Finance Case Study

UML Specification and Verification with UMC

Maurice ter Beek

Joint work with
Antonio Bucchiarone, Stefania Gnesi, Franco Mazzanti
ISTI–CNR, Pisa, Italy

SENSORIA workshop
München, 10 February 2009

Outline

Formal verification of SOC applications

Service-Oriented Computing

SOC is an emerging paradigm for developing loosely coupled, interoperable, evolvable systems and applications, exploiting the pervasiveness of the Internet and its related technologies

Goal of our research

Develop formal reasoning mechanisms and analytical tools for checking that the services resulting from a composition meet desirable correctness properties and do not manifest any unexpected behaviour

Credit Portal scenario of Finance case study

I Customer starts credit request by uploading data (name, address, desired amount, revenues, expenses, security values) after which a validation service is invoked: if positive the data is uploaded to the bank, otherwise the customer has to update it

II Bank employee reviews customer's credit request after selecting it from her task list: if positive the employee prepares an offer to the customer, otherwise the credit request is rejected

III a) Bank employee supervisor reviews the offer after having selected it from her task list: if she rejects it the customer is informed and the process ends, otherwise an offer is sent to the customer who can then accept or reject the offer and the process ends

III b) If the bank employee has decided against an offer to the customer, the customer can update his request information (e.g. reduce the credit amount, provide additional security values) and the process is repeated, otherwise the process ends

UML4SOA profile

Extension with stereotypes and constraints specified in OCL

N. Koch, P. Mayer, R. Heckel, L. Gönczy and C. Montangero, UML for Service-Oriented Systems. SENSORIA D1.4a, Sept. 2007

http://www.uml4soa.eu/

UML4SOA is a UML2 profile which contains specialized elements for modelling service interactions, compensation, exception and event handling defined as UML2 stereotypes

Specification and verification of scenario

Specification presented yesterday by Federico Banti

F. Banti, A. Lapadula, R. Pugliese, and F. Tiezzi, Specification and analysis of SOC systems using COWS: A finance case study. Unpublished manuscript, 2008.

http://rap.dsi.unifi.it/cows/papers/finance_case_study_COWS.pdf

Model and analysis presented yesterday by Francesco Tiezzi

Credit Portal scenario specified by a set of UML activity diagrams, using the UML4SOA profile, and subsequently modelled in COWS and analysed by CMC (properties expressed in SocL)

