SENSORIA Development Environment

The SENSORIA Development Environment aims to support the development of service-oriented applications by integrating the various tools developed as part of the Sensoria project as well as external tools to support the design, development, analysis, and deployment of service-oriented systems.

The SENSORIA Case Tool (SCT) provides the basis for the SENSORIA Development Environment. Its main goal is to provide a service-oriented platform for tool integration and orchestration where

- tools are services, and provide arbitrary functionality
- tools can be used as-is, or combined using orchestration mechanisms
- tools can be published and discovered

The SENSORIA Developement Environment (SDE) can be downloaded on our web site, http://www.sensoria-ist.eu. Links to various integrated tools are provided there as well.

Examples of Integrated Tools

VIATRA2
VIATRA2 (an Eclipse GMT project) designs model transformations to support the precise model-based systems development with the help of invisible formal methods. Formal methods are hidden by automated model transformations projecting system models into various mathematical domains (and, preferably, vice versa).
http://www.eclipse.org/gmt/

WS-Engineer
The LTSA WS-Engineer plug-in allows service models to be described by translation of the service process descriptions, and can be used to perform model-based engineering.
http://www.doc.ic.ac.uk/ltsa/eclipse/wsengineer/

PEPA
PEPA is a stochastic process algebra which is used for modelling systems composed of concurrently active components which co-operate and share work. PEPA allows the modeller to study either behavioural or performance properties.
http://www.dcs.ed.ac.uk/pepa/

SENSORIA Development Environment

Main innovative features of the SENSORIA Development Environment are:

- precise model-driven development process for service-oriented applications
- use of integrated, hidden mathematical methods to improve the quality of the overall service
- service-oriented extension of standard visual modelling techniques
- tool integration by model transformations
- deployment process which targets popular service-oriented platforms

Main envisaged impact on the market:

- reduced time-to-market of the integration of services
- increased quality of service (measured in both qualitative and quantitative terms)
- portability of SENSORIA methods to existing platforms helping to maintain highly profitable services

List of Partners

Ludwig-Maximilians-Universität München, Università di Trento, University of Leicester, Warsaw University, TU Denmark at Lyngby, Università di Pisa, Università di Firenze, Università di Bologna, ISTI Pisa, Universidade de Lisboa, University of Edinburgh, ATX Software SA, Telecom Italia Lab, Imperial College London, Cirquent (FAST) GmbH, Budapest University of Technology and Economics, S&N AG, University College London, Politecnico di Milano

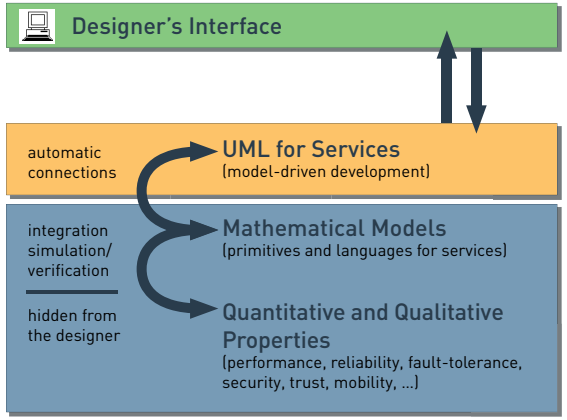
Project Coordinator
Ludwig-Maximilians-Universität München
Institute for Informatics

Prof. Dr. Martin Wirsing
info@sensoria-ist.eu
Tel +49 89 2180 9154
Fax +49 89 2180 9175

Development Environment for SOA

http://www.sensoria-ist.eu

The SENSORIA project provides an approach for the visual design, formal analysis, and automated deployment of software based on the Service-Oriented Architecture (SOA).



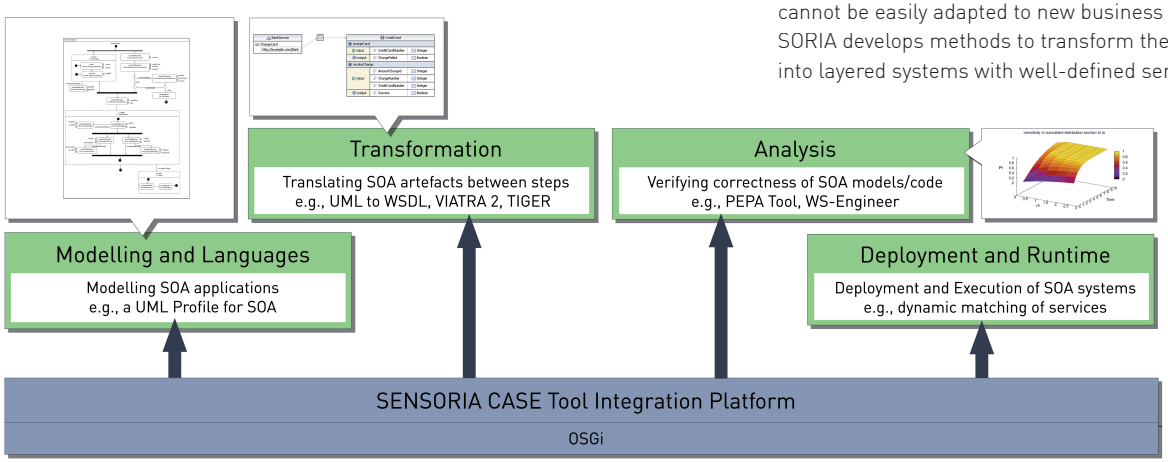
The SENSORIA project provides a number of tools for developing SOA applications. These tools are integrated in the SENSORIA Development Environment which is freely available to both researchers and practitioners.

