The journal for the international
Ada community

Ada User Journal

Volume 37
Number 4
December 2016

Article from the Industrial Track of Ada-Europe 2016

Produced by Ada-Europe

Editor in Chief

Luís Miguel Pinho	CISTER/ISEP. Rua Dr. António Bernardino de Almeida 431, P-4200-072 Porto, Portugal. Tel: +35-22-834-05-02; Fax: +35-22-834-05-09; Email: AUJ_Editor@Ada-Europe.org

Ada User Journal Editorial Board

Dirk Craeynest	c/o K.U.Leuven, Dept. of Computer Science, Celestijnenlaan 200-A, B-3001 Leuven (Heverlee), Belgium. Email: Dirk.Craeynest@cs.kuleuven.be
Patricia López Martínez (Assistant Editor)	Grupo de Computadores y Tiempo Real, Universidad de Cantabria, Av. de los Castros s/n, 39005 Santander, Spain. Email: lopezpa@unican.es
Jacob Sparre Andersen (News Editor)	Jacob Sparre Andersen Research & Innovation, Vesterbrogade 148 Copenhagen, Denmark. Email: jacob@jacob-sparre.dk
Jorge Real (Deputy Editor)	DISCA-ETSINF Edificio 1G, Universitat Politècnica de València Camino de Vera s/n, 46022 Valencia, Spain. Email: Secretary@Ada-Europe.org

Ada-Europe Board

Tullio Vardanega (President) University of Padua	Italy
Dirk Craeynest (Vice-President) Ada Belgium & KU Leuven	Belgium
Dene Brown (General Secretary) SysAda Limited	United Kingdom
Ahlan Marriott (Treasurer) White Elephant GmbH	Switzerland
Luís Miguel Pinho (Ada User Journal) Polytechnic Institute of Porto	Portugal
Jorge Real (Member at Large) Universitat Politècnica de València	Spain

Ada-Europe General Secretary

Dene Brown
SysAda Limited
Signal Business Center
2 Innotec Drive
BT19 7PD Bangor
Northern Ireland, UK

Tel: +44 2891 520 560
Email: Secretary@Ada-Europe.org
URL: www.ada-europe.org

Information on Subscriptions and Advertisements

Ada User Journal (ISSN 1381-6551) is published in one volume of four issues. The Journal is provided free of charge to members of Ada-Europe. Library subscription details can be obtained direct from the Ada-Europe General Secretary (contact details above). Claims for missing issues will be honoured free of charge, if made within three months of the publication date for the issues. Mail order, subscription information and enquiries to the Ada-Europe General Secretary.

For details of advertisement rates please contact the Ada-Europe General Secretary (contact details above).

ADA USER JOURNAL

Volume 37
Number 4
December 2016

Contents

Spatio-temporal Model-Checking for Collective Adaptive Systems in QUANTICOL

V. Ciancia, D. Latella
Consiglio Nazionale delle Ricerche – Istituto di Scienza e Tecnologie della Informazione "A. Faedo", Pisa, Italy; email: {Diego.Latella, Vincenzo.Ciancia}@cnr.it
M. Loreti
Università di Firenze – Dipartimento di Statistica, Informatica, Applicazioni, Firenze, Italy and IMT Alti Studi, Lucca, Italy; email: Michele.Loreti@unifi.it
M. Massink
Consiglio Nazionale delle Ricerche – Istituto di Scienza e Tecnologie della Informazione "A. Faedo", Pisa, Italy; email: Mieke.Massink@cnr.it

Abstract

Spatial aspects of computation are becoming increasingly relevant when dealing with systems distributed in physical space. Traditional formal verification techniques are well suited to analyse the temporal evolution *of system models; however, properties of* space *are typically not taken into account explicitly. In this position paper we briefly review some of the recent developments of spatial and spatio-temporal model-checking in the context of the European research project QUANTICOL funded by the FET-Proactive programme on Fundamentals of Collective Adaptive Systems. We illustrate some typical applications of spatial and spatio-temporal model checking on collective adaptive systems and provide an outline for further developments.*

Keywords: Temporal Logics, Spatial Logics, Model-checking, Collective Adaptive Systems.

