SENSORIA

A novel engineering approach to service-oriented computing

www.sensoria-ist.eu

Sensoria meeting — Lucca, Italy — November 2006

SENSORIA

A novel engineering approach to service-oriented computing

Martin Wirsing
Coordinator
wirsing@ifi.lmu.de

Ludwig-Maximilians-Universität München

www.sensoria-ist.eu

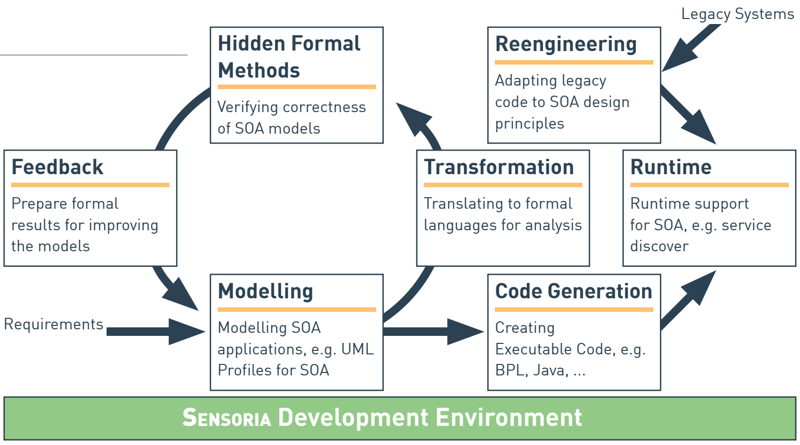
Service-Oriented Computing (SOC) is of enormous strategic importance for society and a key enabling technology for IT-rich areas of modern life. Several of the key concepts of Service-Oriented Computing (SOC) are directly connected to economic benefits to IT-dependent organizations. Moving to SOC saves investments, as existing software may be wrapped and thus made available in a more decoupled way.

By using services as the basic computational entities, developers can grow new applications by merely combining existing services rather than having to create a new monolithic application from scratch, while at the same time keeping the system loosely coupled. Also, the flexible nature of service interactions allows organizations to quickly react to changes in its environment by replacing services or re-configuring/re-organizing service compositions.

The IST-FET Integrated Project **SENSORIA** has developed a novel comprehensive approach to deal with SOC where foundational theories, techniques and methods are fully integrated in a pragmatic tool-supported software engineering approach, addressing, for example, early verification and validation, semi-automatic development and deployment of self-adaptable (composite) services.

The **SENSORIA** techniques enable service engineers to model their applications on a very high level of abstraction using service-oriented extensions of the standard UML, or domain-specific service-oriented modelling languages to translate into hidden formal representations by automated model transformations as well as generate executable code. Our tools are able to perform checks of functional correctness of services, early performance analysis, prediction of quantitative bottlenecks in collaborating services, and verification of service level agreements. Finally, results of the mathematical analysis are made available in the models to provide feedback to the engineer.

SENSORIA development process support


Legacy Systems
Hidden Formal Methods
Verifying correctness of SOA models
Reengineering
Adapting legacy code to SOA design principles
Feedback
Prepare formal results for improving the models
Transformation
Translating to formal languages for analysis
Runtime
Runtime support for SOA, e.g. service discover
Requirements
Modelling
Modelling SOA applications, e.g. UML Profiles for SOA
Code Generation
Creating Executable Code, e.g. BPL, Java, ...
SENSORIA Development Environment

Modelling Languages

SRML - Sensoria Reference Modelling Language

University of Leicester

Universidade de Lisboa

José L. Fiadeiro
jose@mcs.le.ac.uk

www.sensoria-ist.eu/srml

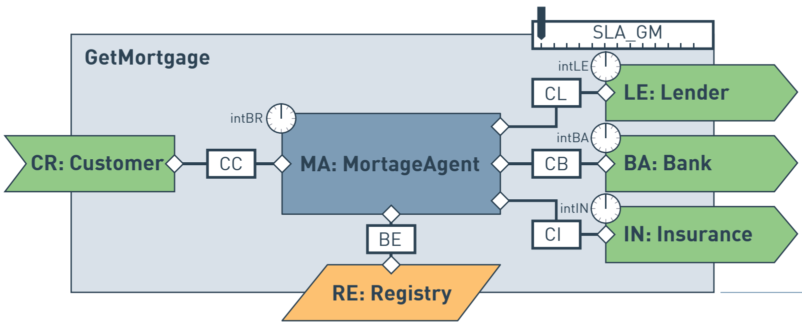
The prototype language SRML (for **Sensoria** Reference Modelling Language) operates at the higher levels of abstraction of business modelling, i.e. it provides a number of semantic modelling primitives for service-oriented systems that are independent of the languages and platforms in which services are programmed and executed.

In SRML, the orchestration of services is expressed in terms of a number of internal and external parties that are connected to each other through interaction protocols and jointly execute a (distributed) business process. The configuration of this business process may change at run time as the discovery of required services is triggered.

A formal computation and coordination model offers a layer of abstraction for capturing, orchestrating and analysing properties of the conversational protocols that characterise service-oriented interactions. In SRML, properties of required and provided services are specified in temporal logic and can be analysed over orchestrations defined in terms of state transition systems using the UMC model checker. Time-related properties of services can be analysed using the Markovian process algebra PEPA.

An algebraic operational semantics supports the run-time discovery, selection and binding mechanisms of the language and offers a business-reflective model of dynamic (re)configuration. SLA constraints and the associated ranking and selection mechanisms are formalised over the c-semiring approach to constraint optimisation.

Finally, extensions of use-case and message-sequence diagrams provide support for a number of methodological aspects of engineering business services and activities.



SRML service representation

Ludwig-Maximilians-Universität München

Budapest University of Technology and Economics

Università di Pisa

London Software Systems

■ **Imperial College London**

UML Family of Profiles for SOC

The Unified Modeling Language (UML) is accepted as lingua franca in the development of software systems. However, standard UML does not contain specific support for SOA systems. Based on existing extensions such as the upcoming OMG standard SoaML, we have added advanced SOA support to the UML by means of domain-specific modelling extensions in the form of profiles.

The **Sensoria** family of UML profiles for SOA consists of five extensions addressing different aspects of a SOA system: UML4SOA, UML4SOA-NFP, Business Policies Support, Service Modes and Service Deployment. They focus on behavioural aspects of service-oriented software such as message passing among requester and provider of services, compensation of long-running transactions, modes, and policies associated to services. The support of these service concepts in the modelling language avoids diagrams overloaded with technical constructs and improves the readability of the models.

UML profiles and their corresponding metamodels constitute the basis for model transformations and code generation defining a model-driven development process. In particular, the MDD4SOA (Model Driven Development for SOA) transformers also developed within the scope of the SENSORIA project are model transformations implemented as Eclipse plug-ins. They automatically transform service orchestrations specified with our UML4SOA profile to executable code, such as BPEL/WSDL, Java and JOLIE.

