Developing context-sensitive software infrastructures

The software of the future that will adapt to changing requirements is the overall focus of the 'Role-based Software Infrastructures for continuous-context-sensitive systems' (RoSI) Research Training Group, led by Professor Dr-Ing Wolfgang Lehner and Professor Dr Uwe Aßmann, at the Faculty of Computer Science at the Technische Universität Dresden, Germany. The main aim of this research is to validate the feasibility of consistent role modelling as a concept for developing context-sensitive software infrastructures.

The increasingly dynamic environments of today's software systems imply that it is no longer feasible to expect human developers to always be able to react immediately to changing conditions. Software systems must be self-aware so that they can autonomously adapt their behavior to suit their environment. Your mobile phone, for example, should be able to automatically recognise when you are in a meeting and mute its alarm accordingly.

Role-based modelling is a promising approach for handling a software system's adaptivity and self-awareness. Role-based systems have already been established for roles such as application development and persistence, but these are isolated approaches that use the role-based model on their specific layer. There are, for example, authorisation systems based on different user groups and roles, relational role-mapping concepts to provide a role-based persistence layer, and a variety of methods to ensure role-based process management. Nevertheless, these are isolated approaches on the single levels of the complete software stack. To date, they have not been used consistently on all levels of abstraction in the software development process, through the modelling of concepts, languages, applications, and software systems.

The software of the future that will adapt to changing requirements is the overall focus of the Role-based Software Infrastructures for continuous-context-sensitive systems (RoSI) Research Training Group, led collaboratively by Wolfgang Lehner and Uwe Aßmann at the Faculty of Computer Science at the Technische Universität Dresden, Germany. The project is funded by the German Research Foundation, Deutsche Forschungsgemeinschaft, DFG.

RoSI RESEARCH TRAINING GROUP

RoSI is made up of ten Principal Investigators and five Junior Scientists. In addition, more than 40 PhD students have been involved in this major research project since its onset in 2013. The research training group combines excellence in research with a comprehensive qualification program for doctoral candidates. This program provides students with opportunities to exploit the Principal Investigators' international network, which allows for extended research stays abroad. The program also offers individual training by way of soft-skill courses. These are combined with meticulous supervision that ensures high-quality research results.

Software is the foundation of the modern economy and society. Professors Lehner and Aßmann explain that while the process of designing, deploying, and providing a broad range of runtime platforms is widely appreciated, current software lacks the proficiency to adapt quickly and consistently to changing environments. Within the RoSI project, the team aims to devise an innovative programming paradigm that goes beyond traditional object-orientation.
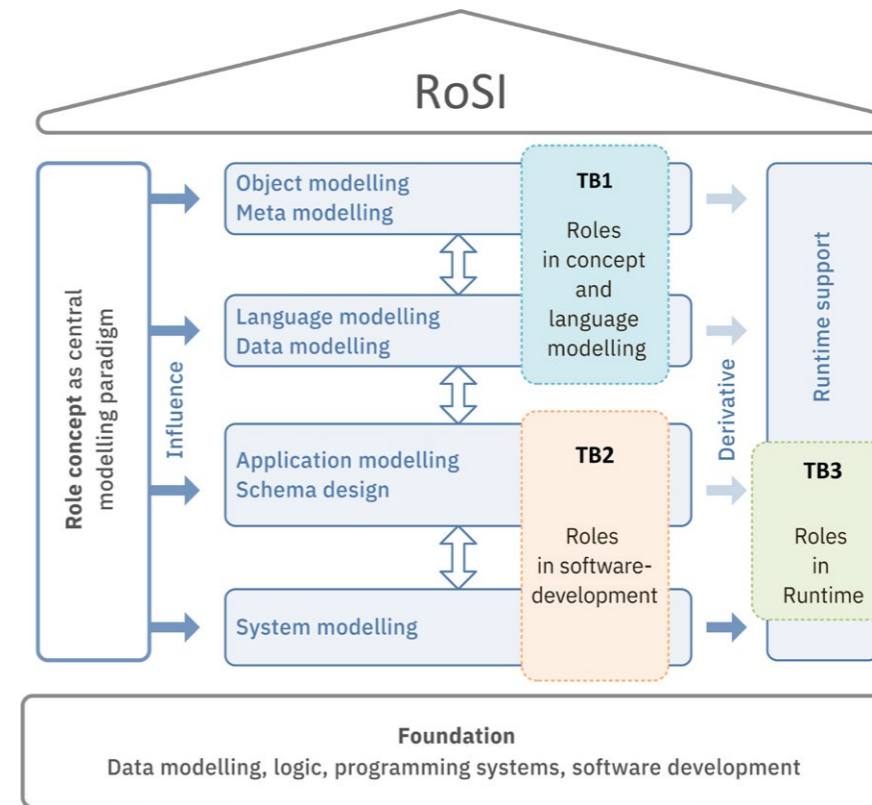

Figure 1. Research areas of role-based modelling.