1 Introduction

The concept of smart cities is on the research agenda of many EU and other international institutions and think-tanks. Although not the only factor for success of smart cities, innovative ICT-based technology is seen by many as a key factor that would allow modern cities to reach or maintain a good and sustainable quality of life for their inhabitants, with timely and equitable distribution of resources. At the core of many proposals ranging from smart buildings and transportation to a smart electricity grid, is the transformation of a centralised system architecture and control to a much more decentralised and distributed design. Similar issues of optimal distribution and congestion avoidance play a role in smart transportation, whether based on public transport or community initiatives such as shared bikes.

The very fact that such systems are highly distributed and their adaptive behaviour relies on the tight and continuous feedback between vast numbers of consumers and producers, makes such systems typical examples of large scale collective adaptive systems (CAS). These are systems that consist of a large number of *spatially distributed* heterogeneous entities with decentralised control and varying degrees of complex autonomous behaviour. QUANTICOL[1] is a research project funded by the FET-Proactive programme on Fundamentals of Collective Adaptive Systems. It aims to develop novel quantitative analysis techniques to support the design and operational management of a wide range of collective adaptive systems, with particular focus on applications arising in the context of smart cities.

Spatial aspects of computation are becoming increasingly relevant when dealing with systems distributed in physical space. Traditional formal verification techniques are well suited to analyse the temporal evolution of system models; however, properties of space are typically not taken into account explicitly. The global behaviour of CAS critically depends on interactions which are often local in nature, and thus aspects of locality immediately raise issues of spatial distribution of objects.

One of the project's proposals to facilitate *reasoning* about spatial aspects of CAS is the development of *spatial model-checking*. Model checking has been widely recognised as a powerful approach to the automatic verification of concurrent and distributed systems (see [1] and references therein). It consists of an efficient procedure that, given an abstract model M of the system, decides whether M satisfies a logical formula Φ. Traditionally, such formulas are drawn from a temporal logic and used to verify temporal aspects of a system such as "there exists a run of the system that eventually reaches a state in which the queue is full". Such temporal logics have later been extended with probabilistic and stochastic notions allowing for the verification of properties such as "the probability is 0.1 that the system reaches a state in which the queue is full" [1, 2, 3]. In the context of the QUANTICOL project stochastic and probabilistic model-checking has been further extended to address large scale CAS using model-checking techniques based on fluid and mean field approximations originating from the area of statistical physics [4, 5].

---

[1] Web site: `www.quanticol.eu`

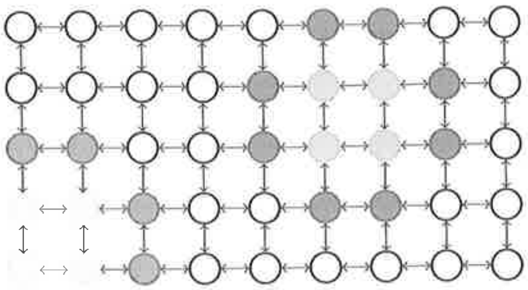

Figure 1: A graph inducing a *quasi-discrete* closure space

In spatial model-checking, instead, one is interested in verifying properties of *space*. Typical spatial properties concern questions of being near to a place satisfying a certain property, or of being reachable through space or of being surrounded by particular points. Spatial model-checking requires a spatial logic and a spatial representation on which such a logic can be interpreted, and, of course, efficient spatial model-checking algorithms. Furthermore, spatial model-checking can be combined with temporal model-checking leading to spatio-temporal model-checking. This gives rise to the verification of properties concerning the behaviour of a system in space and time. For example, in a collective system such as bike sharing one could then verify complex properties such as "eventually, when a station is full, there is a moment in the future in which all its adjacent stations will be full as well".

In this position paper we provide a brief overview of some of the recent developments on spatial model-checking in the context of the QUANTICOL project and some pointers to related publications.

2 Spatial Logic for Closure Spaces

The development of spatial logics dates back to the work by early logicians such as Tarski, who studied possible semantics of classic modal logics, using topological spaces. Topological spaces may be seen as generalisations of Euclidean spaces by focussing on the notion of *closeness* without making reference to an explicit metric. The field of spatial logics is well developed in terms of descriptive languages and aspects such as computability and complexity [6], but does not yet address formal verification problems. In particular, discrete spatial models are still a relatively unexplored field. One of the spatial logics, the Spatial Logic for Closure Spaces (SLCS) [7], proposed in the context of the QUANTICOL project is based on so-called Closure Spaces. These are a generalisation of topological spaces that include both continuous and discrete spatial models, among which the widely used discrete mathematical structure of graphs. Graphs are extremely versatile. Their use includes, for example, the representation of digital images. Graphs give rise to the subset of Closure Spaces which are known as Quasi-discrete Closure Spaces [8].