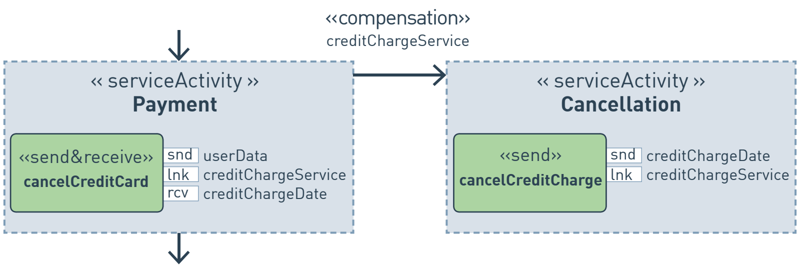
Ludwig-Maximilians-Universität München

Philip Mayer
mayer@pst.ifi.lmu.de

www.sensoria-ist.eu/uml4soa

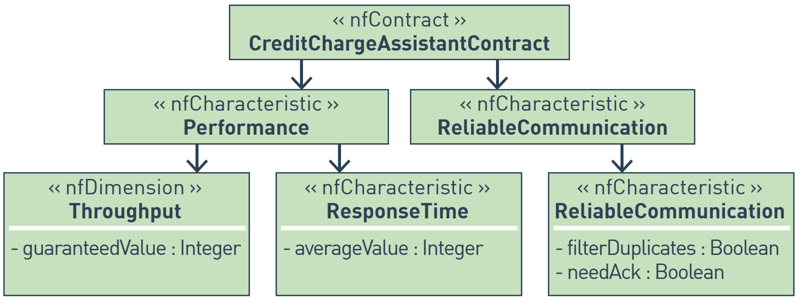
UML4SOA

UML4SOA. This profile supports modelling orchestration of services. Orchestration is the ability to compose existing services creating a description of the interaction of several services supporting constructs such as message passing, addressing partner services, compensation and event handling.



Modelling compensation

UML4SOA-NFP. This profile aims at the modelling of arbitrary "quality of service" properties defined for a particular given client-server pair. Since in real service configurations, service properties can vary for different classes of clients, we follow a contract-based approach, where non-functional properties of services are defined between two participant components: the service provider and the service requester.



Budapest University of Technology and Economics

Lázló Gönczy
gonczy@mit.bme.hu

Modelling non-functional properties

Business Policies Support. This profile deals with the connection of services and business policies, in the context of StPowla. The goal of STPOWLA is to define the business process so that the business stakeholder can easily adapt it to the current state of affairs, by controlling the resources used by the basic tasks in the workflows. To this purpose, the stakeholders issue policy definitions, which constrain the resource usage as a function of the state of the workflow when a task is needed.

Università di Pisa

Carlo Montangero
monta@di.unipi.it

Service Modes. This profile supports modelling of adaptive service brokering offering SOA architectural artefacts, which are an abstraction of a specific set of services that must interact for the completion of a specific subsystem task. A mode will determine the structural constraints that rule a (sub)system configuration at runtime.
Service Deployment. This profile supports modelling deployment architecture nodes (Servlet, WebServer) and deployment artefacts (ServiceOrchestration and Resource).

London Software Systems
Imperial College London

Howard Foster
howard.foster@imperial.ac.uk

JOLIE – Java Orchestration Language Interpreter Engine

JOLIE is a language for dealing with Service-oriented Architecture (SOA) programming, which allows for the modelling of both services (e.g. Web Services) and orchestrators. It supports the design, development and deployment of services. Its language is easy and intuitive and supports interoperability issues. It provides embedded standard protocols such as HTTP and SOAP and can be easily enhanced for interoperating with different platforms by exploiting open or closed/proprietary technologies.

JOLIE can also be used for updating legacy software infrastructures into SOAs, building new systems by exploiting the distributed modularity offered by the service-oriented paradigm. JOLIE is an open-source project developed in Java.

Università di Bologna

ItalianaSoftware

Claudio Guidi
cguidi@cs.unibo.it

www.sensoria-ist.eu/jolie

A Formal Foundation for Service-Oriented Computing

Università di Firenze

Università di Pisa

Università di Bologna

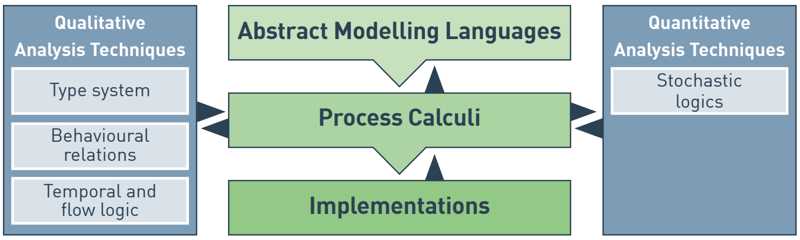
Universidade de Lisboa

Rocco De Nicola
rocco.denicola@unifi.it

Ugo Montanari
ugo@di.unipi.it

Process calculi and the Sensoria approach

The calculi for service specification and analysis play a central role in **Sensoria**. They are needed both to describe, discover and compose systems and to prove that their behaviour is consistent with the expectation of the designer. They have been used as target of services specifications provided in abstract modelling languages like SRML or UML4SOA but have been also equipped with tools for both qualitative and quantitative analysis and have been the basis for defining abstract machine that guarantee provably correct implementations of services.



Central role of process calculi in the Sensoria approach

Central role of process calculi in the Sensoria approach

For the qualitative analysis of services, we have:

- equipped process calculi with type systems that enable checking conformance of services with contracts and deadlock freedom of the outcome of the composition of given services;
- introduced temporal logics that permits naturally expressing typical service-oriented properties like session data correlation, service availability, service responsiveness, and by means of an on-the fly verification engine, the verification of the specified properties;
- used flow logics to statically compute over-approximations of services behaviour and guarantee expected properties;
- introduced behavioural relations the permit studying the relationships between concrete descriptions of services (close to the implementation) and abstract descriptions (providing the specification).

For the quantitative analysis of services, we have:

- defined stochastic variants of the calculi that allows the description of services whose basic actions are paired with a rate expressing their frequency and enables us to analyze services by using markovian techniques and model checking of stochastic logic.
- defined a service-oriented stochastic logic - a probabilistic, timed, temporal logic, that is resource oriented to address open-endedness of service oriented computing.