Modelling and analysis of scenario

Today's presentation

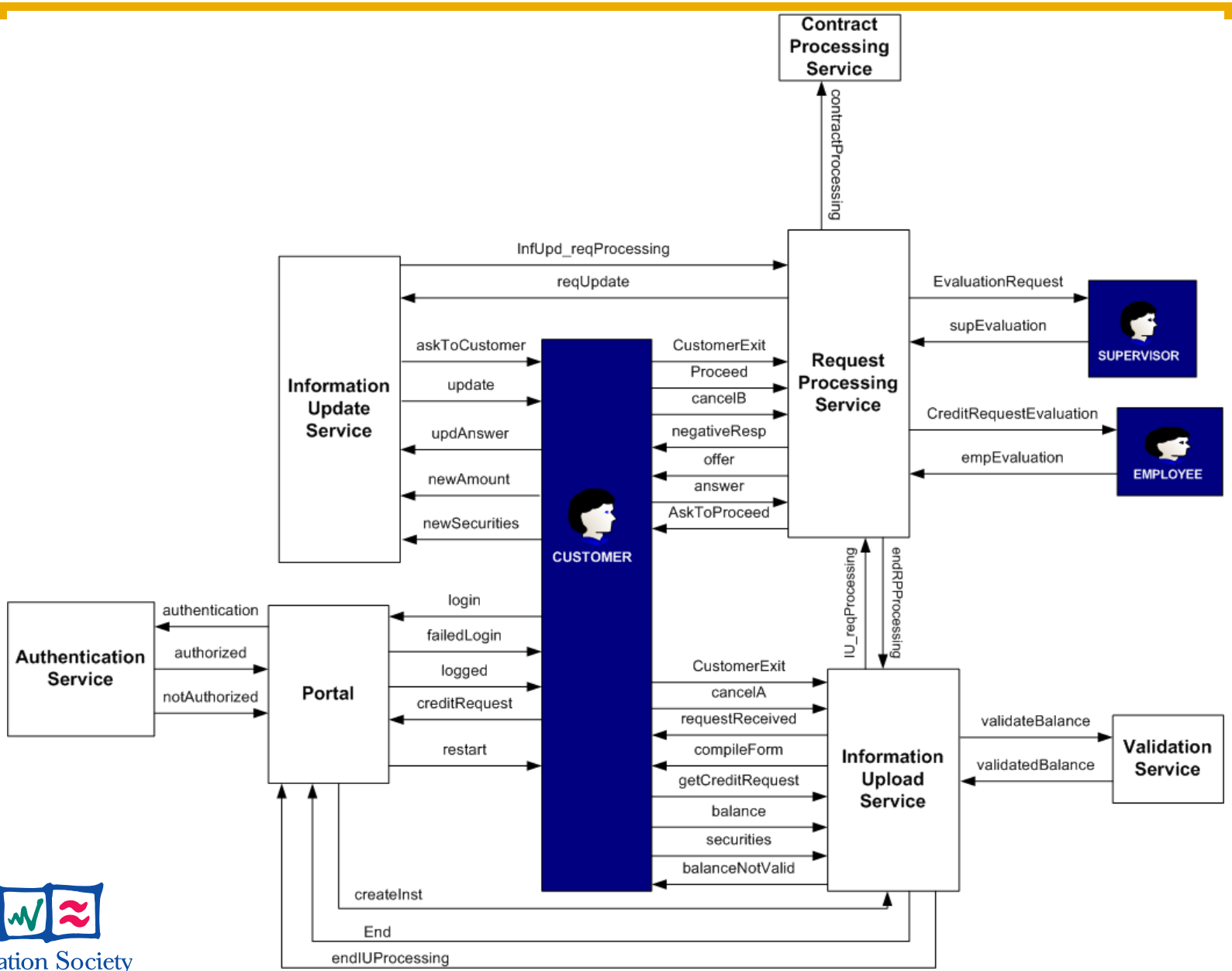
M.H. ter Beek, A. Bucchiarone, S. Gnesi, and F. Mazzanti, UMC Model of a Finance Case Study. Unpublished manuscript, 2008.

http://fmt.isti.cnr.it/WEBPAPER/finance.pdf

Starting from the same UML4SOA specification

Credit Portal scenario modelled as a set of communicating UML state machines and analysed by UMC (properties expressed in SocL)