The logic SLCS builds on the tradition of modal logics and on the modal logics approach to spatial logics in which the two well-known modalities *possibility* and *necessity* are given a topological interpretation, namely that of *closure* and its dual *interior* (on the reals or a similar metric space) [6]. In SLCS both modalities have been given an interpretation suitable for reasoning about *discrete* spaces. Besides these two basic notions, a further logical operator has been introduced, namely the *surrounded* operator. This operator takes inspiration from the well-known temporal until-operator but is casted and re-interpreted in a discrete spatial setting. In summary, SLCS is equipped with two spatial operators: a "one step" modality, called "near" and denoted by $\mathcal{N}$, turning the closure operator into a logical operator, and a binary spatial until operator $\Phi_1 \, \mathcal{S} \, \Phi_2$. The basic idea is that a point x in the (quasi-discrete closure) space satisfies $\mathcal{N}\Phi$ if it is adjacent to a point that satisfies Φ. For instance, if we consider the model of Fig. 1, the green and the blue nodes satisfy $\mathcal{N}\,green$. The dual operator of $\mathcal{N}$ is the interior $\mathcal{I} = !\mathcal{N}(!\Phi)$. The green nodes satisfy $\mathcal{I}(green \cup blue)$.

A point x satisfies $\Phi_1 \, \mathcal{S} \, \Phi_2$ whenever there is "no way out" from a set of points, including x, and that each satisfy Φ_1 unless passing by a point that satisfies Φ_2. For instance, in Fig. 1, *yellow* nodes satisfy $yellow \, \mathcal{S} \, red$ while *green* nodes satisfy $green \, \mathcal{S} \, blue$.

This small set of spatial logic operators, together with the basic boolean operators such as negation and conjunction, is surprisingly expressive. For example, a number of interesting derived operators can be defined, including the well-known spatial "somewhere" and "everywhere" operators, and various forms of reachability. Moreover, an efficient model-checking algorithm has been developed for this set of operators that was first presented in [7]. In [9] the logic is extended with an additional "propagation" operator, $\mathcal{P}$, such that a point x satisfies $\Phi_1 \, \mathcal{P} \, \Phi_2$ if and only if it satisfies Φ_2 and there is a path rooted in a point satisfying Φ_1 where all other points satisfy Φ_2. This operator can be useful for describing, for instance, situations in which, a "safe" point x can be reached starting from a point where something dangerous takes place (e.g. Φ_1 could model the fact that there is a source of radiation there while Φ_2 represents shielded, safe, points in space). In the above mentioned paper, the logic has also been extended with *collective* operators, which are interpreted on *sets* of points instead of individual points. Finally the model-checking algorithms have been extended accordingly. We refer to [9] for details; a tutorial on the subject is provided in [10].

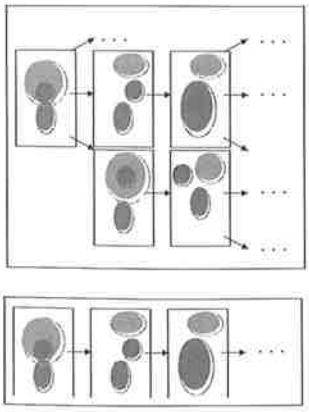

Figure 2: A temporal structure representing a computation tree of *snapshots* induced by the time-dependent valuations of the atomic propositions (top). A path in the model (bottom).

Figure 3: Formation of a wave-like pattern; evolution in steps of 10; pink points are part of a wave for 2 subsequent steps; cyan points for 10 subsequent steps. Other colours represent the intermediate number of steps.

3 Spatio-temporal Logic for Closure Spaces

The spatial logic SLCS has been extended with classical branching time temporal logic operators [11], leading to STLCS, in such a way that spatial and temporal operators can be arbitrarily nested. This way interesting properties of the spatio-temporal dynamics of a system can be expressed. In STLCS a temporal structure represents a computation tree of spatial *snapshots* induced by the time-dependent valuations of the atomic propositions (see Fig. 2). A spatio-temporal model checker was developed for STLCS and introduced in [11], called `topochecker`. This model checker[2] can verify multiple properties simultaneously and show their results in different selected colours. In case the results involve the same points, the later results overwrite the previous ones. The time complexity of the spatial model checking algorithm is linear in the number of points and arcs in the space and in the size of the formula.