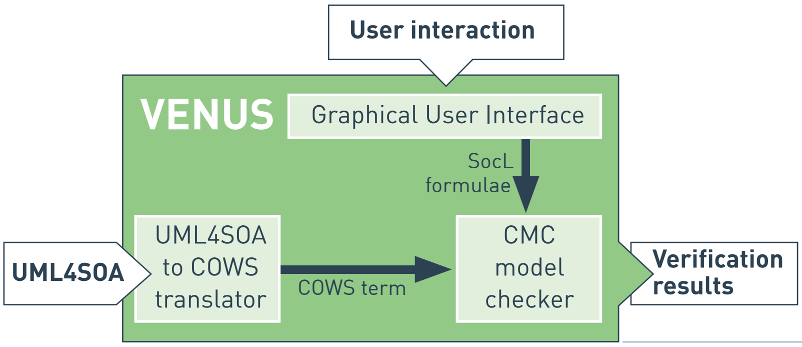
Verification Enviroment for UML models of services

Università di Firenze

Francesco Tiezzi
tiezzi@unifi.it

www.sensoria-ist.eu/cows

To support the use of our calculi we have developed a number of tools; an example of this is the VENUS (Verification ENvironment for UML models of Services) tool that permits the verification of service properties by relying on (transparent) mathematically founded techniques.



VENUS architecture

VENUS has been explicitly developed for being accessible by users not familiar with formal methods. Its theoretical bases are the calculus COWS, the temporal logic SocL and the model checker CMC. VENUS automatically translates UML4SOA models of services and natural language statements of service properties into, respectively, COWS terms and Socl formulae, and then checks them using CMC, possibly providing counterexamples.

Service-Oriented Computing Topics

Orchestrations and choreographies characterize service-oriented architectures. We addressed these and many more specific topics of service-oriented systems, like the following.

Università di Pisa

Telecom Italia Lab

Marzia Buscemi
m.buscemi@imtlucca.it

QoS negotiation. Quality of Service (QoS) plays a key role in service composition as services providing the same functionalities can be differentiated according to their QoS guarantees. We propose the cc-pi calculus as a constraint-based model of QoS negotiations. The cc-pi calculus combines basic operations of concurrent constraint programming with a symmetric, synchronous mechanism of interaction between senders and receivers. Furthermore, the cc-pi calculus is parametric with respect to the choice of an underlying constraint system that is defined using a suitable semiring structure, equipped with a notion of names. We adopt two case studies of **Sensoria**: The telecommunication case study for specifying and enforcing Telco policies, the finance case study for showing how to model.

Università di Pisa

Università di Trento

Call-by-contract. Call-by-contract is a novel invocation mechanism for Web services, which allows services to call each other according to their behaviour. We have proposed both a design framework and a core

Massimo Bartoletti
bartolet@di.unipi.it

programming language for call-by-contract service orchestration. We have devised several analysis techniques for constructing orchestration plans which are always guaranteed to respect the requested contracts.

Università di Firenze

Università di Pisa

Università di Bologna

Universidade de Lisboa

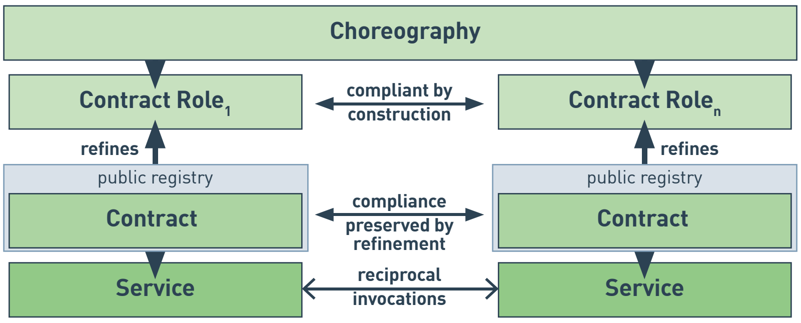
Ivan Lanese
lanese@cs.unibo.it

Long running transactions. Long running transactions are distributed transactions lasting for long periods of time, which thus can not enjoy the classic atomicity property. In case of failure of the transaction, a compensation is executed to take the application to a consistent state. We have studied primitives and techniques for modelling long running transactions in the field of Service-Oriented Computing. We have both studied and compared classic approaches (try-catch, SAGAs,...) and proposed new ones, such as dynamic compensation update. We have also tackled the peculiar aspects that emerge in the SOC environment, such as the problems of smooth session closure and of remote fault notification.

Università di Bologna

Mario Bravetti
bravetti@cs.unibo.it

Choreographies and Behavioural Contracts. Choreographies are specification languages that represent a desired interaction behaviour for a service-based system where each service has a role. From a choreography we derive a set of behavioural contracts, one for each role, that are guaranteed to be compliant (e.g. deadlocks do not occur). We consider several notions of refinement, which guarantee different forms of compliance and are related to different forms of interaction, e.g. synchronous or queue-based asynchronous communication.



Compliance-preserving contract refinement

Università di Pisa

University of Leicester

Gian Luigi Ferrari
giangi@di.unipi.it

Event-based Service Coordination. We develop a new methodology for service oriented computing based on an event-based coordination model. The proposed framework goes all the way from a foundational process calculus, the Signal Calculus, and its choreography model, Network Coordination Policy, over a Java middleware, Event-based Service Coordination, to its application for Sensoria case studies. The usefulness of the proposed approach has been illustrated by tackling the problem of designing and implementing long running transactions. Furthermore, a set of refactoring rules have been developed for refining the implementation during the deployment phase.

Multi-party sessions. In SOC it is important to precisely ascertain which participants are entitled to receive messages. The set of such participants can be abstracted as a session. For instance, a travel booking service has to serve many requests concurrently; this can be represented as a set of sessions each involving (an instance of) the booking service and a client. It must be guaranteed that it never happens that a confirmation for a client is received by another one. Moreover, service-oriented applications feature interactions among several distributed participants, namely protocols often require more participants to be able to correctly interact.

The main engineering approaches studied in **Sensoria** for such complex interactions are based on correlation sets and sessions:

- Correlations: the participants to a session are determined by correlation values explicitly included in the messages.
- Explicit naming: service invocations automatically generate unique names pinpointing sessions which are then managed to let participants enter/leave sessions.

Universidade de Lisboa

University of Leicester

Università di Pisa

Università di Bologna

Università di Firenze

Emilio Tuosto
et52@mcs.le.ac.uk

Architectural Design Rewriting. Architectural Design Rewriting (ADR) is an approach for the design of software architectures by reconciling graph transformation, process calculi and software engineering techniques. The key features that make ADR a suitable and expressive framework are the algebraic presentation of hierarchical graph-based structures, which can improve the automated support for specification, analysis and verification of service-oriented architectures and applications.

ADR has been entirely developed under **Sensoria**, allowing to establish interesting links with many other formalisms developed within the project, ranging from SRML, UML4SOA, service modes and service-oriented process calculi, and making it possible to strengthen the collaboration among different research groups with different expertise within the project.