The highly innovative nature of the SENSORIA approach originates from its mathematical foundations and automated model-transformations.

SENSORIA Development approach

SENSORIA provides a comprehensive approach for precisely architecting service-oriented systems by applying hidden formal methods.

- Modelling front-end. Service-oriented applications are designed using high-level visual formalisms such as the industry standard UML or domain-specific modelling languages to precisely capture domain-specific requirements.
- Hidden formal analysis of services. Back-end mathematical model analysis is used to reveal performance bottlenecks or interactions leading to errors or violation of service contracts. For critical services we envisage deep semantic analysis for certification.
- Automated model transformations. Formal representations are generated by automated model transformations from engineering models.
- Service deployment. As a result, service models of proven quality serve as the basis for deployment transformations to generate configurations for standards-compliant service platforms.
- Orchestration and tool integration. The SENSORIA Development Environment provides an Eclipse-based fully-customizable tool chain for the entire model-driven workflow (development, analysis and deployment etc.).
- Reengineering of legacy services. Many existing systems are built as monolithic non-extensible applications which cannot be easily adapted to new business processes. SENSORIA develops methods to transform these applications into layered systems with well-defined service interfaces.



Scenario for analyzing services

Modern automobiles are equipped with many electronically controllable sensors and devices. In the near future, vehicles will communicate with each other and a background infrastructure in order to provide services such as infotainment, repair assistance, road planning, etc.

Typical analysis questions

By using the SENSORIA approach we aim to provide justified answers for questions like:

- Is it true that the credit card of the driver won't be charged if there are no available garages?
- Will a tow truck be sent within 15 minutes?
- Will the driver have a proper car ordered automatically?

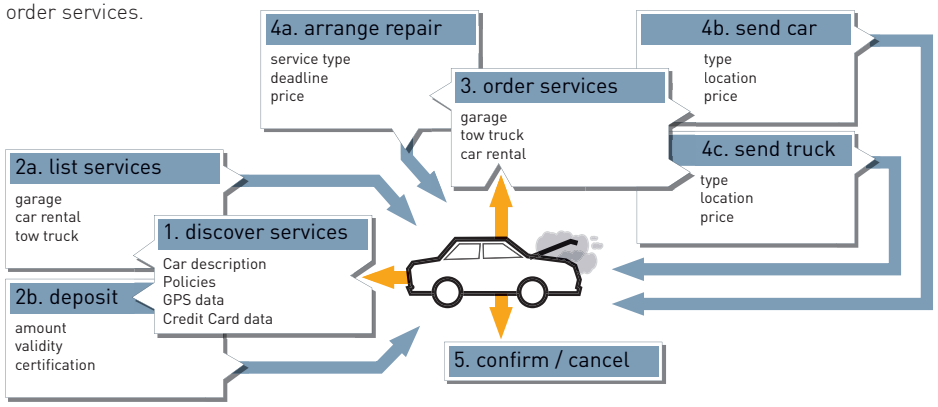
Analysis answers

SENSORIA tools extend standard design workflow by providing justifiable answers:

- No deposit will be charged if service is cancelled.
- The car will arrive within 10 minutes with 90% probability.
- The driver will get everything according to his policy if payment is confirmed.

An OnRoadAssistance scenario:

1. The diagnostic system reports a severe failure in the car engine so that the car is no longer drivable.
2. The car's discovery system identifies garages, car rentals and towing truck services in the car's vicinity.
3. The in-vehicle service platform selects and orders a set of adequate offers taking into account personalised policies and preferences of the driver.
4. The owner of the car has to deposit a security payment before being able to order services.



Design and analysis workflow

1. The high-level system model is designed using UML4SOA profile developed in SENSORIA.
2. Derivation of standard BPEL description.
3. Correctness analysis in WS-Engineer reveals that it is not possible that the credit card is charged and no tow truck arrives (this step can also be performed on a BPEL model from any industrial tool).
4. Performance analysis in PEPA answer quantitative questions, e.g. "90% of all assistance requests will be served within 2 minutes" or "The GPS location service is a quality bottleneck in the system."
5. Standard deployment descriptors (such as WSDL) are generated by the VIATRA2 model transformation framework.
6. Tools are integrated in the SENSORIA Case Tool.