To illustrate spatio-temporal model checking, we consider an example of the formation of a wave-like pattern as described in [10]. Such patterns can emerge when two particular chemical substances, or morphogens, A and B interact and diffuse over a surface in a way similar to that of the formation of Turing patterns, which were first studied and discovered by Alan Turing in his groundbreaking paper on morphogenesis in 1952 [12].

The spatio-temporal model consists of a sequence of snapshots of the first 100 time steps. This model is obtained as the numerical solution of the set of reaction-diffusion equations that describe the dynamics of the concentrations of both substances in a regular grid of discrete patches (see also [10]). The last snapshot of the sequence is repeated in an artificial way to obtain an infinite path. Each snapshot consists of a regular graph of 31 by 31 discrete patches, where each node is connected to its four direct neighbours. Each patch represents the local concentration of the two substances.

The spatio-temporal logic can be used to identify which points (denoting patches) are part of a pattern for a number of consecutive steps in the dynamic evolution of the space. First we define the spatial property 'pattern' as an area of points with low concentration a of chemical substance A surrounded by points with higher concentration of A:

$$\texttt{pattern} \;=[\texttt{a<0}]\texttt{S}[\texttt{a>=0}]$$

Then we define the various periods, ranging from 3 to 10 time steps, during which a point remains part of the pattern as follows (using the front end notation of `topochecker` for STLCS formulas):

$$\begin{aligned}
&\texttt{pattern2steps} && = \texttt{pattern\&AX(AXpattern)}\\
&\texttt{pattern3steps} && = \texttt{pattern2steps\&AX(AX(AXpattern))}\\
&\cdots\\
&\texttt{pattern10steps} && = \cdots
\end{aligned}$$

Here operator `A` denotes 'for all paths' and `X` is the next step operator from temporal logics. So the formula `pattern2steps` is satisfied by points that satisfy property 'pattern' now, and for all paths (there is only one in this single linear sequence) in the next snapshot the point satisfies 'pattern' in the next snapshot (on all paths). We can verify such properties starting from the initial snapshot, but also starting from any other chosen snapshot in the sequence.

Figure 3 shows the evolution of the wave-like pattern when the formulas are evaluated taking as initial snapshot the one at time 10, 20, 30 and 40, respectively. The results show that the pattern seems to stabilise starting from the north-western corner of the figure after which the points towards the south-eastern corner become increasingly stable, at least for 10 subsequent steps in time.

The spatial logic can be applied on any finite graph structure. The results for a 3D version of the wave-like pattern is shown in Fig. 4. The colours in that figure have the only purpose of being able to distinguish the pattern in a 3D representation. All coloured points satisfy the property "pattern" introduced before.

Another example shows how STLCS can be used to detect the formation of clusters of full bike stations in a simulation of a model of a bike sharing system [13,14]. The bike sharing model has a number of stations comparable to that of a city like London, but for simplicity they are assumed to be placed on a regular graph of 19 by 38 nodes (see Fig. 5). Full stations and clusters of full stations can be defined as:

$$\begin{aligned}
&\texttt{full} && = [\texttt{vacant==0}]\\
&\texttt{cluster} && = \texttt{I(full)}
\end{aligned}$$

Here [`vacant==0`] is an atomic proposition and `I` denotes the *interior* of a set of points (nodes). A point denoting a station evolves into a cluster when it becomes full, and stays full until it becomes part of a cluster. This may be detected

---

[2] Available at http://www.github.com/vincenzoml/topochecker.

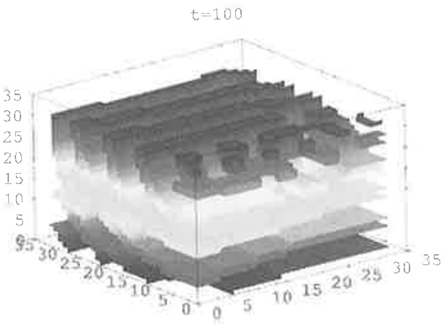

Figure 4: Analysis of property "pattern" in a 3D structure.