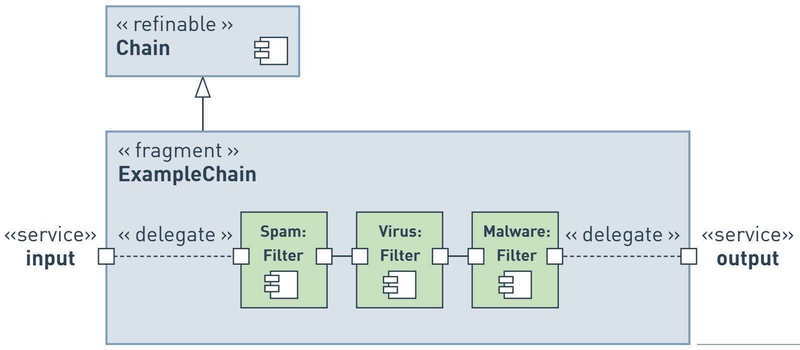
Università di Pisa

University of Leicester

Ludwig-Maximilians-Universität München

London Software Systems
- **Imperial College London**
- **University College London**

Alberto Lluch Fuente
lafuente@di.unipi.it



ADR architectural configuration

Methods and Tools for Qualitative Analysis

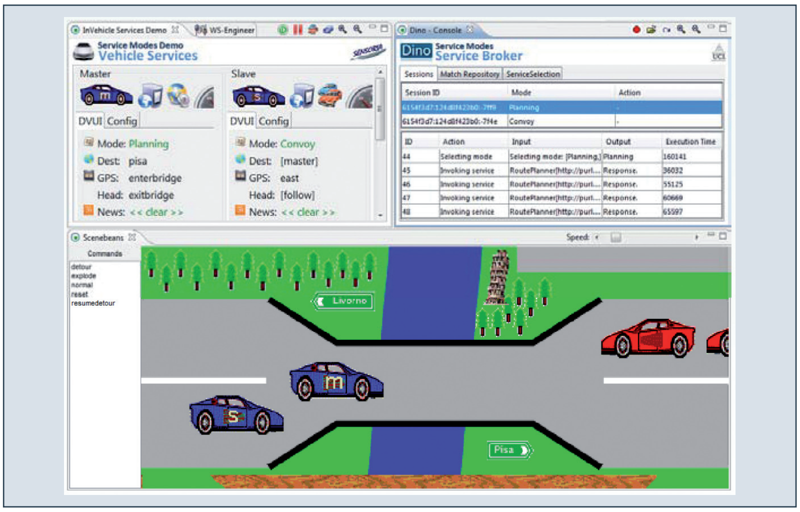
London Software Systems
- **Imperial College London**
- **University College London**

Howard Foster
howard.foster@imperial.ac.uk

www.sensoria-ist.eu/wse

Adaptive and Dynamic Service Compositions. The LTSA WS-Engineer + Modes tool provides mechanical support for the analysis of Service Mode models to ensure safety and correctness of adaptive and dynamic service composition specifications. The goal is to ensure that quiescence (consistency of system state before and after changes) is upheld once the system is deployed. Properties for analysis include:

- Service Protocol Compatibility to analyse the expected provided and required service interactions between services in each mode composition,
- Mode Behaviour Reachability to analyse the expected service composition behaviour with that which is offered by the services in the mode configuration and
- Modes Composition Analysis to analyse the composition of system behaviour specified in all the modes given in the Modes architecture specification.



Master:
Mode: Planning
Dest: pisa
GPS: enterbridge
Head: exitbridge
News: << clear >>

Slave:
Mode: Convoy
Dest: [master]
GPS: east
Head: [follow]
News: << clear >>

Session ID	Mode	Action
6154f3d7:124d8f423b0:-7ff0	Planning	-
6154f3d7:124d8f423b0:-7f4e	Convoy	-

ID	Action	Input	Output	Execution Time
44	Selecting mode	Selecting mode: [Planning.]	Planning	160141
45	Invoking service	RoutePlanner[http://purl...	Response.	36032
46	Invoking service	RoutePlanner[http://purl...	Response.	55125
47	Invoking service	RoutePlanner[http://purl...	Responce.	60669
48	Invoking service	RoutePlanner[http://purl...	Response.	65597

Commands:
detour
explode
normal
reset
resumedetour

Service modes demonstration

The Sensoria UML family of profile assists service engineers in specifying components, behaviour, dynamic adaptation and service brokering. UML service mode models are mapped to finite state processes and analysed for safety and correctness using the Labelled Transition System Analyser (LTSA).

Università di Trento

Università di Pisa

Roberto Zunino
zunino@disi.unitn.it

LocUsT Model Checker. The LocUsT tool is a model checker which verifies safety properties on the behaviour of services. LocUsT is used both as a static checker of usage policies, and as a verification core for the call-by-contract service orchestration framework developed within Sensoria.

The SocL Logical Framework. SocL is a service-oriented logical framework (logic + verification engine) with the following characteristics: The logic is an efficiently verifiable, state- and event-based, parametric, branching-time temporal logic which allows to naturally express typical service-oriented properties like session data correlation, service availability, and service responsiveness.
Its verification engine is an on-the fly verification engine which uses a bounded model-checking approach for the verification of formulas also on infinite models, which generates "on demand" only the needed fragment of the overall state space, and which abstracts from the internal details of the underlying computational model by just viewing it as an abstract doubly-labelled structure.

ISTI Pisa

Università di Firenze

Franco Mazzanti
mazzanti@ist.cnr.it

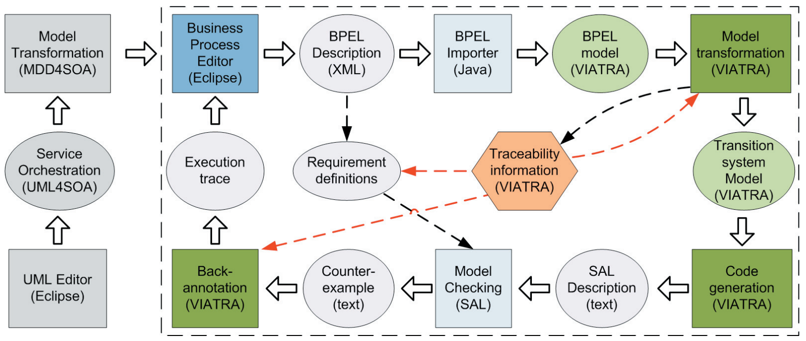
CMC-UMC. CMC and UMC are two prototypical instantiations of the SocL logical verification framework for the analysis of qualitative properties of service-oriented systems. They only differ w.r.t. the underlying computational models which are based on COWS specifications in the case of CMC, and on UML statecharts in the case of UMC.