Communication diagram of UMC model

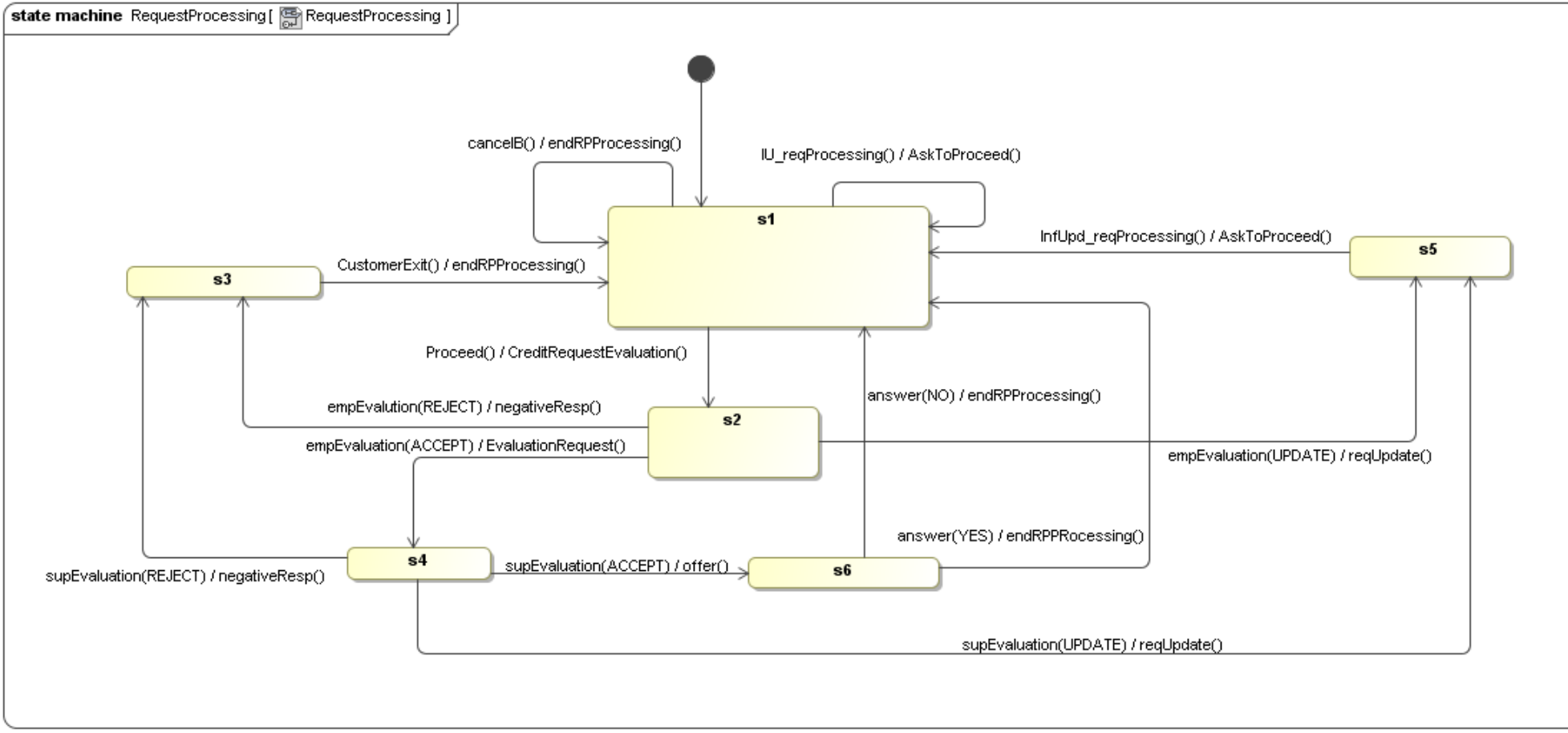


Information Society Technologies

Request Processing UMC code

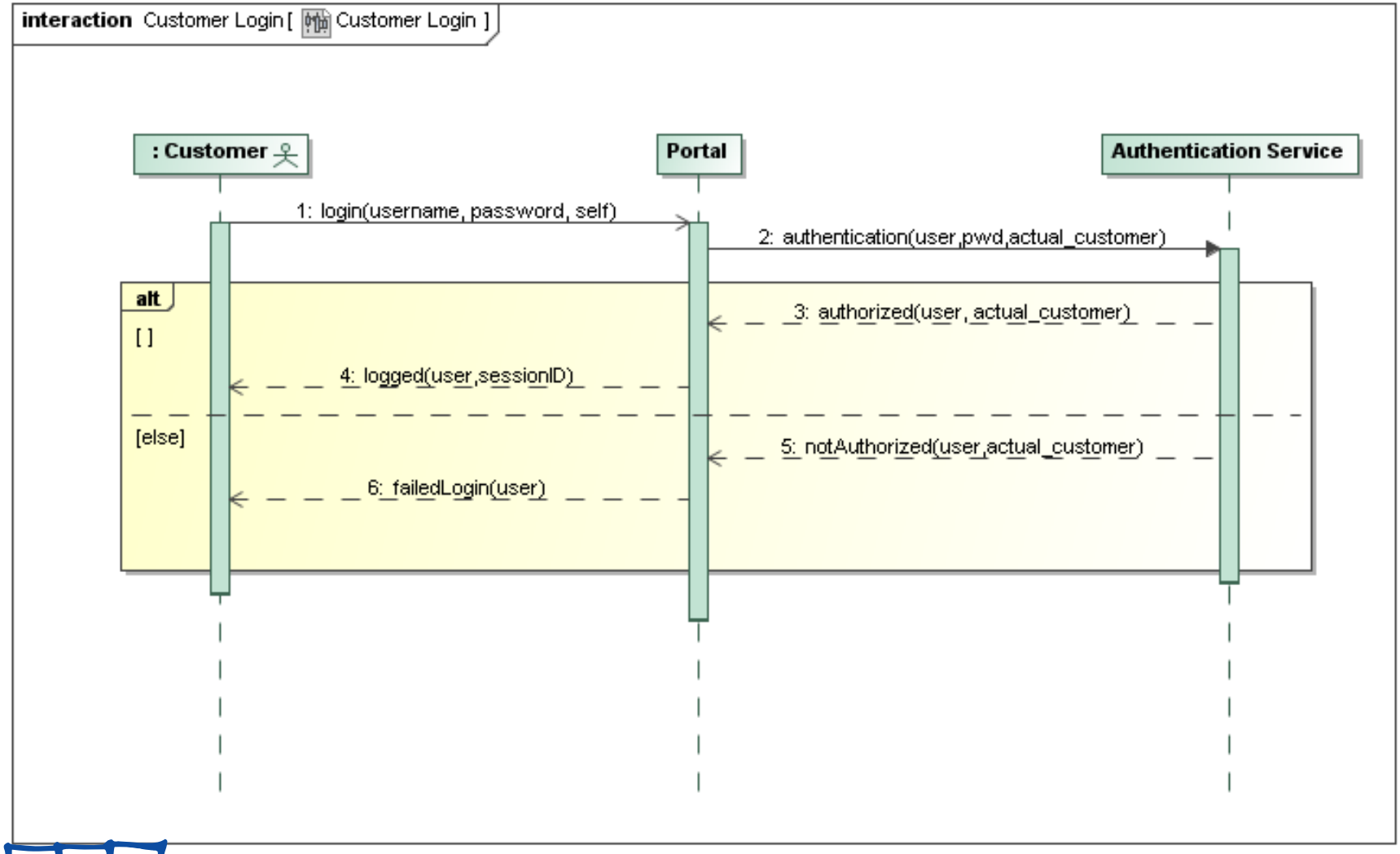

```
Class RequestProcessing is
Signals:
 empEvaluation(SESSION_ID:int,...)          // from employee
 ...
Vars:
 sessionID:int; amount:int; ...
State top = s1,s2,s3,s4,s5,s6
Transitions:
 s2 -> s4      // positive empEvaluation (fwd to supervisor)
       {empEvaluation(SESSION_ID,...)[RESULT==ACCEPT] /
       sessionID := SESSION_ID;...;
       mySupervisor.EvaluationRequest(sessionID,...);
       }
...
end RequestProcessing;
```

Request Processing state machine



Information Society Technologies

Customer login sequence diagram



Comparison of COWS and UMC models

- The actors *Employee* and *Supervisor* of the COWS model have associated lists of the tasks to validate: Due to the natural way in which UMC uses FIFO queues to deal with messages in their order of arrival, the UMC model does not need to make us of such auxiliary objects
- Communications in COWS are modelled as synchronous pairs of send-receive actions, whereas communications in UMC are modelled as two separate actions (sending and storing into a FIFO queue, and removing and dispatching from that queue)
- This obviously contributes significantly to the fact that a UMC model usually has a much higher number of states than a COWS model of the same scenario