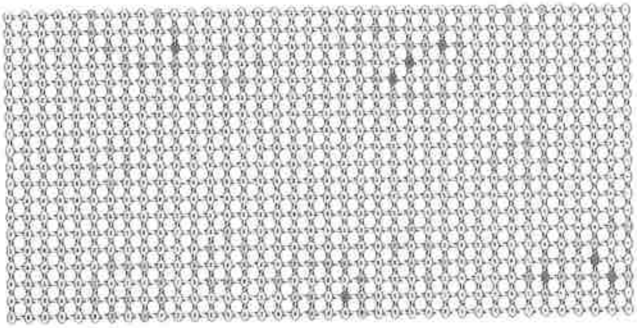

Figure 5: Formation of clusters (red) and boundary of points that will become a cluster (green).

by using the following formulas and using different colours to visualise the model-checking results as shown in Fig. 5:

```
implies(f,g)   = (!f)|g;
nextCluster    = (EF full)&
                    (AG implies(full,
                      A full U cluster))
```

The definition of `nextCluster` characterises points that will eventually become full and, for every future state, whenever full, they will remain full until they become part of a cluster. Such points are central in the formation of clusters, as they represent stations that always form a cluster when they become full. In Fig. 5, these points are shown in red, in a state of the simulation where there are many of them. For comparison, the boundary of the points that will become a cluster are shown in green, that is, those points satisfying `(NEF cluster)&(!EF cluster)`.

These are only a few examples that illustrate the use and potential of spatial and spatio-temporal model-checking in the context of CAS. Further details and examples can be found in a recent tutorial by the authors on spatial logic and spatial model-checking for closure spaces [10] and the references therein.

4 Conclusions and Outlook

We have provided a brief overview of recent work on the development of spatial and spatio-temporal model-checking for the analysis of Collective Adaptive Systems in the context of the EU FET-Proactive project QUANTICOL. The operators of the spatial logic SLCS have been inspired by topological operators and by a spatial version of the until-operator of temporal logic.

A prototype proof-of-concept spatio-temporal model-checker `topochecker` has been developed and used for the analysis of the dynamic spatio-temporal behaviour of various collective adaptive systems. Operators of the spatial logic have also been combined with a temporal logic for signals in [15] and provided with a quantitative semantics to assess the robustness with which formulas are satisfied.

Other recent work extends the approach to spatio-temporal *statistical* model-checking based on a statistical analysis of sets of simulations [16]. This approach provides insight in the *probability* with which a spatio-temporal property holds. For example, one can assess the probability that stations in a bike sharing system will get full. The approach exploits the use of a *single* set of simulations for the spatio-temporal properties of *all* points in a quasi-discrete closure space by means of the MultiVeStA [17] tool combining `topochecker` and the simulator for bike sharing models [13].

Future work is planned on the extension of the approach with suitable metric spaces and further operators, in particular for the application of the approach in the domain of medical imaging. Preliminary work on these ideas can be found in [18]. Furthermore, a possible integration of spatial model checking with highly scalable mean-field based model checking is envisioned and the exploration of suitable spatial model reduction methods. We also plan to investigate the development of a spatio-temporal model checker as a suitable combination of `topochecker` and an ADA implemented temporal model checker of the KandISTI family [19,20].

Acknowledgements This work is supported by the EU project *QUANTICOL: A Quantitative Approach to Management and Design of Collective and Adaptive Behaviours*, 600708. The authors would like to thank Luca Bortolussi, Stephen Gilmore, Gianluca Grilletti, Laura Nenzi, Rytis Paškauskas and Andrea Vandin who are involved in the QUANTICOL project and who are co-authors of the various articles involving spatial and spatio-temporal model-checking closely related to the work mentioned in the present position paper.

References

[1] C. Baier and J.-P. Katoen (2008), *Principles of model checking*, MIT Press.

[2] C. Baier, B. R. Haverkort, H. Hermanns, and J. Katoen (2003), *Model-checking algorithms for continuous-time markov chains*, IEEE Trans. Software Eng., vol. 29, no. 6, pp. 524–541.

[3] M. Z. Kwiatkowska, G. Norman, and D. Parker (2011), *PRISM 4.0: Verification of probabilistic real-time systems*, Computer Aided Verification - 23rd International Conference Proceedings, (G. Gopalakrishnan and S. Qadeer, eds.), vol. 6806 of LNCS, pp. 585–591, Springer.

