COMPAS

Compliance-driven Models, Languages, and Architectures for Services

7th Framework Programme

Information and

Communication Technologies

SEVENTH FRAMEWORK PROGRAMME

www.compas-ict.eu

CWI

Telcordia®

pwc

THALES

LIRIS

TILBURG UNIVERSITY

VITALAB VIENNA INTERNET TECHNOLOGIES ADVANCED RESEARCH LAB
Distributed Systems Group

UNIVERSITY OF TRENTO - Italy
Information Engineering
and Computer Science Department

Universität Stuttgart
Germany

This document reflects only the authors' view. The European Community is not liable for any use that may be made of the information contained herein.

Publishable Summary

1. The Challenge

Over the last years, business compliance, i.e., the conformance of business procedures with laws, regulations, standards, best practices, or similar requirements, has evolved from a prerogative of lawyers and consulting companies to a major concern also in IT research and software development. Given the increasing IT support in everyday business as well as the repetitive and work-intensive nature of compliance controls and audits, this evolution can be seen as a natural extension of current enterprise software, especially in light of the novel, technical opportunities offered by the Service-Oriented Architecture (SOA). Yet, until only few years ago, compliance management was not perceived as major concern in IT research.

2. Addressing the Challenge: The Project's Proposition

In this context, COMPAS was surely one of the forerunners and first international research efforts recognising both the need for IT support in compliance management and the spreading of the SOA in today's business realities. COMPAS is a Specific Targeted Research Project (STREP) funded by the European Commission under the 7th Framework Programme. The project had a budget of 3.920.000 € and started in February 2008 with a duration of 36 months. COMPAS is a NESSI Project and targets standardization of some parts of its contributions.

Pragmatically, COMPAS did not aim at over-engineering the compliance problem, e.g., by allowing compliance experts to enforce compliance of individual messages flowing through a company's IT infrastructure, and instead focused on compliance awareness, that is, on the design for, monitoring, and reporting on compliance. As such, it particularly follows the pace of business, not that of IT systems, a feature that turns it into a valuable instrument in the hands of those who have to deal with compliance at an everyday basis.

The COMPAS approach should not be expected as an ultimate solution to compliance management, in the sense that, like other similar research projects, it does not cover all possible compliance requirements imaginable. Yet, this is mostly due to the very nature of compliance, which is a multi-faceted and interdisciplinary problem that cannot be approached via IT only and instead highly depends on the correct identification and interpretation of the laws and regulations that apply to a given business sector as well as on the attitude a company has (or not) toward compliance. Nevertheless, COMPAS significantly advanced the state of the art in IT compliance management, identifying both which contributions IT can bring to compliance management and which capabilities, instead, are outside its reach.

The COMPAS project realized a practical, yet general enough, modelling approach for specifying service-oriented architectures with compliance concerns. In particular, business processes can be designed and compliance controls can be associated with processes and process elements. For this we profit from applying a model-driven engineering approach and use annotation techniques for relating system and requirement models at design-time. To the best of our knowledge, the COMPAS approach is the first approach that makes this link at design-time and – supported through a model-aware service environment (MORSE) – utilizes such relations at runtime for compliance monitoring.

3. Who Can Benefit from COMPAS

The results from the COMPAS project are relevant to everyone such as large and middle-sized companies who want

- to specify and document compliance requirements originating from laws, regulations, or policies;
- to link IT – in particular business processes and services – to compliance requirements originating from laws, regulations, or policies;
- to establish and realize compliance management for their IT-based business solutions and services.

4. Highlights of Achievements

This result in the following advantages compared to other modelling and/or monitoring approaches while combining their strengths: First, the various stakeholders (e.g., compliance expert, business administrator, IT expert, etc.) can participate in the development process: they are supported through (1) the adoption of the separation of concerns principle and (2) suited domain specific languages with defined levels of abstraction. Second, the information in these models can be used for an automated generation of compliance documentation and a generator can consider the information for building compliant systems or detecting static compliance violations. Third, the monitoring can take place at a higher level of abstraction (i.e., the level of the process and compliance models). This eases the (root cause) analysis and report of dynamic compliance violations (i.e., compliance violations that occur during execution) and helps stakeholders to easily relate to the respective modelling artefacts.

All of the components of the COMPAS architecture have been implemented and tested and we have conducted extensive use-case evaluation for demonstrating the suitability and feasibility of COMPAS approach. We realize the chance of our contributions to impact the field of monitoring in general as well as the field of adaptation. Finally, we expect more work to be conducted by expanding model-driven engineering across the boundaries of generation, i.e., by embedding and working with traceability information and dynamic model look-up and model-based reflection and monitoring.

This document summarizes the achievements of the individual project partners throughout the course of the project.

5. The Results

The COMPAS project had been accomplished to design and implement novel models, languages, and an architectural framework to ensure compliance of services to design rules and regulations. In the COMPAS approach model-driven techniques, domain-specific languages, and service-oriented infrastructure were applied to enable organizations to develop business compliance solutions easier and faster. Compliance refers to the entirety of all measures that need to be taken in order to adhere to laws, regulations, guidelines, and internal policies.

The resulting "design-for-compliance" architecture framework ensures compliant composition of business processes and services, and that allows specification, validation, and enforcement of comprehensive compliance policies related to these processes and services. The framework provides the possibility to enhance business process languages, such as (but not limited to) the

Business Process Execution Language (BPEL), with enforceable compliance concepts and policies. Additionally, the necessary specification languages and models for expressing typical compliance concerns were developed.

A formally grounded and implemented behaviour model for services and service composition were provided, enabling the formal validation of compliance of composed services to the behaviour and process constraint specifications. Consequently, compliance concerns can be checked statically as well as dynamically. Finally, monitoring and management tools had been developed for tracking and validating those compliance concerns that can only be verified at runtime. These tools were complemented with reasoning and mining tooling that helps to discover compliant instances services and processes.

The COMPAS project was scheduled into five milestones:

Milestone 1: Definition of Case Studies for Business Compliance	[M1-6]
Milestone 2: Initial Meta-models and Languages for Business Compliance	[M7-11]
Milestone 3: Initial MDSD Software framework for Business Compliance	[M12-23]
Milestone 4: Initial Compliance Governance Concepts and Software framework	[M12-23]
Milestone 5: Integrated Compliance Software framework and Runtime Infrastructure	[M24-35]

The following table presents a summary of the progress of the project with regard to the milestones. Specifically, the results achieved for the different milestones and any relevant prototypes delivered. Detailed progress results are provided in later sections.