UCTL: Action- and state-based logic

The emerging trend in industry is to use UML (state diagrams)

To use the full potential of specification languages (like UML) that allow both action and state changes to be modelled, various action- and state-based logics have been developed, allowing one to express in a natural way not only properties of evolution steps (i.e. related to the executed actions) but also internal properties of states (e.g. the values of object attributes)

UCTL: Action- and state-based branching-time temporal logic

M.H. ter Beek, A. Fantechi, S. Gnesi and F. Mazzanti, An action/state-based model-checking approach for the analysis of communication protocols for Service-Oriented Applications. In *FMICS'07*, LNCS 4916, 2008, 133–148.

From UCTL to SocL

- UCTL defined as extension of action-based CTL (ACTL)
- UCTL includes both CTL and ACTL
- semantic domain *Doubly-labelled Transition Systems* (L^2TS)

SocL: Service-oriented specialization of UCTL

A. Fantechi, S. Gnesi, A. Lapadula, F. Mazzanti, R. Pugliese, and F. Tiezzi, A model checking approach for verifying COWS specifications. In *FASE'08*, LNCS 4961, Springer, 2008, 230–245.

- Observable actions are of the form $t(i, c)$
  - t: kind of interaction (i.e. acceptance of a request, a reply, etc.)
  - i: name of interaction (i.e. name of operation exposed by service)
  - c: tuple of data values (correlation values) identifying a specific activation of a service operation (often denoting a *session*)

CMC-UMC on-the-fly model checkers

A uniform verification framework for abstract SOC properties

Two instantiations of a common logical verification framework for the analysis of functional properties of service-oriented systems

http://fmt.isti.cnr.it/{u/c}mc

The service-oriented SocL logic is used to describe the required system properties (using the explicit abstraction mechanism)

CMC (v0.6) and UMC (v3.6) share the same roots (and part of the actual code) but differ in the underlying computational model

