

Using formal methods without knowing them

Francesco Tiezzi

DSIUF, Università degli Studi di Firenze

Sensoria Final Review
Munich — February 23, 2010



Enable verification of properties of service-oriented systems modelled by graphical (UML-based) modelling languages ...



Enable verification of properties of service-oriented systems modelled by graphical (UML-based) modelling languages ...
... through the use of **formal methods**



Enable verification of properties of service-oriented systems modelled by graphical (UML-based) modelling languages ...
... through the use of **formal methods**

Formal methods

- Process calculi: COWS
- Temporal logics: SocL
- Model checking: CMC



Enable verification of properties of service-oriented systems modelled by graphical (UML-based) modelling languages ...
... through the use of **formal methods**

Formal methods

- Process calculi: COWS
- Temporal logics: SocL
- Model checking: CMC

UML is quite well known and widely used within industrial contexts, ...



Enable verification of properties of service-oriented systems modelled by graphical (UML-based) modelling languages ...
... through the use of **formal methods**

Formal methods

- Process calculi: COWS
- Temporal logics: SocL
- Model checking: CMC

UML is quite well known and widely used within industrial contexts, the same cannot be said of the formal methods



Venus: a **V**erification **E**nvironment for **U**ML models of **S**ervices

A software environment for verifying behavioural properties of UML models of services by exploiting process calculi and temporal logics



Venus: a **V**erification **E**nvironment for **U**ML models of **S**ervices

A software environment for verifying behavioural properties of **UML models of services** by exploiting process calculi and temporal logics

- UML models of services: UML4SOA activity diagrams

Venus: a **V**erification **E**nvironment for **U**ML models of **S**ervices

A software environment for verifying **behavioural properties** of UML models of services by exploiting process calculi and temporal logics

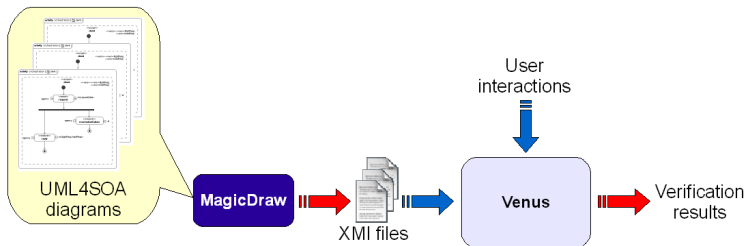
- UML models of services: UML4SOA activity diagrams
- Venus shepherds the (non-expert) users to set the behavioural service properties they want to verify

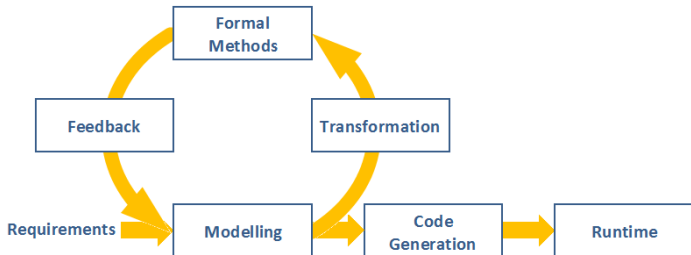


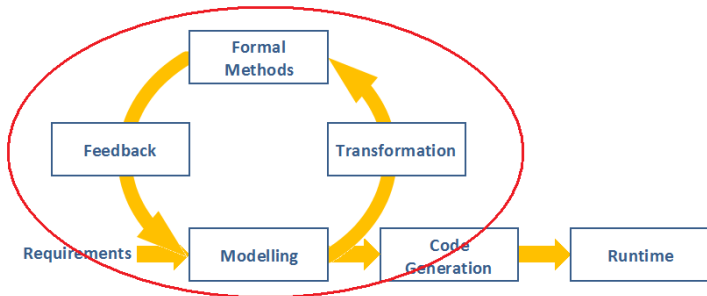
Venus: a **V**erification **E**nvironment for **U**ML models of **S**ervices

A **software environment** for verifying behavioural properties of UML models of services by exploiting process calculi and temporal logics

- UML models of services: UML4SOA activity diagrams
- Venus shepherds the (non-expert) users to set the behavioural service properties they want to verify
- It is a proof-of-concept implementation







The considered service provides a customer company with the possibility to ask for a loan to a bank and then orchestrates the necessary steps for processing the credit request