This novel concept is based on roles that will adjust software to changing environments, such as mobility, and requirements, including long-living software. Furthermore, they will embrace the entire software life-cycle, from modelling, through programming and installation together with changes during runtime.

THE ROLE CONCEPT

Currently, the object-orientated programming approach means that software systems have only been defining a static relation between two objects. In the future, these objects will take on flexible roles. The role concept will enable changes of context to be modelled on various levels of abstraction. Identified role concepts will be systematically connected to each other to facilitate model transformations and synchronisations.

The role concept is based on the idea of separations of concerns, where a computer program is separated into smaller modules (or micro services) that deal with only one thing. Under the role concept, the core is separated from its fluent and context-dependent parts. Roles and the underlying primitives can be used to describe and implement context-dependent behavior and structure. Thus, role-based software modelling

Software systems must be self-aware so that they can autonomously adapt their behaviour to suit their environment.

and programming enables distributed and context-aware applications, capable of adapting their behavior based on changing environmental conditions and demands during runtime.

ROLE MODELLING

Validating the feasibility of consistent role modelling and its practical applicability is the principal goal of this research program. The researchers describe how this encompasses the concept modelling (in meta-languages), the language modelling, and the modelling on the application and software system level. Additionally, scientific elaboration of the role concept is required to enable modelling of the change of context on different levels of abstraction.

Consistency means that roles are used systematically for context modelling on all levels of the modelling process. This offers substantial benefits to software systems engineering because context changes are interrelated on different levels of abstraction. Moreover, they can be developed and maintained concurrently. This lays the foundations for cutting-edge software-architectures that satisfy the needs of future dynamic and heterogenous software systems. The numerous potential applications include future smart grid, natural energy-based computing, cyber-physical systems in home, traffic, enterprise resource planning software, and context-sensitive search engines.

RESEARCH GOALS

The research team has identified four aims that contribute to the main research goal. Firstly, the universal concept of roles is formalized to show that it is a viable model for developing context-sensitive software infrastructures. Then the dynamic aspect of roles is explored. The third aim involves investigating the consequences of the new attributes of role-based software across its lifetime. This requires the development of novel concepts to cope with the dynamic processes. Here, research centres on how the context-specific aspects of the role-concept influence the process of developing software and the software duration. Finally, context-sensitive role-modelling is evaluated employing practical examples. This will both help to illustrate the results and present the advantages of these concepts.

The research training group have already developed several software tools:

CROM

The **C**ompartment **R**ole **O**bject **M**eta-model (CROM) is a comprehensive role-based modelling and programming language that incorporates the context-dependent and relational nature of roles together with model constraints. In addition to the meta-model that provides its abstract syntax, this repository also includes a code generator for the modelling language, formal CROM.
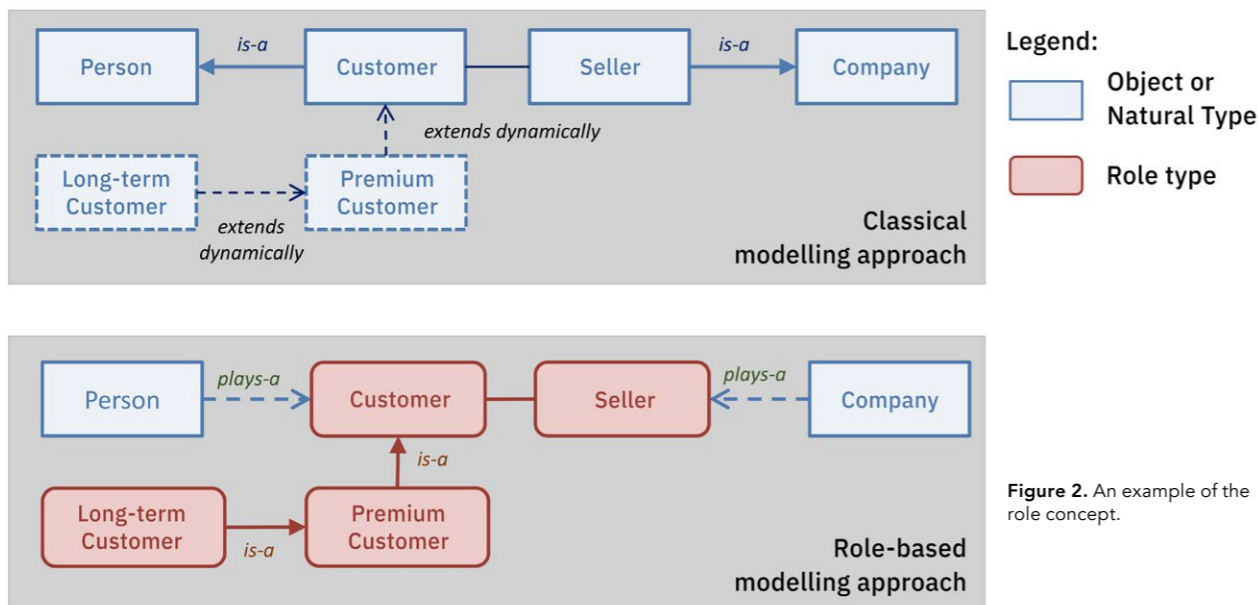
Legend:
- Object or Natural Type
- Role type