Milestone	**Results Achieved**	**Prototypes Delivered**
Milestone 1: Definition of Case Studies for Business Compliance [M1-6]	• Report on industry experience, state-of-the-art reports • Case studies for research evaluation	
Milestone 2: Initial Meta-models and Languages for Business Compliance [M7-11]	• Overall conceptual and concrete architecture perspectives for COMPAS project • Conducted initial specification of compliance language constructs and operators (see [D2.2]) • Introduced a formal model and validation framework for describing, reasoning and automated analysis of business processes (see [D3.1]). • Introduced a goal-oriented data model for warehousing process execution and compliance data (see [D5.2]).	
Milestone 3: Initial MDSD Software framework for Business Compliance [M12-23]	• Initial standalone MDSD prototypes and documentation • A video was provided demonstrating the process steps in the ICT security case study from THALES.	• The *MDSD software framework (View-based Modeling framework* (see [D1.3]) • The *Compliance Request Language Tool (CRLT)* (see [D2.6]) • A *library and user interface (Eclipse plugin) for verification of service descriptions* (see [D3.3]) • A collection of *extensions to the Business Process Execution Language* (BPEL) (see [D4.2])
Milestone 4: Initial Compliance	• Initial standalone compliance	• The *MDSD software framework*

Governance Concepts and Software framework [M12-23]	governance prototypes • A video was provided demonstrating the process steps in the ICT security case study from THALES.	*(View-based Modeling framework* (see [D1.3]) • *Infrastructure for supporting reusable SOA units* (e.g. process fragments) and for generation and execution of compliant processes (see [D4.4]) • *A compliance governance dashboard* for visualization of compliance state (see [D5.5]).
Milestone 5: Integrated Compliance Software framework and Runtime Infrastructure [M24-35]	• Developers' Integration Meetings • Initial demonstration of standalone prototypes (components) for the software framework and runtime infrastructure	

The tasks from these milestones are subdivided into a number of work packages that comprise areas of expertise from the nine (9) partners that make up the project consortium. The consortium includes academic and industry partners who complement each other's skills through carrying out academic research and providing industry experience in the form of case studies. The industry partners provided the experience that guided the exploitation of the research products.

The COMPAS project has significant positive impact on different areas in service-oriented computing, from industry solutions to addressing new open research issues on how services are developed, composed and maintained. One of the main achievements is the development of a comprehensive SOA business compliance software framework that enables a business to express various compliance concerns using one and the same software framework. The major impact of the COMPAS project spans over the following areas:

- End-to-end business compliance software framework
- Reducing the development complexity
- Business process specification and better reuse of existing services
- Verification and validation of services
- Contributions to open standards

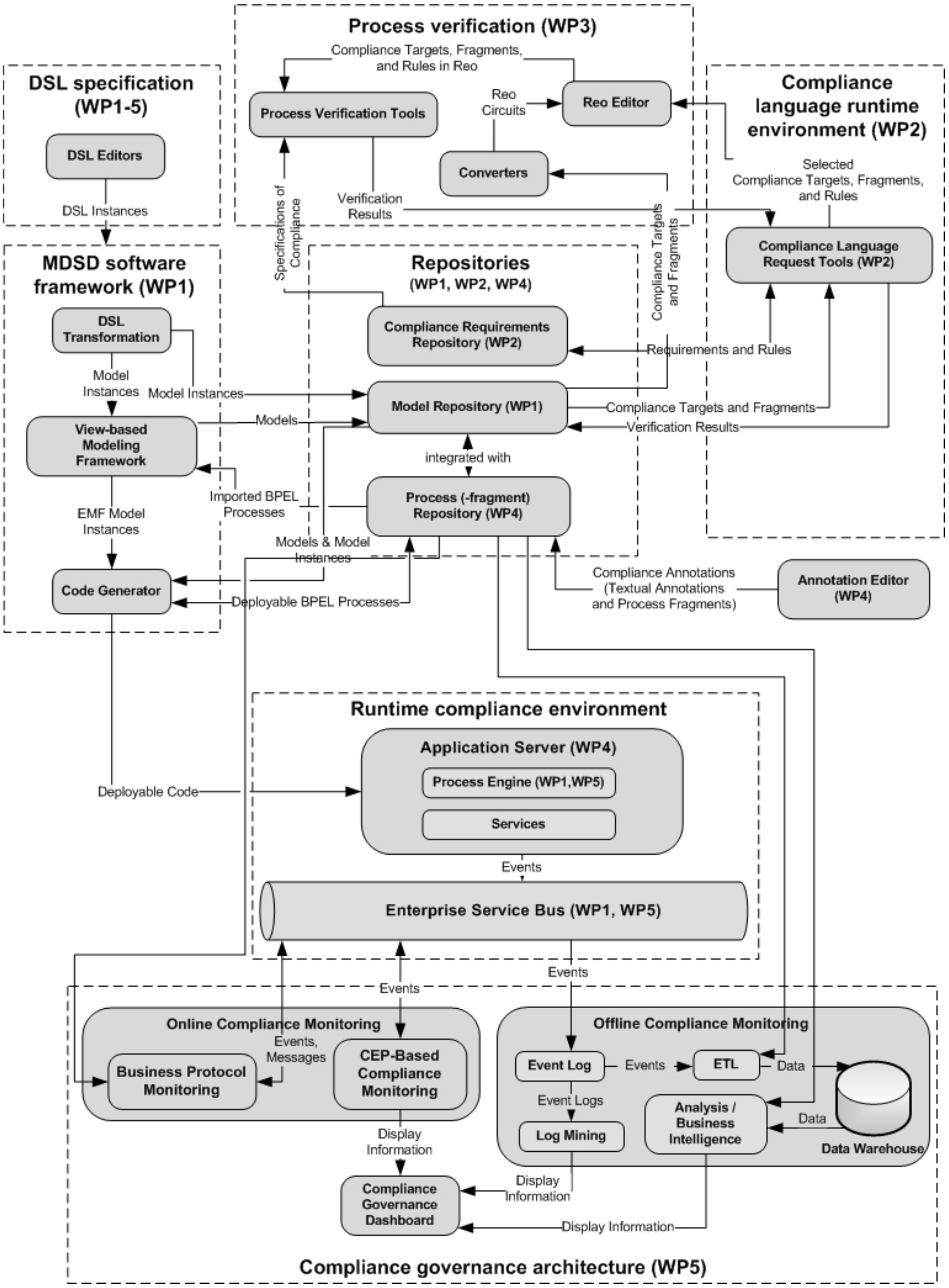
The project also has an impact in terms of inputs to standards and reference architectures, and open source platforms and frameworks. Scientific collaborations inspired by the project have produced numerous scientific publications in various international conferences and journals, whilst industry involvement leads to sharing many valuable experience and knowledge from industry partners.