Compare the two models w.r.t. their possible abstract traces

Minimized versions preserving the equivalence w.r.t. set of full traces

CMC versus UMC

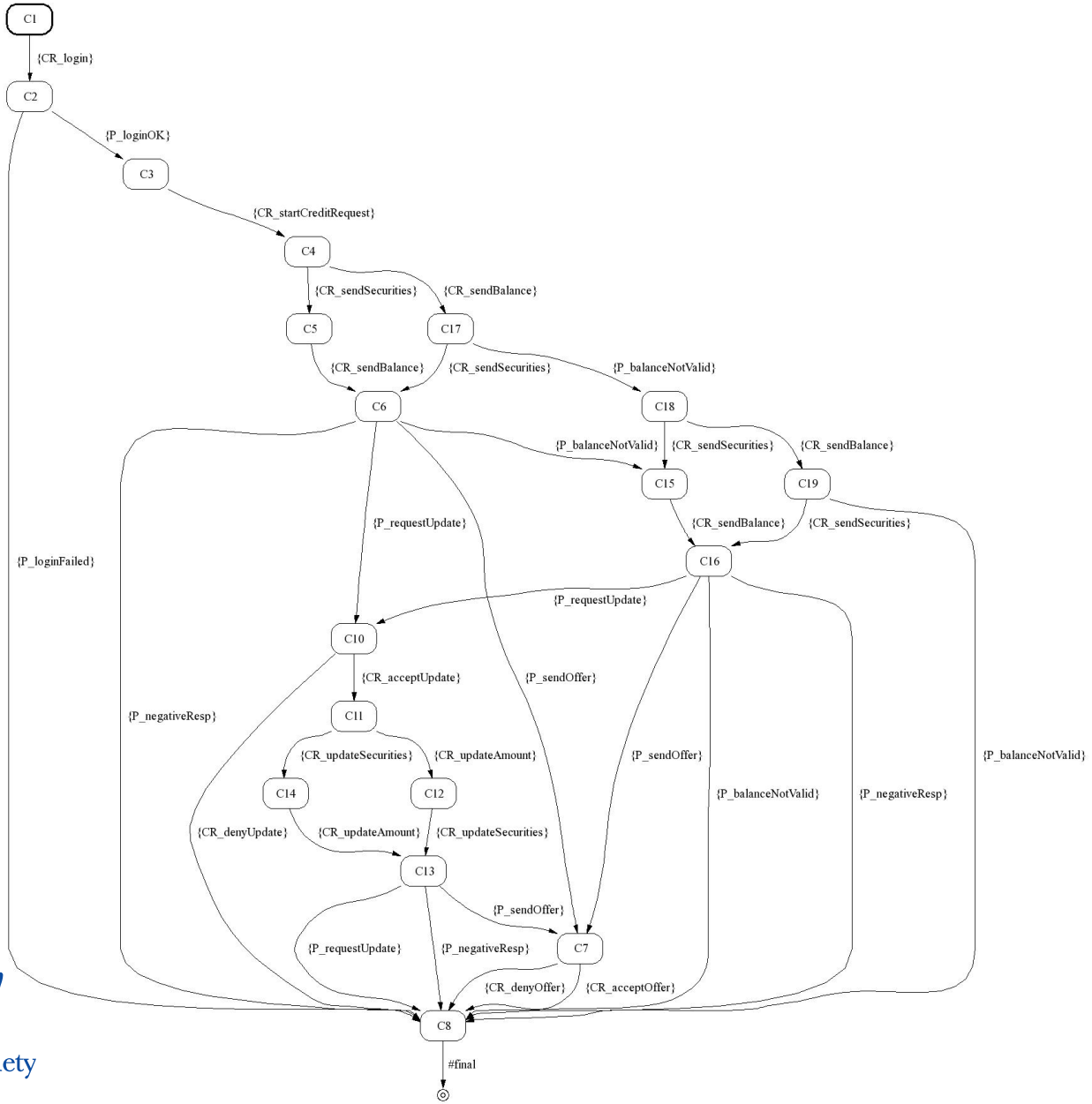
M.H. ter Beek, S. Gnesi and F. Mazzanti, CMC-UMC: A Framework for the Verification of Abstract Service-Oriented Properties. To appear in *Proc. SAC'09—SOAP track*, ACM, 2009, 1844–1850.

Computational models of a system can be built either using COWS or designing it as a collection of interacting UML state machines, and an on-the-fly model checker can be used to verify the requirements and possibly generate counterexamples

- COWS is a richer specification language
  - delimitation operator can generate fresh names
  - replication operator permits to spawn in parallel
- UMC has lower memory requirements

