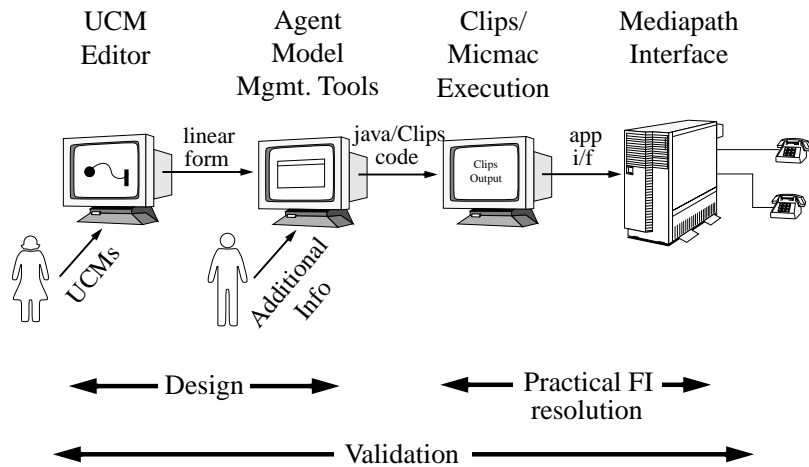
- select on arrival?
- compete?



Slide 18

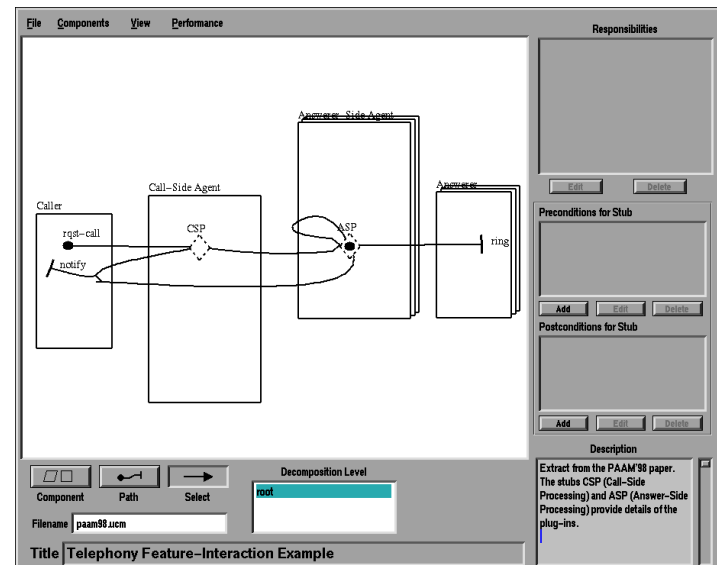
Agent research

www.sce.carleton.ca/rads/agents



Slide 19

UCM editor



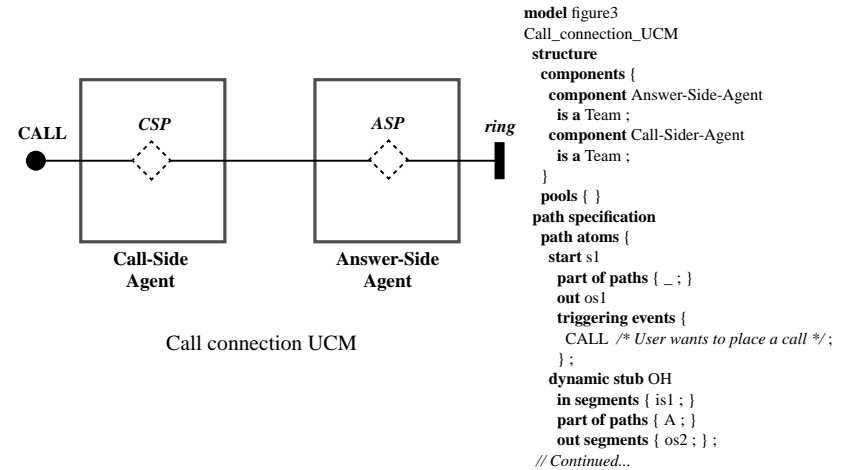
Slide 20

UCM Linear Form

- **Purpose:** textual and parsable representation of UCMs, independent from the UCM Navigator tool and its internal representation.
- It is defined in a grammar by a set of BNF rules (see example below).
- The UCM Navigator tool generates UCM descriptions in linear form (.lfm)
- Linear form descriptions are parsed by *lfpmparse*, a tool written in Java using *JJtree* and *JavaCC*, Sun's Java Compiler Compiler.

Slide21

Linear Form Parser: lfpmparse

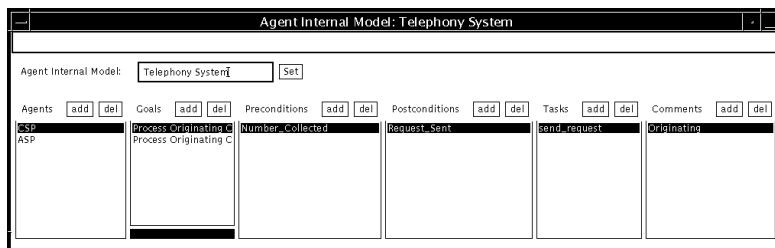


- *lfpmparse* generates a Java object model that is to be used for the partial synthesis of several models (Agent Internal, Agent Relationship, Conversational, and Contract), also implemented in Java.