Ultimately, the project also yields innovations in the important area of business compliance, which have an impact on everyday life - consider the banking crisis that started in 2007 as an example. A lot of regulatory compliance issues have come into question and been revised as a result of this crisis. Organizations need to have a systematic way to implement and adapt their systems in such a dynamic environment. While the impact on the availability to and use by citizens of new products and services are currently only rated as low, it is expected to increase once mature products have been built based on COMPAS technology and concepts.

6. The Pilots

6.1. Overall COMPAS Architecture

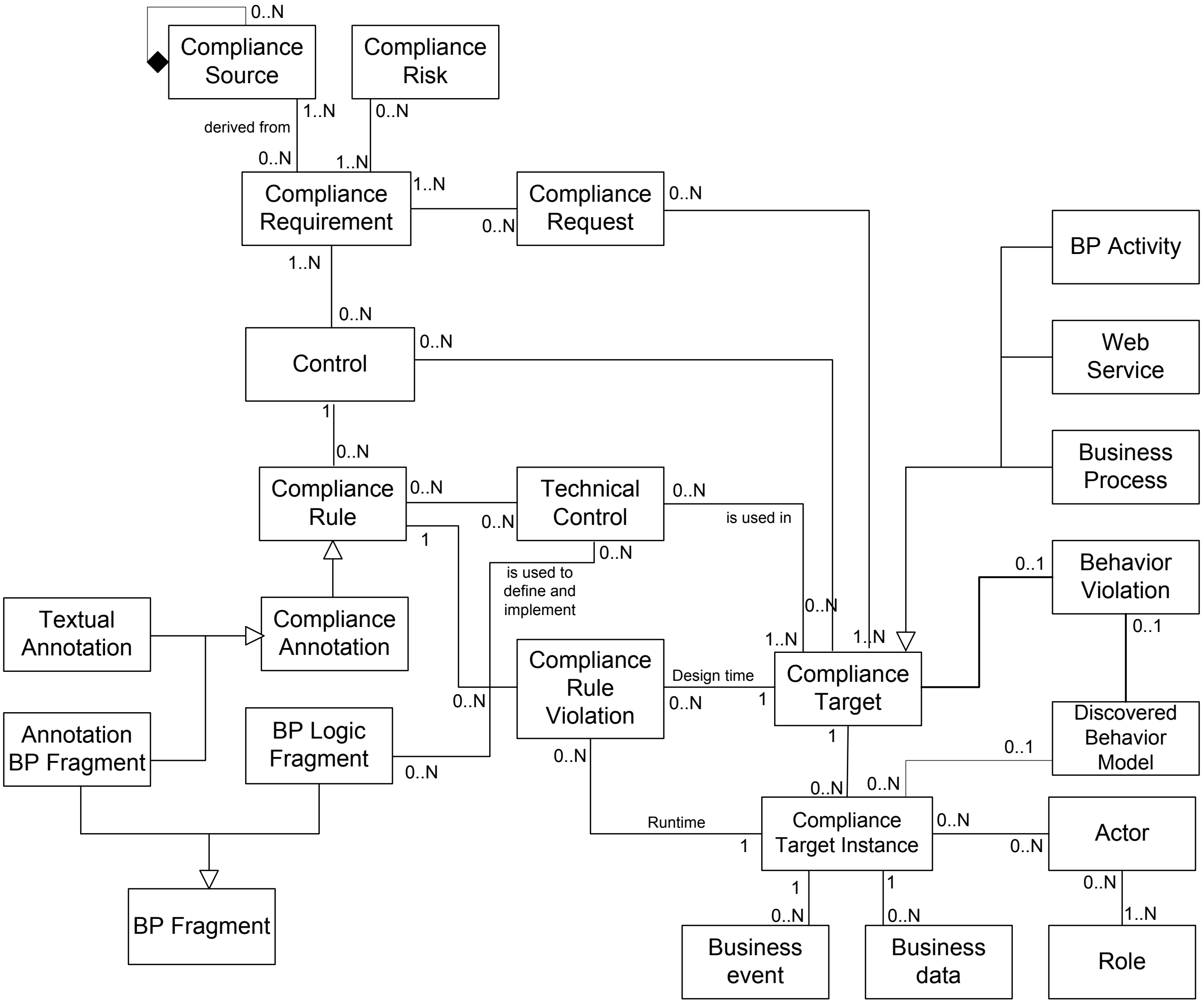
The figure below shows a high-level view on the architecture that had been implemented in the course of the COMPAS project. Each of the components and its integration with other components had been described in project deliverables. A short description of the components is available online at the public COMPAS Web site at http://www.compas-ict.eu/components.



Process verification (WP3)
Compliance Targets, Fragments, and Rules in Reo
Process Verification Tools
Reo Circuits
Reo Editor
Converters
Verification Results
Specifications of Compliance
DSL specification (WP1-5)
DSL Editors
DSL Instances
Compliance language runtime environment (WP2)
Selected Compliance Targets, Fragments, and Rules
Compliance Language Request Tools (WP2)
MDSD software framework (WP1)
DSL Transformation
Model Instances
Model Instances
View-based Modeling Framework
Models
EMF Model Instances
Imported BPEL Processes
Code Generator
Models & Model Instances
Deployable BPEL Processes
Repositories (WP1, WP2, WP4)
Compliance Requirements Repository (WP2)
Model Repository (WP1)
integrated with
Process (-fragment) Repository (WP4)
Compliance Targets and Fragments
Requirements and Rules
Compliance Targets and Fragments
Verification Results
Compliance Annotations (Textual Annotations and Process Fragments)
Annotation Editor (WP4)
Runtime compliance environment
Deployable Code
Application Server (WP4)
Process Engine (WP1,WP5)
Services
Events
Enterprise Service Bus (WP1, WP5)
Events
Events
Online Compliance Monitoring
Events, Messages
Business Protocol Monitoring
CEP-Based Compliance Monitoring
Display Information
Offline Compliance Monitoring
Event Log
Events
ETL
Data
Event Logs
Log Mining
Analysis / Business Intelligence
Data
Data Warehouse
Display Information
Display Information
Compliance Governance Dashboard
Compliance governance architecture (WP5)

6.2. COMPAS Conceptual Model

The figure below shows the conceptual model of the concepts which have been developed in the course of the project. The definition of the terms used in this conceptual model has been made available online at the public COMPAS Web site at http://www.compas-ict.eu/terminology in order to make the project and its results more easily accessible for the public.