Both tools can be publicly accessed as web applications, downloaded as platform-specific native applications (for Linux, Mac, Windows) or as platform-independent Java packages, and their basic functionalities are available as plugins inside the SDE. Automated / interactive translation aids allow service designers to specify service-oriented applications as UML4SOA diagrams and analyze them by means of the CMC-UMC verification framework.

BPEL Analysis and Back-Annotation. BPEL is the industrial de-facto standard used for modelling service orchestrations. We have developed an end-to-end method which facilitates the analysis of several liveness and safety properties of such an orchestration. Although several such methods exist, very little attention has been paid on the back-annotation of the results to the original process modelling language; our method helps both the intuitive definition of requirements and shows analysis results directly on the business process, allowing an interactive "simulation" of traces generated by model checking.

Budapest University of Technology and Economics

Daniel Varró
varro@mit.bme.hu



BPEL analysis and back-annotation process

Techniques and Tools for Quantitative Analysis

Università di Trento

Paola Quaglia
quaglia@disi.unitn.it

www.sensoria-ist.eu/scows-lts

www.sensoria-ist.eu/scows-amc

Quantitative analysis of for example resource usage or quality-of-service metrics is a powerful tool in early evaluation of service provision. The **Sensoria** approach offers the following service description mechanisms and tools facilitating such quantitative analysis of SOAs.

sCOWS. Stochastic COWS permit the description of services whose basic actions are associated with a rate expressing their delay. This makes sCOWS a language suitable for modelling services whose behaviour can be analysed using Markovian techniques.

sCOWS_LTS is a tool that offers sCOWS probabilistic model checking through the generation of the LTS (Labelled Transition System) corresponding to the specification, and its subsequent translation to a Continuous Time Markov Chain that can be used as input for the PRISM model checker of CSL (Continuous Stochastic Logic) formulas.

sCOWS_AMC is a tool that implements approximate statistical model checking of sCOWS terms against CSL. This is obtained by generating simulation traces of computations and applying statistical reasoning.

University of Edinburgh

Stephen Gilmore
stg@staffmail.ed.ac.uk

www.sensoria-ist.eu/srmc

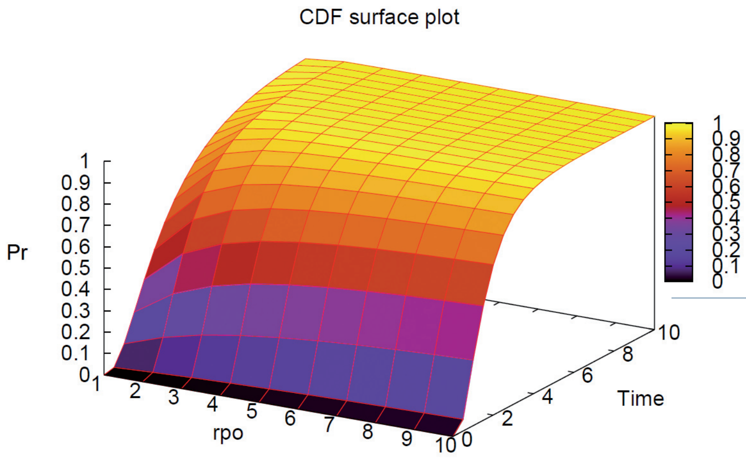
SRMC - **Sensoria** Reference Markovian Calculus - is a stochastic process calculus which captures the inherent uncertainty in service-oriented systems.

The language contains linguistic features to differentiate between systems on the basis of the service providers which they have deployed. These service providers do not need to be performance identical and they do not even need to be identical in behaviour. SRMC allows the model to express both kinds of uncertainty and to evaluate this to give performance predictions which are valid whichever configuration of service providers is selected.

The SRMC language is implemented by transformation into the PEPA process calculus. PEPA - Performance Evaluation Process Algebra - is a concise modelling language which offers the unique capability to analyse models of service-oriented systems using both discrete and continuous methods.

The advantage which this flexibility brings is that models of very large systems with large numbers of repeated components, as typically found in service-oriented computing, can be efficiently analysed using continuous methods. This makes the language applicable in modelling domains where other languages fail. In the **Sensoria** project we have used the PEPA language to analyse the scalability of large-scale systems.

The PEPA language was enhanced during the **Sensoria** project to use an entirely novel continuous-space semantics allowing large-scale models to be solved with ease. The implementation technology for this is a component of the **Sensoria** Development Environment, a state-of-the-art Eclipse-based tool for formal analysis of service-oriented systems.



Analysing an SRMC model

MarCaSPiS is a process language specifically designed for addressing quantitative aspects of service-oriented computing. The language is a Markovian extension of the CaSPiS **Sensoria** core calculus, the basic concepts being service-definitions, with weighted input activities, service invocations, concretion, and value return, with rated output activities, and sessions.

SoSL - Service-oriented Stochastc Logic - is a probabilistic, timed, temporal logic, which extends CSL in several ways: it is both state- and action-based and, in particular, classes of actions can be specified by means of action specifiers; moreover, the logic is resource-oriented in order to address open-endedness of service-oriented computing. A stochastic model-checking algorithm and related tool have been developed which use model-checking capabilities of MRMC for checking MarCaSPiS properties expressed in SoSL.

ISTI Pisa

Università di Firenze

Diego Latella
diego.latella@isti.cnr.it

www.sensoria-ist.eu/marcaspis

SoSL-MC is a model checker that permits verifying whether a given MarCaSPiS specification satisfies or not a SoSL formula. The idea is to use an existing state-based stochastic model-checkers, the Markov Reward Model Checker (MRMC), and wrapping them in the SoSL model-checking algorithm.

SoSL-MC, which is implemented in OCaML, permits analysing the execution of MarCaSPiS programs and generating their reachability graphs. Moreover, after loading a MarCaSPiS specification and a formula, it verifies, by means of one or more calls to MRMC, the satisfaction of the formula by the specification.