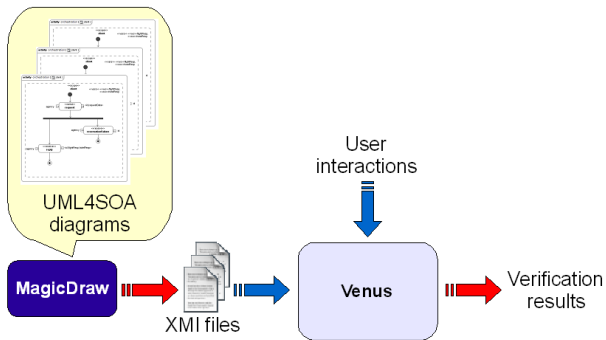
The considered excerpt

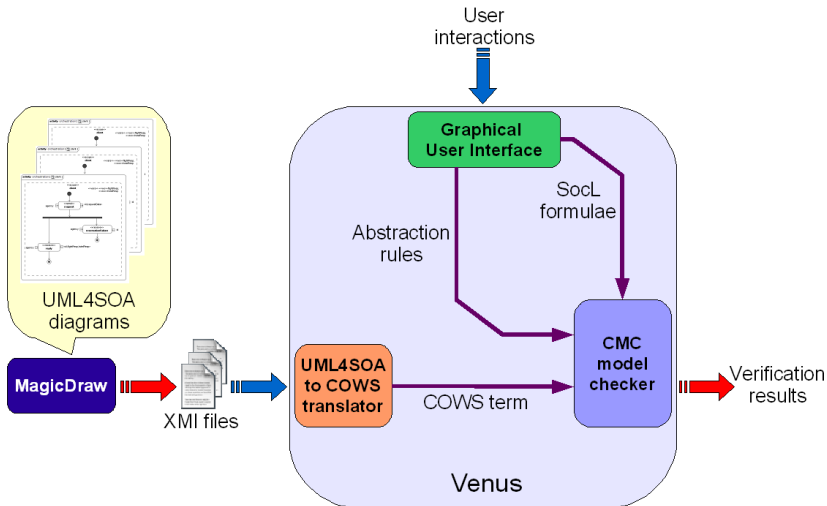
- Initially, the customer logs in to the credit request service by providing his username and password, then uploads the necessary data for his request
- At any moment the customer may require to abort the process

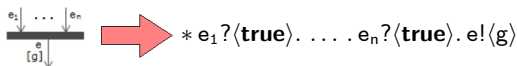
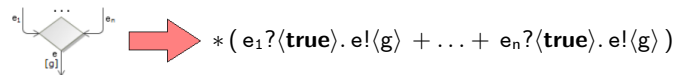
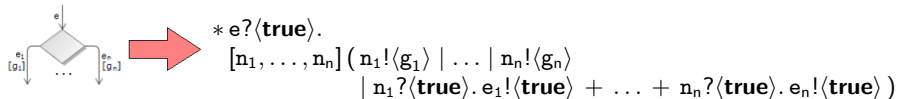
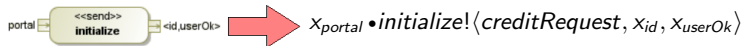


Tool demonstration . . .









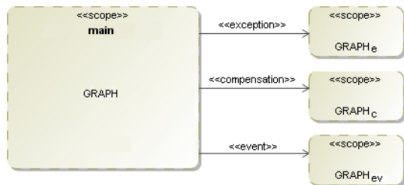


$[r, stack]$
 $([k] (GRAPH ; \{c \bullet main? \langle \rangle . GRAPH_c \}$
 $| \{Stack\} | * GRAPH_{ev})$
 $| r? \langle \rangle . \{GRAPH_e \})$

«compensate»
main $\rightarrow c \bullet main! \langle \rangle$

«raise»
abort $\rightarrow kill(k) | \{r! \langle \rangle \}$

«compensate all» $\rightarrow stack \bullet compAll! \langle \rangle$



$[r, stack]$
 $([k] (GRAPH; \{c \bullet main?\langle \rangle\}. GRAPH_c \}$
 $\quad | \{Stack\} | * GRAPH_{ev})$
 $\quad | r?\langle \rangle. \{GRAPH_e \})$

«compensate» main $\rightarrow c \bullet main!\langle \rangle$

«raise» abort $\rightarrow kill(k) | \{r!\langle \rangle\}$

«compensate all» $\rightarrow stack \bullet compAll!\langle \rangle$

Our COWS implementation of UML4SOA constructs follows a compositional approach



- Venus aims at enabling verification of UML4SOA models of services

- Venus has been explicitly developed for being accessible by users not familiar with formal methods

- Venus theoretical bases are three SENSORIA outcomes:
 - the process calculus COWS
 - the temporal logic SocL
 - the model checker CMC



For further details visit



<http://rap.dsi.unifi.it/cows/>