Classical modelling approach

Role-based modelling approach

Figure 2. An example of the role concept.

FRaMED

The **F**ull-fledged **R**ole **M**odeling **ED**itor (FRaMED) enables the graphical specification of role-based software systems. It supports the creation and modification of natural types and compartment types, as well as the creation and modification of role types and relationships within a compartment type. It also supports a variety of constraints on both roles and relationships.

InVerDa

Regardless of changing database structure, existing data and software system functionality must be maintained.
InVerDa is a tool for integrated, robust, and easy to use database versioning that allows a single database to have multiple co-existing schema versions. Developers can implement schema modifications to add a new schema version to an existing database schema. The new schema versions become available immediately. Applications can read and write data through any schema version simultaneously. What is written in one version is also reflected across all other versions. In addition, the database administrator can decide which schema version should be configured for the data to primarily materialise. Should the workload mix change, the database administrator

can change the materialisation independent of the evolution and not affect the availability of any schema version.

SCROLL

SCROLL (**SC**ala **ROL**es **L**anguage) is an embedded scala implementation for roles. Scala is a high-level language that combines object-oriented and functional programming. SCROLL was developed as a lightweight library approach, where the user can specify both roles and context dependent behavior. This library facilitates the easy integration of legacy code and a high separation of concerns.

Role-based methodology is the foundation for next-generation programming concepts, languages, and runtimes.

It also allows role-based implementations that lead to more reuse and better separation of concerns.

SELF-RECOVERABLE DATA STRUCTURES AND OTHER CONSIDERATIONS

Modern hardware developments such as non-volatile RAM, hardware transactional memory, remote direct access memory and heterogeneous computing cores require the architecture of database systems to be rethought so that these technological developments can be used. The research being carried out by the RoSI Research Training Group focuses on the influence of non-volatile RAM on database

architecture. Non-volatile memory retains stored information even after the power supply is removed. The research team is particularly interested in the development of data structures for systems with non-volatile RAM. This includes developing hybrid data structures that use both transient storage – that is, temporary storage for non-permanent data – and volatile storage that requires power to preserve stored data.

The aim is for these data structures to be 'self-recoverable' so that they can be used for both reading and writing after a restart with minimal recovery effort. During normal operation, a possible recovery should not incur a significant overhead. Furthermore, a generally reduced concurrency, ie, a drop in ability to allow multiple users to affect multiple transactions, should not arise.

The research team is also considering 'Any-Point Crash Recovery', the detection of incorrect states, methods for increasing concurrency or 'Selective Concurrency', as well as fine-grained replication for hybrid data structures. Professors Lehner and Aßmann remark that the RoSI Research Training Group puts particular emphasis on experimental evaluation and system integration for end-to-end scenarios.

To find out more, visit rosi-project.org

Behind the Research

Uwe Aßmann
E: uwe.assmann@tu-dresden.de

Wolfgang Lehner
E: wolfgang.lehner@tu-dresden.de

Research Objectives

The Role-based Software Infrastructures for continuous-context-sensitive systems (RoSI) project aims to develop a novel programming paradigm that adjusts software to changing environments and requirements, throughout the complete lifecycle of software.