Università di Firenze

Michel Loreti
michele.loreti@unifi.it

www.sensoria-ist.eu/sosl-mc

Tools for Model-Driven Development

Ludwig-Maximilians-Universität München

Philip Mayer
mayer@pst.ifi.lmu.de

www.sensoria-ist.eu/sde

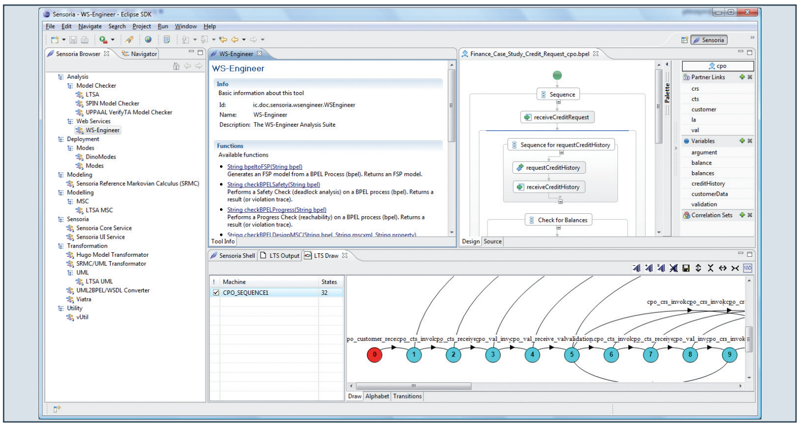
SDE - Sensoria Development Environment

SDE is an Eclipse-based development environment for service-oriented software, which supports, through integrated tools, service modelling, analysis, code generation, and runtime functionality. The core of SDE offers:

- **A SOA-based platform.** The SDE itself is based on a Service-Oriented Architecture, allowing easy integration of tools and querying the platform for available functionality. The tools hosted in the SDE are installed and handled as services.
- **A Composition Infrastructure**. As development of services is a highly individual process and may require several steps and iterations, the SDE offers a composition infrastructure, which allows developers to automate workflows as an orchestration of integrated tools.
- **Hidden Formal Methods.** Developers can use formal tools without requiring them to understand the underlying formal semantics, the SDE encourages the use of automated model transformations which translate between high-level models and formal specifications.

www.sensoria-ist.eu/sdetools

The following tools for the development of SOA systems, including formal analysis tools, have been integrated into the SDE as Eclipse plug-ins.



WS-Engineer

Info
Basic information about this tool

Id: ic.doc.sensoria.wsengineer.WSEngineer
Name: WS-Engineer
Description: The WS-Engineer Analysis Suite

Functions
Available functions

- String bpeltoFSP(String bpel)
  Generates an FSP model from a BPEL Process (bpel). Returns an FSP model.
- String checkBPELSafety(String bpel)
  Performs a Safety Check (deadlock analysis) on a BPEL process (bpel). Returns a result (or violation trace).
- String checkBPELProgress(String bpel)
  Performs a Progress Check (reachability) on a BPEL process (bpel). Returns a result (or violation trace).
- String checkBPELDesignMSC(String bpel, String mscxml, String property)

!	Machine	States
☑	CPO_SEQUENCE1	32

SDE and several installed tools

Ludwig-Maximilians-Universität München

Cirquent GmbH

Philip Mayer
mayer@pst.ifi.lmu.de

www.sensoria-ist.eu/mdd4soa

MDD4SOA – A suite of transformations enabling a model-driven development process based on UML4SOA models.

MDD4SOA consists of a set of Eclipse plugins which directly work on UML4SOA models, creating executable code for the Web Service standards family (BPEL, WSDL, XSD), the Java programming language, and the Jolie SOA orchestration language. Furthermore, the suite contains transformations for adding UI support to BPEL processes, and the generation of deployment artefacts for industry-standard application servers.

SRMC – The plug-in is a software tool that supports the stochastic process algebra PEPA allowing quantitative analysis of systems. SRMC/UML complements SRMC translating a subset of UML2 models (interactions and state machines) into an SRMC description for performance evaluation. Results are reflected back into the UML model.

University of Edinburgh

Stephen Gilmore
sg@ed.ac.uk

www.sensoria-ist.eu/srmc

CMC – Model checker and analyser of abstract behavioural properties for systems defined by interacting UML statecharts, which are textually represented (COWS process algebra). Allows to generate abstract full-trace minimized graphs of the system.
UMC – Model checker and analyser of abstract behavioural properties for systems defined by interacting UML statecharts, which are represented in umc format. Allows to generate abstract full-trace minimized graphs of the system.

ISTI Pisa

Franco Mazzanti
mazzanti@ist.cnr.it

www.sensoria-ist.eu/cmc
www.sensoria-ist.eu/umc

LTSA WS-Engineer – supports cross-cutting mechnical analysis of service compositions (design, interactions, choreography, deployment) to ensure safety.
Service Modes – supports analysis of service mode models for adaptive and dynamic service composition.
Dino Service Modes – supports mechanical generation of Dino Broker runtime specifications from service mode models for use with the Dino Broker tool.
Dino Broker – supports runtime discovery and brokering of services using requirements and capabilities specifications for required and offered service specifications.

London Software Systems
- **Imperial College London**
- **University College London**

Howard Foster
howard.foster@imperial.ac.uk

www.sensoria-ist.eu/wse

LTSA WS-Engineer

SOA2WSDL Transformer - takes high level UML4SOA models and produces WSDL output.
UML2Axis Transformer – takes high level UML4SOA models and produce WSDL, WS-ReliableMessaging, WS-Security and Apache Axis-specific configuration files as output.

Budapest University of Technology and Economics

Lázló Gönczy
gonczy@mit.bme.hu

www.sensoria-ist.eu/uml2axis

LYSA Static Protocol Analyzer – provides prototype LYSA editor to assist users in the modelling of security protocols and to verify properties related to secrecy and authentication.

TU Denmark at Lyngby

Henrik Pilegaard
hepi@imm.dtu.dk

www.sensoria-ist.eu/lysa

VIATRA2 - is a live transformation framework that has been implemented, based on the incremental pattern matching engine of VIATRA which support the development of event-driven transformations. The novel model transformation by example approach has been developed to enable high-level specification of model transformations by the definition of typical input-output model parts. This tool is also part of an official Eclipse project.

Budapest University of Technology and Economics

Dániel Varró
varro@mit.bme.hu

www.sensoria-ist.eu/viatra

Tools for Deploying Service-Oriented Systems

London Software Systems

- **Imperial College London**
- **University College London**

David Rosenblum
d.rosenblum@cs.ucl.ac.uk

www.sensoria-ist.eu/wse

Dino Service Modes Service Broker

Adaptive and Dynamic Service Compositions

The Service Modes pattern can also be used to specify service brokering components and their requirements or capabilities. The LTSA WS-Engineer + Modes tool provides transformations from service mode models to service brokering requirements and capability specifications. The transformations generate documents which are deployed on to a runtime broker. Thus, at runtime the requirements documents are used by service clients to create a new brokering session and trigger discovery of required services.

Capabilities may also be registered with the service broker, which offers provided services and adds service capability to discoverable services. [Poster, WS-Engineer Tool, Dino Runtime Broker, Video]

Budapest University of Technology and Economics

Lázló Gönczy
gonczy@mit.bme.hu

www.sensoria-ist.eu/uml2axis

Customizable Model Transformations for Deployment

Up to very recently, Web services messaging standards used to capture a different subset of non-functional parameters making even closely related standards incompatible with each other. Also the service developer had to specify service configurations at a very low technical level. To tackle these problems, we propose a model-driven approach to efficiently design and deploy standards-compliant service configurations with non-functional parameters.