6.3. Objectives and Achievements

6.3.1. Modelling of compliance concepts (meta-models and languages) at design time.

Due to the model-driven approach, an accurate modelling of compliance concepts has been essential for the project and for the work of all partners from the beginning on. After iterative design steps, the consortium partners agreed on the COMPAS conceptual model in a dedicated meeting. At this meeting, the model has been designed in cooperation with PWC as experts in the field of compliance.

An important aspect when designing the COMPAS conceptual model was compliance traceability, asking the question: how do compliance requirements relate to compliance sources such as laws or regulations? Such pre-requirements specification traceability has been supported in the model, thus. Another equally important aspect has been generality: as a consequence, the compliance annotation of service-oriented architectural (SOA) elements became generic.

6.3.2. Using model-driven domain-specific languages to support the stakeholders.

Various stakeholders are involved in the design process of compliant business processes, ranging from technical to business experts. We support the stakeholders with tailored domain-specific languages (DSL) following a model-driven approach. As a consequence, business experts do not have to specify any technological artefacts to comply to appropriate compliance concerns. Based on the DSL specifications, an automatic generation of executable code is possible. Technical and business experts can collaborate better for securing the compliance concerns at design time and runtime. In COMPAS, we have developed the Quality of Service Language (QuaLa) for specifying the services' QoS compliance concerns.

6.3.3. Model-driven approach for the generation of business processes with compliance concerns at generation time.

The automatic and model-driven generation of business processes with compliance concerns from conceptual models has been achieved through the view-based modelling framework (VbMF). For this, BPEL and WSDL code is generated and model-traceability information is supplied in form of a traceability matrix.

6.3.4. Supporting model-based reflection for the compliance monitoring at runtime.

Business administrators, compliance experts and other stakeholders specify various concerns (e.g., the control flow of a business process, compliance sources for a compliance requirement, etc.) at design time. Yet, during execution the runtime needs to relate to these concepts. In a distributed and evolving environment we addressed this issue by making models and model elements uniquely identifiable and retrievable. For this we made use of Universal Unique Identifiers (UUIDs) as described by the International Telecommunication Union (ISO/IEC 9834-8, 2004), and provided a Model-Aware Service Environment (MORSE) that realizes transparent UUID-based model-versioning.

6.3.5. Establish monitoring and management of business events in order to proactively identify problems and/or opportunities associated with a given request.

Our aim in COMPAS and specifically in WP5 was to extend current data warehousing and reporting technology toward **event-based business process warehousing and analysis**, which supports the offline monitoring and management of the performance and compliance of executed business process instances. This goal has been achieved by means of the following ingredients that have been conceived and developed throughout the project: an event log for runtime business events, a business event-centric data warehouse, a set of ETL (Extract-Transform-Load) procedures that are able to feed the data warehouse, a graphical reporting dashboard for inspection of the compliance state, and a root cause analysis tool. We call these components collectively compliance governance infrastructure.

6.3.6. Provide specific support for monitoring and management of compliance.

Compliance has been taken into account in four different ways: by jointly agreeing on a set of events from which it is possible to assess whether a process instance has been executed compliantly or not, by storing the respective compliance requirements in the data warehouse, by equipping the dashboard with compliance-specific navigation paths (from coarse requirements to low-level events), and by implementing a decision tree mining algorithm that identifies correlations between business data produced during process execution and compliance evaluations.

6.3.7. Provide offline governance of compliance through mining and analysing logs.

The **analysis of non-compliant situations** has been addressed by two root cause analysis techniques: decision tree mining and business protocol mining. The decision tree approach is able to identify whether there are dependencies among the data exchanged during the execution of a process instance and the final compliance assessment of a process. Once a dependency is identified, such can be used for two purposes. First, the dependency may allow the process analyst do trace back non-compliant situations to their root cause. Second, the decision tree can be used to predict likely compliance assessments already during the on-going execution of a process. The protocol mining approach complements this technique. It aims at reconstructing a so-called protocol (the logic of the exchanged messages in a process) from the event log. As such, it allows the process analyst to check whether a deployed process is really been executed as expected by its design.

6.3.8. Provide tool support and integration in the COMPAS architecture.

The **integration of the compliance governance infrastructure with the overall COMPAS architecture** occurs via two main channels: via the event log and via the MORSE repository. The event log collects all runtime data that is necessary to assess the compliance of executed process instances. The MORSE repository contains all the process models and compliance requirements that are necessary to interpret collected events and to support the compliance-centric navigation through the data in the data warehouse. The specific tool that performs this last interpretation and preparation of the data is the ETL procedures.

6.3.9. Concept of reusable process artefacts to assure compliance of business processes and service compositions.

The concept of **process fragments** to ease the task of compliant process design by reusable building blocks has been developed in the COMPAS project. We applied the concept of process fragments to implement the compliance requirements related to process activities and control flow. To ensure a correct integration in the process, we included concepts on compliance rule formalization developed by University of Tilburg and the approach on compliance verification by CWI Amsterdam. Using this combination of techniques, compliant business process design is achieved: Compliance rules can be captured using fragments and these fragments can be included in a process without breaking the compliance modelled by the fragments.

6.3.10. Language and runtime support for reusable process artefacts.

We developed language extensions to BPEL in order to support the specification of process fragments for compliance. Process fragments have been implemented using these extensions and have been successfully integrated into the process models of the use cases. To achieve most possible impact, we prepared a standardization proposal for these extensions. Process fragments for compliance are integrated into the process model during design time, i.e. before its execution. After integration, a standard process model without extensions for compliance fragments can be generated. Therefore, this concept supports reusable process artefacts, but it does not require an extension or modification of the process engine as standard process models are executed.