Slide22

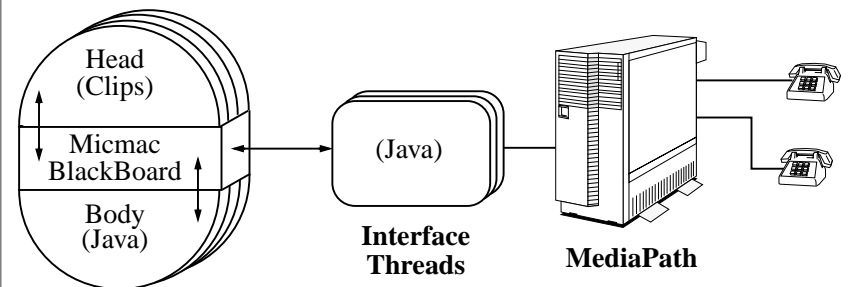
Agent Model Management Tools

- Ongoing prototyping of Java applications for agent models generation and housekeeping.
- Example: *Agent Internal Model*



Slide 23

Prototype Environment Architecture



Coordination between agents and Mediapath through Micmac

Slide 24

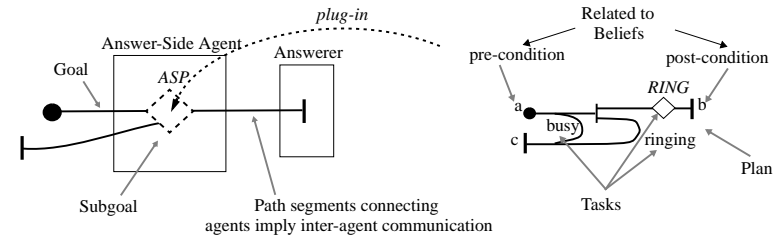
Agent Internal Model

Describes agents in terms of their goals, beliefs and tasks.

	Goal	Precondition	Postcondition	Task	Comment
1	Process originating call	Number is collected	Request sent to answerer	send_request	Originating
2	Process originating call	Outgoing call connection requested	Call permitted or rejected	check_list doPermit doReject	OCS
3	Process call request	There is an incoming call	Caller and/or answerer are notified	ring notify_caller	Terminating
4	Process call request	CF is on. There is an incoming call	Caller notified of a new destination	doForward	CF