From such engineering models, we automatically generate service descriptors (WSDL) and configuration descriptors for standard platforms like Apache Axis, including the definition of non-functional requirements, and server-side deployment artifacts. Our method is based on customizable model transformations which allow the developer to describe non-functional requirements in UML4SOA and generate different analysis and deployment code, also allowing an early estimation of the performability of a given middleware configuration.

Università di Firenze

Rosario Pugliese
rosario.pugliese@unifi.it

www.sensoria-ist.eu/blitec

Rapid development of WS-BPEL applications

BliteC is a software tool for supporting a rapid and easy development of WS-BPEL applications. BliteC translates service orchestrations written in Blite, a formal language inspired to but simpler than WS-BPEL, into executable WS-BPEL programs.

The tool simplifies the task of developing WS-BPEL applications because Blite provides a textual programming notation and is equipped with an unambiguous semantics, while BliteC properly packages the produced files to be readily deployed and executed in a WS-BPEL engine.

Transformation of Legacy Systems into SOAs

ATX II - Tecnologias de Software S.A.

University of Leicester

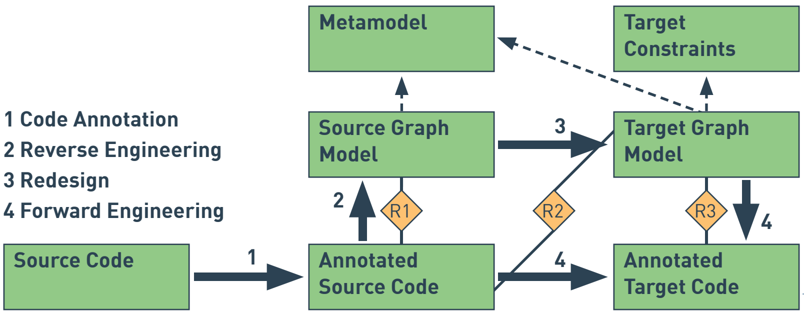
Carlos Matos
carlos.matos@atxtechnologies.com

With the growth in SOA adoption, the need for a systematic approach towards reengineering for SOA also increases. However, several principles of service-orientation pose major challenges for these efforts:

- The separation of business from presentation logic;
- The loosely coupled relationship between services;
- The coarse-grained nature of services.

As legacy systems were not built with these concerns in mind, much effort is needed to accommodate them. The work in **Sensoria** for reengineering towards SOA consisted in the development of a methodology that allows a high degree of automation, providing support for the full reengineering cycle and taking into consideration scalability matters.

Our proposal can be seen as an instance of the horseshoe model, a conceptual model for reengineering at different levels, with a focus on transformations at the level of architectural models. This goal is achieved by using techniques such as code pattern matching (to annotate the code), reverse engineering, graph transformation and forward engineering.



Reengineering approach

The process is instantiated in two dimensions to address the technological and the functional evolution. The former is concerned with the technical purpose of the code and the latter focuses on its implementation of relevant business-level functionalities. In order to evaluate this approach a prototype was developed covering the following steps:

1. Code annotation implemented via an Eclipse plug-in developed by ATX, called CareStudio, which allows the specification and execution of code pattern matching rules.
2. Reverse engineering achieved via a small tool that was designed just for the purpose of obtaining a graph model from the annotated source code.
3. Definition of graph transformation rules implemented by using the Eclipse plug-in Tiger EMF Transformer.
4. Forward engineering is based on a tool that invokes Eclipse Java refactorings.

Pattern-based Service Engineering

All Partners

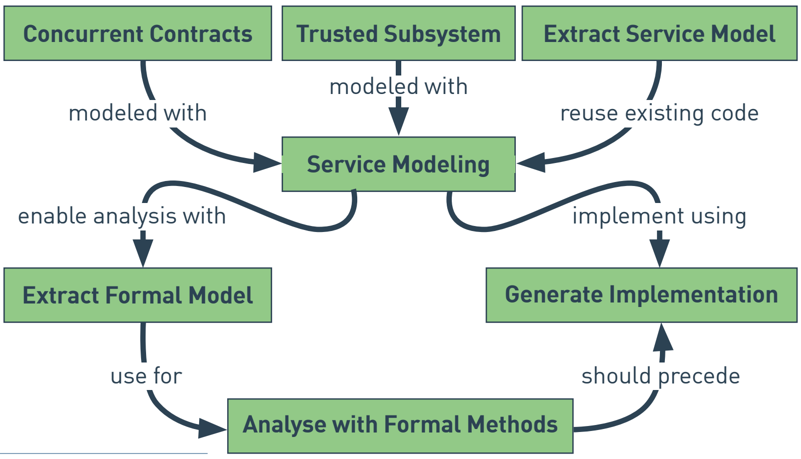
Matthias Hölzl
hoelzl@pst.ifi.lmu.de

The broad range and the depth of the methods developed as part of the **Sensoria** project means that it may be difficult for developers to identify the technique or tool that solves a particular problem arising in the development process, unless the developers are familiar with the whole range of scientific results of the project. To ameliorate this problem we developed a catalogue of patterns that serve as an index to our results and that illustrates, in a concise manner, the advantages and disadvantages of the individual techniques.

The **Sensoria** patterns are not limited to implementation issues, they encompass a wide range of abstraction levels, from classical implementation-oriented patterns to architectural or process patterns. We structure the patterns in a way that approximately follows the "Pattern Language for Pattern Writing", but add some pattern elements that seem to be helpful for describing patterns specifically related to service-oriented software engineering.

Some of the patterns support the whole development process, from the modelling stage to deployment activities and give an overview of many of the research areas pursued in the **Sensoria** project. The patterns identified and described so far are:

- Service modelling
- Extract formal models
- Analyse with formal models
- Generate implementation
- Extract service model
- Concurrent contracts
- Trusted subsystems



Relationship among Sensoria patterns

Case Studies

The **Sensoria** project has employed realistic case studies for developing intuitions that have fed and steered the research process according to the expectations of society and its economy, discussing and communicating ideas among partners and, finally, communicating research results to and getting feedback from the research community at large, both in industry and academia. Two of the case studies are from industrial applications in the automotive and finance domains and one is an academic application for distributed e-learning and course management.

ISTI Pisa

Stefania Gnesi
stefania.gnesi@isti.cnr.it

Automotive domain. Due to the advances in mobile technology, it is now possible to take connectivity to the car: Internet and telephone access in vehicles are possible today, giving rise to a variety of new services for the automotive domain. Examples of scenarios analysed during the project are on road assistance in case of accident, and route planning illustrating how a vehicles' navigation system can react to external events. The selected automotive scenarios allowed to encompass the complexity of SOAs.