6.3.11. Tool support and integration in the COMPAS architecture.

Besides tools that provide the “glue” for integration in the COMPAS architecture, we mainly developed three components to support the concepts of process fragments and compliant business process execution. (i) For design-time support of process fragments we developed the fragment-oriented repository *Fragmento*. This repository provides advanced functions for the management of process fragments for compliance, e.g. a process stored in Fragmento can be annotated with security policies or with process fragments that constrain its behaviour. (ii) To support traceability during execution, we developed an *extension of the eventing functionality of the open-source process engine Apache ODE*. The need to address *traceability has been identified as crucial* in an early stage of COMPAS. Traceability denotes the property of being able to trace a requirement throughout the process lifecycle. It forms the bridge between compliance requirements from design time and compliance violations from runtime and thus enables drill-down of violations to their origin. At runtime, this traceability information is emitted in execution events by the extended process engine. These events contain Universally Unique Identifiers (UUIDs) of the process model, process instance, process activities, an event type, and optional further properties. (iii) To support monitoring the execution of a process instance based on a process graph we developed the Web-based monitoring tool **Business Process Illustrator (BPI)**. This tool allows for following the execution of a process, while abstracting from details of little importance for understanding, e.g. hiding of fault handlers or assign activities. Furthermore, we can use this monitor to highlight those fragments in a process that are related to compliance.

In summary, the concepts and tools we have developed meet our expectations. We have implemented our approach in COMPAS infrastructure components. The presented concepts cover the compliance requirements related to the internals of a process. Our work on case studies revealed that this is a subset of the compliance requirements. Thus, other actions have to be taken in addition to provide a holistic compliance management.

6.3.12. Designing concepts for expressive languages based on the DSL and specification language concepts.

A major issue to enable the effective management and enforcement of compliance requirements is to decouple compliance specification from business process specification. Compliance requirements should be organized and represented at various levels of abstraction to accommodate different stakeholders' needs. Decoupling involves the specification and management of compliance requirements and all relevant concepts (e.g. risks, controls, compliance regulations and directives, etc.) as a separate entity – starting from abstract requirements to concrete and organization-specific rules – and requires them to be linked to the relevant business processes/fragments to enable their traceability. For this purpose, a conceptual model has been developed, where compliance requirements and all related concepts can be organized, stored and maintained, enabling their usability and traceability.

6.3.13. Developing an expressive language for compliance concerns.

The main objective of WP2 is to provide an expressive language for compliance requirements; we name it "Compliance Request Language (CRL)". Compliance requirements should be based on a formal foundation of a logical language to pave the way for automatic reasoning and analysis techniques that assist in verifying and ensuring design-time business process compliance. In this aspect, we make use of process verification tools against formal compliance rules. We have analysed a wide range of compliance legislations and frameworks including Basel II, Sarbanes-Oxley, IFRS, FINRA, COSO, and COBIT, and examined a variety of relevant works on the specification of compliance requirements. Our analysis identified a set of features that CRL should possess, such as expressiveness, usability, non-monotonicity, intelligible feedback, etc. Based on these findings, we have conducted a comparative analysis between a set of formal languages that are candidates to serve as the formal foundation for the CRL. The comparative analysis put more strength on temporal logic mainly because of its maturity and the availability of its associated sophisticated verification tools that have proven to be successful in the verification of various large-scale systems. In particular, we have adapted Linear Temporal Logic (LTL).

We have introduced the meta-model and the grammar of Compliance Request Language (CRL) that is grounded on LTL and property specification patterns (cf. Dwyer et al. 1998, Property Specification Patterns for Finite-State Verification), which are high-level abstraction of frequently used temporal logic formulas. Patterns are intended to solve one of the major problems relevant to the usage of formal languages: the usability. In addition to the original patterns, we have also identified and introduced a set of compliance patterns to capture recurring requirements in the compliance context. CRL enables the user to build pattern-based representations of compliance requirements and based on the mapping rules from patterns to LTL, formal compliance rules are automatically generated using the tools we developed for this purpose.

Based on these foundations, we also proposed an approach to identify root causes of compliance violations during design-time, to provide remedies as guidelines/ suggestions that can help the business and/or compliance experts to resolve compliance deviations. Identifying the root-causes of violations and providing the experts with appropriate guidelines to resolve non-compliance is an important issue that should be considered and integrated in a comprehensive compliance management solution.

6.3.14. Design and implement tools and a supporting infrastructure that helps users to use the expressive languages for compliance concerns.

We developed the integrated environment – Compliance Request Language Tools (CRLT) together with the Compliance Requirements Repository (CRR), where data maintained by the CRLT resides. We also described how the CRLT is integrated with the COMPAS Architecture. The CRLT (http://eriss.uvt.nl/compas) integrates with the Model Repository and Process Verification Tools and comprises components that allows the definition and management of the compliance requirements; handling of interactive user specified compliance requests in a compliance language (design-time verification of the compliance targets), and the design of visual representations of compliance requirements using patterns for the automated generation of formal compliance rules.

6.3.15. A monitoring framework based on message abstraction.

This abstraction is called business protocol. We provide an extension of XPath to accommodate verification issues. The resulting language (called BPath) is also a query language that can be used to track and make visibility on business process execution. First, a BPEL business process specification is transformed into a business protocol. Then, monitoring properties and queries are formulated using BPath monitoring language over the business protocol. At runtime, all incoming or outgoing messages are captured by the business protocol monitor component before reaching their original destination. The process engine as well as the monitoring framework will publish respectively the execution and monitoring events, which are stored in the execution log. The execution log is of two types: states log, generated by the business protocol monitor, and events log generated by the process engine. We have implemented this approach and shown its relevance on some scenarios.

6.3.16. Automatic extraction of communication protocols.

Model extraction and mining could also help to discover the behaviour of a running model implementation using its interaction and activity traces. Process monitoring handles the tracking of individual processes in order to extract activity and execution information. Process mining, sometimes named offline or post-mortem monitoring, is used to analyse the event logs instead of the runtime process instances. The result of the analysis is then compared to the existing system models, and can either result in model updates or – where suggested by assessment – result in some corrective actions to overcome such discrepancies. We investigated extraction approaches by resorting to linear algebra. The proposed methodology allows us to extract the business protocol while merging the classic process mining stages. On the other hand, our protocol representation based on time series of flow density variations makes it possible to recover the temporal order of execution of events and messages in the process. In addition, we proposed the concept of proper timeouts to refer to timed transitions, and provide a method for extracting them despite their property of being invisible in logs. The approaches have been implemented in the form of prototype tools, and experimentally validated on scalable datasets.