Slide 25

From UCMs to Agent Internal Models



UCMs	Agent Terminology	Agent Internal Model
Path segments traverse an agent	Goals	Goal in the goal column
Dynamic stub	Subgoal	Goal in the task column Goal in the goal column
Static stub (decomposition)	Set of agent tasks	Task in the task column
Path segment preconditions	Beliefs	Preconditions column
Path segment postconditions	Beliefs	Postconditions column
Responsibilities	Agent tasks	Tasks in the task column
Path segments connecting two agents	Agent coordination	Tasks in the task column

- Each Plan (row) is to be implemented as a *Clips engine*.
- Goals, preconditions, and postconditions are *facts*
- Tasks are *functions*

Slide 26

Conversational Model (one of several others)

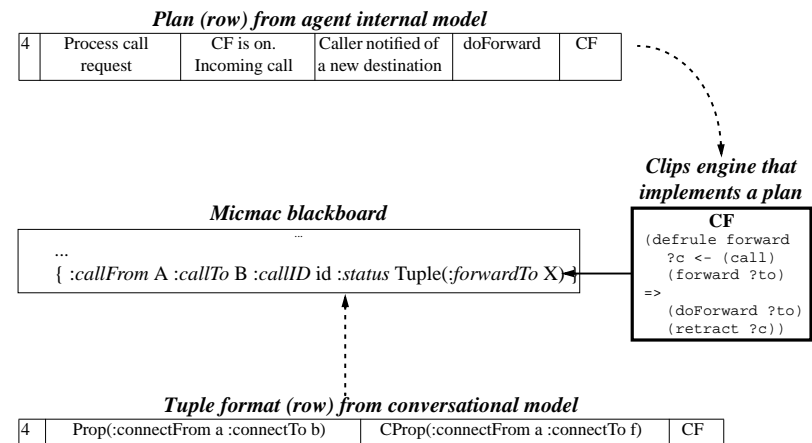
Describes the coordinations among the agents. It basically identifies what messages are exchanged in order for agents to cooperate and negotiate with each other to fulfill inter-agent dependencies.

	Received	Sent	Comment
1		Prop(:connectFrom a :connectTo b)	Originating
2	Prop(:connectFrom a :connectTo b)	ACCEPT REJECT	OCS
3	Prop(:connectFrom a :connectTo b)	ACCEPT REJECT	Terminating
4	Prop(:connectFrom a :connectTo b)	CProp(:connectFrom a :connectTo f)	CF
5	CProp(:connectFrom a :connectTo f)	Prop(:connectFrom a :connectTo f)	Originating

- The conversational model is implemented by tuples.
- Agents exchange tuples using the Micmac blackboard.

Slide 27

From Agent Models to Tuples and Clips Engines



Slide 28

Conclusions

- Telecom systems are a fertile field for agents.
- Telephony FI is an example: New features are added, and conflicts must be resolved, often dynamically; agents offer the prospect of incrementally adapting.
- The dynamic nature of agent solutions presents great difficulties for telecom software designers.
- Contribution 1: UCMs are used to express how a system *dynamically modifies its own large scale "behaviour structures"*; high level designs/prototypes of multi-agent systems flow in a systematic way from this.
- Contribution 2: FI resolution approach that can be systematically designed and implemented, *competing feature engines*.
- Part of an intended approach for future commercial systems.
- Papers address approach and FI; slides give newest view.

Slide 29

Some references

- R.J.A. Buhr
Scenario-Path Signatures as Architectural Entities for Complex Systems
www.sce.carleton.ca/ftp/pub/UseCaseMaps/ucmUpdate.ps
- R.J.A. Buhr, D. Amyot, M. Elammari, D. Quesnel, T. Gray, S. Mankovski
High Level, Multi-Agent Prototypes from a Scenario-Path Notation: A Feature-Interaction Example
www.sce.carleton.ca/ftp/pub/UseCaseMaps/ucmUpdate.ps
- R.J.A. Buhr
Research project description: *High Level Design and Prototyping of Agent Systems*
www.sce.carleton.ca/rads/agents

Slide 30