Detail

Address

Professor Dr-Ing	Professor Dr
Wolfgang Lehner	Uwe Aßmann
Database Research Group	Software Engineering
Technische Universität Dresden	Technische Universität Dresden
Faculty of Computer Science	Faculty of Computer Science
01062 Dresden	01062 Dresden
Germany	Germany

Bio

Wolfgang Lehner is Head of the Dresden Database Research Group at TU Dresden, Germany and spokesperson of RoSI. He is member of the Senate Committee and Grants Committee on Collaborative Research Centres of DFG and member of The Academy of Europe as well as Managing Director of the Proceedings of VLDB (PVLDB). Following his PhD and habilitation at the University of Erlangen-Nürnberg, he held multiple visiting researcher positions in the US (IBM, Microsoft, SAP) and Canada (University of Waterloo). His main research expertise lies in the areas of scalable data management systems focusing on cloud-centric Big Data infrastructures and the design of data-intensive applications for composable hard/software-systems. He is active in many international program committees, gives keynotes, tutorials, and leads a community-wide effort to push Open Science in the context of reproducibility and availability of research artifacts.

Uwe Aßmann has been Professor for Software Engineering at the Faculty of Computer Science since 2004 and the Dean of Faculty since 2016. He obtained a PhD in compiler optimization and a habilitation on 'Invasive Software Composition', a software reuse technology for arbitrary program and modelling languages. He has taken part in the DFG Excellence Clusters, Center for Advancing Electronics Dresden (cfAED) and Center for Tactile Internet (CeTI), researching design techniques for robotic co-working (Lean Robotics). In the RoSI project, he contributes to RoSI's view-based language family for the entire software life cycle, to simplify the development of context-aware systems.

References

Kühn, T, Leuthäuser, M, Götz, S, et al, (2014). A Metamodel Family for Role-Based Modeling and Programming Languages. In: Combemale, B, Pearce, DJ, Barais, O, Vinju, JJ, (eds) *Software Language Engineering. SLE 2014. Lecture Notes in Computer Science*, 8706, 141–160 Springer, Cham. doi.org/10.1007/978-3-319-11245-9_8

Leuthäuser, M, & Aßmann, U, (2015). Enabling View-based Programming with SCROLL: Using Roles and Dynamic Dispatch for Establishing View-based Programming. *Proceedings of the 2015 Joint MORSE/VAO Workshop on Model-Driven Robot Software Engineering and View-based Programming, MORSE/VAO*, 25–33, New York, NY, USA, ACM. doi.org/10.1145/2802059.2802062

Jäkel, T, Weißbach, M, Herrmann, K, et al, (2016). Position Paper: Runtime Model for Role-Based Software Systems, *2016 IEEE International Conference on Autonomic Computing (ICAC)*, 380–387. nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-753022

Herrmann, K, Voigt, H, Rausch, J, et al, (2017). Living in parallel realities – co-existing schema versions with a bidirectional database evolution language. *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1101–1116. doi.org/10.1145/3035918.3064046

Kühn, T, Böhme, S, Götz, S, & Aßmann, U, (2015). A combined formal model for relational context-dependent roles. *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*, 113–124. ACM. doi.org/10.1145/2814251.2814255

Kühn, T, Kassin, KI, Cazzola, W, & Assmann, U, (2018). Modular feature-oriented graphical editor product lines. *Proceedings of the 22nd International Systems and Software Product Line Conference*, 1, 76–86. doi.org/10.1145/3233027.3233034

Weißbach, M, Chrszon, P, Springer, T, & Schill, A, (2017). Decentrally Coordinated Execution of Adaptations in Distributed Self-Adaptive Software Systems, *2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, 111–20. nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-752682

Personal Response

What has been the most rewarding research outcome from the RoSI Research Training Group to date?

❞ The Research Training Group provides an umbrella for interdisciplinary and internationally recognised research themes as well as comprehensive soft-skill training of junior scientists in the field of computer science. This is mirrored by the large number of excellent graduates of the Research Training Group. Many of them have gained leading positions in industry and academia. Moreover, the RoSI Research Training Group has achieved international visibility through an outstanding publication record in international journals and presentations at top-tier conferences, symposia, workshops, and through 'science communication' via various channels targeting the general public. Finally, the individual research projects within RoSI have led to a number of software components that cover a complete end-to-end view of designing, developing, deploying, and running context-aware software applications. ❞