6.3.17. Semantic-based elicitation in business processes.

Modelling Web services is a major step towards their automated analysis. One of the important parameters in this modelling, for the majority of Web services, is the time. A Web service can be presented by its behaviour that can be described by a business protocol representing the possible sequences of message exchanges. Automated analysis of timed Web services such as compatibility and replaceability checking are very difficult and in some cases are not possible with the presence of implicit transitions (internal transitions) based on time constraints. The semantics of the implicit transitions is the source of this difficulty because most of well-known modelling tools do not express this semantics (e.g., epsilon transition on the timed automata has a different semantics). We investigated an approach for converting any protocol containing implicit transitions to an equivalent one without implicit transitions before performing analysis. The proposed approach was implemented.

6.3.18. Graphical environment for service description.

One of COMPAS challenges was the models and tools for specifying and verifying compliance requirements. In our approach, we employed the Reo coordination language to graphically specify service composition glue code and coordinate message exchanges among individual services involved into a process. As our work on compliance source analysis showed, compliance requirements influence various classes of system properties, i.e., control flow temporal constraints, data-centric requirements, time-related properties, probabilistic properties, etc.

Therefore, we extended our graphical environment with annotation tools that allow designers to enrich process specification with necessary information, e.g., define service input/output messages, specify data-dependent branching conditions and functional transformations on process dataflow indicate channel delays, and task timeouts. We extended the initial set of Reo channels supported by our tools with primitives necessary for data manipulation, possibility to create user-defined channels and build hierarchical workflow models. Depending on which set of channels is used for process modelling, its semantics in a form of extended constraint automata is obtained automatically, while the modelling environment is the same regardless of what kind of property we target at verification time.

6.3.19. Formalisation of business process models.

Since service developers may use various notations for process specification, we developed tools for automated conversion of several workflow modelling languages to their formal representation in Reo, which precisely describes control and dataflow in a business process and, thus, disambiguates the initially informal workflow specifications. Given this translation, multiple verification tools, both developed within the scope of the COMPAS project and external tools, can be used for automatic analysis of various classes of formalised compliance requirements.

6.3.20. Formal specification and automated verification of compliance requirements.

At the very low level of abstraction, compliance requirements are represented by logic properties that should hold for a certain system specification. Theoretical computer science offers many well-establish formalisms for specifying system properties. We chose the mu-calculus model as the most expressive logic formalism that subsumes many of the logics used in system verification, including LTL and CTL. To enable model checking of formalised dataflow specifications, we implemented a tool for generating mCRL2 code for a given graphical process model. This allows us to apply the whole range of available state-of-the-art model checking, simulation, visualization and optimization tools and verify the validity of compliance requirements expressed in the form of mu-calculus formulae.

6.3.21. Integration of process verification tools with COMPAS architecture.

For the integration with the overall COMPAS architecture, we developed a set of services for exchanging information about system properties and the results of process verification with template-based property specification tools and service repositories where the process models are stored for reuse, adaptation and code generation. These tools help developers to connect compliance source documents with actual requirements represented in a form of logic formulas to be verified using simulation and model checking techniques.

6.3.22. Develop thought leadership on compliance issues around SOA.

As mentioned by Stuttgart University it was essential to define an adequate conceptual model of compliance in a service-oriented architecture (SOA), because of the model-driven approach. We helped our consortium partners to define this model by providing input from the aspect of compliance whereas they provided input from the technical (SOA) point of view. By combining these two aspects we gained new insights on how to address compliance issues in a SOA. We used our new insights from the COMPAS project to organize round table sessions with some of our PwC relations and we are processing our gained knowledge in articles.

6.3.23. Provide the industrial partners with practical information that can be used to create, perform and evaluate the case studies.

As compliance experts we were involved as an industrial partner. By performing extensive iterative reviews on the case studies from a compliance point of view we were able, together with Telcordia and Thales, to come up with two realistic use cases with sufficient compliance challenges that would be addressed by the COMPAS prototype tooling.

In addition we also provided (review) input on the COMPAS prototype tooling where interaction was needed with either a business user (e.g. Compliance Officer, Business Process Manager, etc.). We did this mainly for the tooling developed by Tilburg University (Compliance Request Language Tool) and Trento University (Compliance Governance Dashboard).

6.3.24. Use the COMPAS results to help SOA enabled organizations with compliance.

When the COMPAS project is finalised we are interested in how the results of the project (e.g. tooling) can be used for commercial use and exploitation and what services can be offered to clients in this area. E.g. how can COMPAS tooling help the auditor perform his work at an organisation with a SOA environment, and what services can we offer to our clients that have a SOA and want to address various compliance requirements and issues.

7. Availability of Results

Project results are available from the website http://compas-ict.eu. The following prototypes have been developed or used by the project:

BPI Business Process Illustrator

Business Process Illustrator (BPI) is a Web-based tool for monitoring the execution of business processes. It allows to a view a graph of a process model enriched with status information of a process instance. The process graph is refreshed regularly. Additionally the user can adapt the graph by highlighting or omitting activities. The source code, binaries, and installation manual are available for download: http://sourceforge.net/projects/bpi/.

CGDs Compliance Governance Dashboards

Compliance Governance Dashboards (CGD) aims at reporting on compliance, creating an awareness of possible problems or violations, and facilitating the identification of root-causes for noncompliant situations. For that, CGD concentrates on the most important information at a glance, condensed into just one page. For more information on CGD please visit the CGD Web site: http://compas.disi.unitn.it/CGD/home.html.

CRLT Compliance Request Language Tools

Compliance Request Language Tools (CRLT) serves two main purposes. First, it offers the interface for the Compliance Requirements Repository to define, store and maintain compliance requirements in various abstractions together with related aspects such as compliance risks, sources, controls and rules. Second; it enables compliance and business experts to formulate compliance requests at design time for checking end-to-end business processes and process fragments against formalized regulatory compliance requirements. For more information on CRLT please visit the CRLT Web site: http://eriss.uvt.nl/compas/.