Cirquent GmbH

Ludwig-Maximilians-Universität München

Nora Koch
kochn@pst.ifi.lmu.de

Finance domain. A typical application of the finance sector is the credit approval process. The credit request scenario models such a loan workflow, including interaction with customers and employees. The credit request workflow is implemented using a SOA-based system: Several specialised services are orchestrated to realise the process. **Sensoria** tools and methods have been used to verify properties of this system, such that a minimum error rate in the credit request and approval process can be achieved.

S&N AG

Jannis Elgner
jelgner@s-und-n.de

eUniversity. To investigate the problem of developing SOA-based university management systems, an academic case study based on a set of eUniversity scenarios was analysed that make use of the specific features of SOAs focusing on administrative functionalities. Scenarios in such an environment include online management of curricula, students and their thesis.

Ludwig-Maximilians-Universität München

Matthias Hölzl
hoelzl@pst.ifi.lmu.de

Application of techniques, methods and languages to case studies

ISTI Pisa

Maurice ter Beek
maurice.terbeek@isti.cnr.it

The following table provides an analytic overview of the specific experience and benefits of having applied **Sensoria** techniques, methods and languages to the case studies. This table illustrates the central role of the industrial case studies from the automotive and finance domains, as well as the more specific role of the academic eUniversity case study.

	Automotive	Finance	eUniversity
UML4SOA	UML Profile complementing existing UML extensions like SoaML by introducing more service-specific model elements, allowing the design of high-level input models for formal modelling and verification		
SRML	Offers a methodological approach with primitives for high-level modelling of business services and activities, including the use of UMC and of the Markovian process algebra PEPA.		
Core Calculi (s)COWS, cc-pi, (Mar)CaSPiS, SOCK/Jolie, etc	Formal models for QoS negotiation, long running transactions, choreographies and behavioural contracts, event-based service coordination, call-by-contract, etc.		
ADR	Provides a formal model of reconfigurations and constraints		
CMC-UMC SocL	Model-checking framework for qualitative analyses of formal models in COWS or UMC, using the logic SocL with predefined patterns of service properties		
WS-Engineer	Provides mechanical support for the analysis of Service Mode models to ensure safety and correctness of adaptive and dynamic service composition specifications		
SoSL	A logic for expressing dependability (workload, reactivity) and performance properties of services, which in natural language is difficult and error-prone		
sCOWS_LTS sCOWS_AMC		Model-checking tools for quantitative analyses of formal models in sCOWS	
SRMC	Translates a subset of UML2 models (interactions and state machines) into an SRMC description for performance evaluation. Results are reflected back into the UML model		
Service Modes	Addresses reconfiguration management within a self-managed service system		
Dino	Provides mechanical support for runtime discovery and brokering of services		
MDD4SOA	The MDD4SOA transformers provide a full transformation from UML4SOA to actual code and execution		
VIATRA Transformations	Deployment transformations help the generation of configuration code and policies describing QoS of services		
Patterns	A pattern catalogue documents the advantages/disadvantages and feedback of the **Sensoria** techniques, methods and languages		
SDE	SDE contains all the **Sensoria** tools and provides orchestration features to combine integrated tools for modelling and analysis		

Spin-off Companies

italianaSoftware s.r.l.

Claudio Guidi
cguidi@italianasoftware.com

www.italianasoftware.com

This company was born as a research spin-off within the European project **Sensoria** at which the University of Bologna was involved for studying and developing new languages for service-oriented architectures.

italianaSoftware proposes a new language, called JOLIE, for designing, developing and deploying services and orchestrators. The company provides solutions and consulting activities for any kind of customer requirement about SOA design and implementation.

In 2007 italianaSoftware won the Imola StartCup, Itech.Off and Ingenium awards and was selected for participating at the National Award for Innovation 2007.

Agilogik GmbH Steingaden

Matthias Hölzl
hoelzl@agilogik.de

www.agilogik.de

Agilogik GmbH is a spin-off of the **Sensoria** project that develops adaptive, service-oriented business solutions based on the research results of **Sensoria**, combined with techniques from artificial intelligence and multi-agent systems. Agilogik provides software for fraud detection, personal productivity and effectiveness enhancement, and for campaign planning, optimization and management.

One core component of Agilogik's products is a sophisticated solver for multi-criteria optimization problems which implements the monoidal soft constraint (MSC) theory that was developed as part of the **Sensoria** project. The MSC solver can efficiently deal with linear and non-linear optimization problems that require trade-offs between competing optimization goals, and it can incrementally update existing solutions when input parameters or trade-offs between goals change.

OptXware Resarch and Development

László Gönczy
gonczy@optxware.com

www.optxware.com

OptXware was founded to industrialize research results of members of Budapest University of Technology and Economics, Fault Tolerant Systems Research Group. The company provides consultation and development services in business systems and offers analysis and optimization for business processes.

OptXware won Hungarian national support as an innovative start-up.

Sensoria in Numbers

Project	
start	1.9.2005
end	28.2.2010
Partners	19
Universities	14
Research organisations	1
Companies	4
Participating countries	7
Participants	
Core researchers	55
Researchers partially involved	65
Associated researchers	37
PhDs thesis on Sensoria results	
Finished	24
Ongoing	29
Project reports (deliverables)	110
Publications	652
Books	6
Book chapters	16
Articles in journals	139
Papers in conferences and workshops	403
Edited volumes	25
Technical reports	63
Presentations and tutorials	192
Summer schools	3
GLOBAN 2006, GLOBAN 2008, SENSUS 2009	
Courses based on project results	108
Conferences/workshops organised by project members	126
Fairs and exhibitions	4
SYSTEMS 2007, CeBIT 2008, ICT 2008, FET 2009	
Software	
CASE tool: **Sensoria** Development Environment (SDE)	1
Integrated tools	19
Additional tools	8

www.sensoria-ist.eu

SENSORIA

Software Engineering for Service-Oriented Overlay Computers

Information Society Technologies (IST) project funded by the EU as Integrated Project (IP) in the 6th Framework Programme (FP6) as part of the Global Computing Initiative (GC) of the Future and Emerging Technologies (FET) programme

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 | S&N AG | Telecom Italia Lab | Politecnico di Milano | Imperial College London | University College London | ATX II Tecnologies de Software S.A. | Budapest University of Technology and Economics | Cirquent GmbH |

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

Work package leaders

Reiko Heckel | Rocco De Nicola | Fleming Nielson | Davide Sangiorgi | Paola Quaglia | Ugo Montanari | David Rosenblum | Stephen Gilmore | Stefania Gnesi | Nora Koch | Andrzej Tarlecki | András Pataricza | Matthias Hölzl

Published by

SENSORIA Consortium
www.sensoria-ist.eu

Text editor

Nora Koch
kochn@pst.ifi.lmu.de

Layout

Urs Haggenmüller
urs.haggenmueller@cirquent.de