[4] L. Bortolussi and J. Hillston (2015), *Model checking single agent behaviours by fluid approximation*, Inf. Comput., vol. 242, pp. 183–226.

[5] D. Latella, M. Loreti, and M. Massink (2015), *On-the-fly PCTL fast mean-field approximated model-checking for self-organising coordination*, Sci. Comput. Program., vol. 110, pp. 23–50.

[6] J. van Benthem and G. Bezhanishvili (2007), *Modal logics of space*, Handbook of Spatial Logics, pp. 217–298, Springer.

[7] V. Ciancia, D. Latella, M. Loreti, and M. Massink (2014), *Specifying and Verifying Properties of Space*, Theoretical Computer Science - 8th IFIP International Conference, J. Díaz, I. Lanese, and D. Sangiorgi, eds.), vol. 8705 of LNCS, pp. 222–235, Springer.

[8] A. Galton (2003), *A generalized topological view of motion in discrete space*, Theoretical Computer Science, vol. 305, no. 1–3, pp. 111 – 134.

[9] V. Ciancia, D. Latella, M. Loreti, and M. Massink (2016), *Model Checking Spatial Logics for Closure Spaces*, Logical Methods in Computer Science, vol. 12, Issue 4.

[10] V. Ciancia, D. Latella, M. Loreti, and M. Massink (2016), *Spatial logic and spatial model checking for closure spaces*, 16th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Quantitative Evaluation of Collective Adaptive System, (M. Bernardo et al., ed.), vol. 9700 of LNCS, pp. 156–201, Springer.

[11] V. Ciancia, G. Grilletti, D. Latella, M. Loreti, and M. Massink (2015), *An experimental spatio-temporal model checker*, Software Engineering and Formal Methods - Collocated Workshops, Revised Selected Papers, vol. 9509 of LNCS, pp. 297–311, Springer.

[12] A. M. Turing (1952), *The Chemical Basis of Morphogenesis*, Philosophical Transactions of the Royal Society of London B: Biological Sciences.

[13] M. Massink and R. Paškauskas (2015), *Model-based assessment of aspects of user-satisfaction in bicycle sharing systems*, Proceedings of the 18th IEEE International Conference on Intelligent Transportation Systems, (Sotelo Vazquez, M. et al., ed.), pp. 1363 –1370, IEEE.

[14] V. Ciancia, D. Latella, M. Massink, and R. Pakauskas (2015),*Exploring spatio-temporal properties of bike-sharing systems*, IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, pp. 74–79, IEEE Computer Society.

[15] L. Nenzi, L. Bortolussi, V. Ciancia, M. Loreti, and M. Massink (2015), *Qualitative and quantitative monitoring of spatio-temporal properties*, RV, vol. 9333 of LNCS, pp. 21–37, Springer.

[16] V. Ciancia, D. Latella, M. Massink, R. Paškauskas, and A. Vandin (2016), *A tool-chain for statistical spatio-temporal model checking of bike sharing systems*, Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, Proceedings, Part I (T. Margaria and B. Steffen, eds.), vol. 9952 of LNCS, pp. 657–673, Springer.

[17] S. Sebastio and A. Vandin (2013), *MultiVeStA: Statistical Model Checking for Discrete Event Simulators*, ValueTools, pp. 310–315, ACM.

[18] G. Belmonte, V. Ciancia, D. Latella, and M. Massink (2016), *From collective adaptive systems to human centric computation and back: Spatial model checking for medical imaging*, Proceedings of the Workshop on FORmal methods for the quantitative Evaluation of Collective Adaptive SysTems, (M.-H. ter Beek and M. Loreti, eds.), vol. 217 of EPTCS, pp. 81–92.

[19] S. Gnesi and F. Mazzanti (2011), *An abstract, on the fly framework for the verification of service-oriented systems*, Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA Project on Software Engineering for Service-Oriented Computing, (M. Wirsing and M. M. Hölzl, eds.), vol. 6582 of LNCS, pp. 390–407, Springer, 2011.

[20] M. H. ter Beek, S. Gnesi, and F. Mazzanti (2015), *From EU projects to a family of model checkers - from kandinsky to kandisti*, Software, Services, and Systems - Essays Dedicated to Martin Wirsing on the Occasion of His Retirement from the Chair of Programming and Software Engineering, (R. D. Nicola and R. Hennicker, eds.), vol. 8950 of LNCS, pp. 312–328, Springer, 2015.