Eclipse Coordination Tools

Eclipse Coordination Tools (ECT) is a framework for verifiable design of component and service-based software using the coordination language Reo. Reo presents a paradigm for composition of distributed software components and services based on the notion of mobile channels. Software application designers can use Reo as a "glue code" language for compositional construction of connectors that orchestrate the cooperative behaviour of components or services. The ECT framework consists of a set of integrated tools that are implemented as plug-ins for the Eclipse platform. ECT provides functionality for converting high-level modelling languages such UML, BPMN and BPEL to Reo, for editing and animation of Reo models, synthesis of automata-based semantical models from Reo, annotation of Reo and automata with QoS constraints and verifying these models using dedicated model checking tools. ECT is an open source project. For more detail and the information how to participate in the development, please refer to the Reo Web site: http://reo.project.cwi.nl/.

FRAGMENTO

Fragmento is a Fragment-oriented Repository that is dedicated to the management of process-related artefacts, such as BPEL processes, WSDL documents, deployment descriptors, and especially, process fragments. Fragmento provides particular functionality in addition to the basic repository functionalities for handling process artefacts (persistence, storage, search, retrieval, version management). Fragmento provides XML schema validation, and provides an extensibility mechanism for integration of additional validation functions. Furthermore Fragmento provides an extensibility mechanism for custom query functions. This allows the implementation of search functions beyond the metadata of a process artefact (e.g., concerning the structure of a process fragment). Fragmento also provides mechanisms for definition of bundles, which allows packaging all artefacts related to a process (or fragment) together into one package. Fragmento is released as

Open Source during the year 2010. For more information on Fragmento please visit the project Web site: http://www.iaas.uni-stuttgart.de/forschung/projects/fragmento/start.htm.

The Model-Aware Repository and Service Environment (MORSE) is a service-based environment for the storage and retrieval of models and model-instances at both design- and runtime. Models and model-elements are identified by Universally Unique Identifiers (UUID) and stored and managed in the MORSE repository. The MORSE repository provides versioning capabilities so that models can be manipulated at runtime and new and old versions of the models can be maintained in parallel. For more information on MORSE please visit the MORSE Web site: http://www.infosys.tuwien.ac.at/prototype/morse.

The Pluggable Framework for Apache ODE extends the Apache ODE BPEL engine to support a generic eventing framework. The eventing framework consists of generic events and architecture for handling the events. The events are tailored towards BPEL, but independent of the concrete engine used. That means, the BPEL engine can be exchanged with another BPEL engine and the events remain the same. Therefore the code dealing with the events does not need to be changed. This is a basis for a BPEL monitoring infrastructure being engine independent. For more information on ODE-PGF please visit the project Web site at: http://www.iaas.uni-stuttgart.de/forschung/projects/ODE-PGF/.

The View-based Modeling Framework (VbMF) provides flexible, extensible methodology and tooling for modelling, developing, and maintaining business processes based on the notion of view models – a realization of the separation of concerns principle, and the model-driven development paradigm – a realization of the separation of abstraction levels. The core concepts of the framework are extended or refined to represent and integrate business compliance concerns. Finally, process implementation, deployment configurations, runtime monitoring directives, and so on, can be automatically generated from view models. For more information on the View-based Modeling Framework please visit the VbMF Web site: http://www.infosys.tuwien.ac.at/staff/htran/#software.

8. Potential Impact of the Results

The impacts of the project can be categorized as follows: scientific impact in respective research communities and industrial impact.

8.1. Scientific Impact

The COMPAS project was one of the first international research efforts to focus on compliance management in service-oriented architectures. Thus, the scientific publications as produced within the scope of the project are very likely to develop a high impact for the community and future research regarding a holistic approach of compliance management in an IT context.

But not only the topic of compliance management will be affected by the COMPAS research project: in order to accomplish the major goal of a compliance IT framework, various fundamental research topics such as in model- and DSL-engineering, business process management, runtime monitoring, model checking and verification, and data-mining had to be studied. Numerous results within these fields have already been published in scientific articles and papers.

8.2. Industrial Impact

The COMPAS project was a research project that conducted fundamental research on compliance management in services-oriented architectures and as such presented a first endeavour to study and address this problem. With this in mind, an industrial application would not be to be expected from such a project, thus. Yet, the results and prototypes developed within the scope of the project promise early application in an industrial context: the modelling approach that takes place at various levels of abstraction contributes a conceptual solution to specify and document compliance and relate concerns to IT. Similarly, the monitoring infrastructure depicts a general enough IT architecture for the runtime system. In short, the results of the COMPAS project clearly describe approaches for realizing compliance management – particularly in an industrial context.

As compliance management increasingly pervades business processes – not only at large but also small and middle-sized companies, it is expected that solutions have to be realized and applied to realize compliance management in an industrial context. The results from the COMPAS project directly worked towards addressing this need and thus are expected to gain momentum in impact.

9. Lessons Learned During the Project

One of the lessons learned throughout the project is that reporting on compliance is not as easy as it could seem in the first place. The process-centric approach of COMPAS requires combining the concerns of two different stakeholders, i.e., process analysts/owners and compliance experts, in the same graphical user interface (GUI). We could also have simply opted for two different, independent views for the two roles, but the discussions throughout the whole project have shown that compliance is a crosscutting concern that most of the times requires strong cooperation of process analysts and compliance experts. Therefore we decided to merge both views into one GUI.

The process-centric approach of COMPAS is further very strong in managing process-related compliance requirements, that is, compliance requirements that are related with the structure and timing of individual tasks/service invocations inside a process, while it is less strong in the identification of data-related compliance requirements (e.g., checking the conformance with a given data format). As a consequence, if there are no explicit data checking tasks in the process, only seldom the decision tree algorithm is able to identify relationships between compliance outcomes and business data. Yet, if there are such activities in the process, the algorithm performs very well. As an extension of the COMPAS approach, it could therefore be a good idea to extend its process-centric approach also toward data-centric compliance concerns.

We soon realized that the evolution of the compliance conceptual model needs to be supported throughout the project. For easing the co-evolution of dependent systems after a model change we fully automated the generation of storage and information retrieval service provider and requester agents in MORSE.

The concept of process fragments for compliance was intended to address compliance requirements which are related to control flow and activities within a process. Process fragments are the right choice for implementing compliance requirements that prescribe what should be executed. Integration mechanisms for process fragments allow integrating such compliance functionality into a process. When looking beyond the internals of a process, however, compliance management comprises more aspects. A process may orchestrate services and may involve people, but it cannot control them. For instance, requirements related to a data storage used by a service cannot be captured with the aid of a process fragment. To give an example: a requirement demanding for encrypted storage of loan request data for at least ten years is related to a database and is out of the scope of control of the process. Although these limitations exist, the approach is well applicable to formulate and integrate process structures that help achieving compliance.