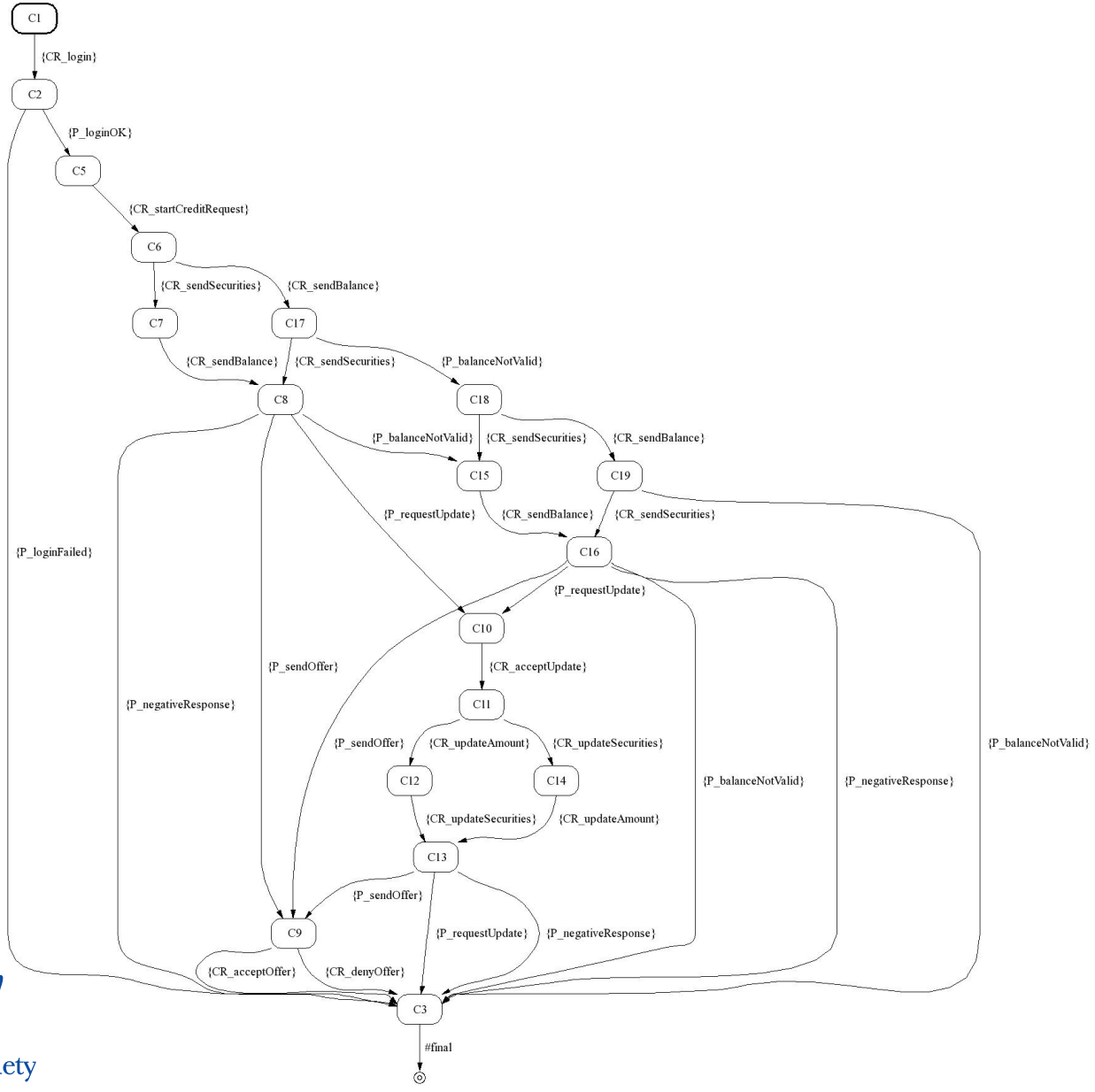
CMC v0.6 can handle state spaces of some 6000 states
UMC v3.6 can handle state spaces upto 1000000 states

UMC: Minimized abstracted evolution of *Customer* behaviour (no cancel)



Information Society Technologies

CMC: Minimized abstracted evolution of *Customer* behaviour (no cancel)



Information Society Technologies

Service Responsiveness: TRUE

A service is *responsive* if it guarantees a response to each received request

Whenever a *Customer* requests for a credit, the *CreditPortal* will always eventually answer (either positive, sending an offer, or negative)

SocL property (as presented yesterday by Francesco Tiezzi)

$$AG\ [\ request(cr, \$id)\ ]$$
$$AF\ \{\ response(cr, \%id)\ or\ fail(cr, \%id)\ \}\ true$$

(request and answer are correlated by variable *id*)

Future Work

- Adapt UMC model to new UML4SOA specification (if needed) of Credit Portal v3 (as presented yesterday by S&N)
- Further analyze UMC model: express a number of significant (service-oriented) properties in SocL and verify them with the associated CMC-UMC model-checking framework
- This will also allow a more detailed comparison of the two underlying specification approaches (joint work planned with Federico Banti, Francesco Tiezzi and Rosario Pugliese)
- Continuous improvement of the model checkers