Compliance entails different aspects of business processes and requires knowledge of various domains. Our analysis of various compliance sources and frameworks identified several compliance concerns where automated verification of relevant formal requirements can only ensure partial compliance as these constraints require human intervention in the form of manual checks, reviews, assessments, etc. for guaranteed assurance. We have also identified several concerns that involve requirements relevant to the retention of records, data encryption, etc., which typically crosscut business processes. In general, these requirements handled through the use of dedicated IT solutions and are not encoded in business process specifications. Hence, the assurance of such requirements is hardly possible with our solutions that focus on requirements that are represented within business process specifications and applicable for design-time phase of the business process compliance.

Design-time is the first step for ensuring compliance during the entire business process lifecycle. Dealing with compliance starting from the business process analysis and design phase is critical as identifying and solving compliance problems in the early phases is less costly than corresponding checks at later phases. However, it is not always feasible to enforce compliance with all constraints imposed on a process models at design time. There are limitations on the aspects that could be fully ensured during design time. For example, during design-time typical segregation-of-duties requirements can only be partially addressed, as such requirement typically demands runtime information for guaranteed compliance.

The type and coverage of the formal compliance rules that can be used for automated compliance verification and monitoring depend not only on the expressive power of the language used for their specification but also on the extent of the information encoded within business process specifications. For example, a compliance rule implementing a control that involves roles or other organizational units cannot be verified if the process specification under consideration does not incorporate process elements that capture these aspects. Thus, the granularity and formality level of the specifications in different phases of the lifecycle and the languages used for their specifications pose limits on the rules that can be used for their verification and monitoring.

From our perspective, one of the successes of the COMPAS project is providing a workable context within which the partners were able to apply theoretical results and formal methods on real, industrial problems. It is a fact that the gap between formal tools and techniques and real-life industrial problems is vast. Generally, complaints from both camps abound: practitioners are often disappointed by the limitations of theoretical results, the shortcomings of formal tools, and the seeming ignorance or indifference of theoreticians and the developers of formal methods and tools about their real world applications; the theoreticians and the developers of formal methods and tools, in turn, are often discouraged by the seeming inability or unwillingness of practitioners to adopt their arcana to express themselves.

Still, the fact remains that there is no escape out of this dilemma: formal tools and techniques are relevant only because they (aspire to) tackle real problems; and the scale and complexity of real problems are well beyond the realm of human intuition or informal techniques. Through our work in COMPAS, we learned that the so-called gap between formal methods and industrial applications is often not a void that one may hope to eventually bridge over by either forcing the practitioners to express themselves in the arcana of formal methods, or encouraging the formal people to become application domain experts. We learned that this so-called gap is in fact a vast terrain full of unexpected, non-trivial problems that need to be discovered first, before they can be solved. Both discovery and solving these problems requires a collaborative exploration of this vast terrain by domain experts along side their more formal colleagues.

The predefined structure of work in the project makes it hard to focus on fundamental research issues that emerge beyond the scope of the initially planned tasks. Due to the fact that all components of the architecture had to be provided on time, we had to forego producing some pieces of functionality that would be useful for the static analysis of COMPAS case studies. Specifically, we believe performance evaluation and verification of probabilistic and stochastic properties of business processes are useful. With such functionality, for example, we can tackle specification and verification of non-functional properties of workflow models, such as checking whether a process meets the performance requirements specified in a service-level agreement. To this end, we developed a compositional automata-based model for behaviour specification that enables automated reasoning about provisioned QoS. However, we had to postpone the development of an actual tool based on this model, due to the need to deliver service simulator engines.

Although COMPAS has been a successful project and delivered all the necessary results, it still requires much effort to produce a complete set of tools which could be applied universally to any domain. Complete automatization (without much of human involvement), more mature and universal language models and enforcement capabilities are still needed to be worked on. Therefore, the major lesson learned is that COMPAS is just a first step to provide universal solution for compliance in SOA/Business Processes world. New projects have to be developed to continue on the solid foundation built by COMPAS. The areas like Domain Specific Languages design or compliance monitoring are so broad that they would need to be continued in new separate projects. It was impossible to prepare a perfect set of languages for any domain which could be universally integrated.

10. Partners

Technische Universität Wien, Coordinator

Schahram Dustdar, Ta'id Holmes, Emmanuel Mulo, Ernst Oberortner, Huy Tran, Uwe Zdun

http://www.infosys.tuwien.ac.at/

Università degli Studi di Trento

Aliaksandr Birukou, Fabio Casati, Vincenzo d’Andrea, Florian Daniel, Patricia Silveira, Soudeep Roy Chowdhury

http://disi.unitn.it/

Universität Stuttgart

Katharina Görlach, Frank Leymann, Dimka Karastoyanova, Oliver Kopp, Thorsten Scheibler, David Schumm, Steve Strauch

http://www.iaas.uni-stuttgart.de/

Stichting Katholieke Universiteit Brabant

Amal Elgammal, Mike Papazoglou, Oktay Türetken, Willem-Jan v.d. Heuvel, Mathijs v.d. Paauw

http://www.tilburguniversity.edu/

Université Claude Bernard Lyon 1

Salima Benbernou, Emmanuel Coquery, Fabien de Marchi, Emad Elabd, Mohand-Said Hacid, Kreshnik Musaraj, Mustapha Meziane, Samir Sebahi

http://liris.cnrs.fr/

Centrum voor Wiskunde en Informatica

Farhad Arbab, Behnaz Changizi, Natallia Kokash

http://www.cwi.nl/

Thales Services SAS

Pascal Bisson, Phong Cao, Cyril Dangerville, Daniel Gidoin

http://www.thalesgroup.com/

Telcordia Poland

Andrzej Cichocki, Dimitrios Georgakopoulos, Jacek Serafinski, Marek Tluczek, Agnieszka Cavalcante

http://www.telcordia.com/

Pricewaterhousecoopers Accountants N.V.

Wim Hutten, Zouhair Taheri

http://www.pwc.com/nl

Members of the consortium who presented the COMPAS results during the second review meeting in Brussels:

11. Contact

Prof. Schahram Dustdar
Technische Universität Wien
Distributed Systems Group
Vienna, Austria

Phone +43-1-58801-18414
Fax +43-1-58801-18491
E-mail dustdar@infosys.tuwien.ac.at