ACI Sécurité 2004 : FIACRE

-

Fiabilité des Assemblages de Composants Répartis

-

Models and Tools for Safety and Security Analysis of Distributed Components and their Composition

mid-term report

june 2006

1 Keywords

Distributed applications, Composition of components, Model extraction, Temporal logic, Model-checking, Behavioural specification, Behavioral typing, Model-based verification, Compositional verification

2 Summary

In forthcoming years, distributed component based programming will have a strong impact on software development methods. In order for this approach to fully work, while component libraries become available, it is necessary to be able to compose existing components into more complex objects, and to guarantee that this composition will work correctly and fulfill its expected role. Classical, static interface typing does not allow to reach this goal.

Gathering teams specialized in behavioural specifications of components, languages and models for distributed, mobile, and communicating application programming, and methods and tools for compositional verification, the goal of FIACRE is to design methods and tools for specification, model extraction, and verification of distributed, hierarchical, and communicating components. Our proposal is articulated around the following axes:

- Definition of a specification formalism for component behaviours, which must be adapted to verify distributed applications and allow an easy translation into the low-level formalisms that are used for verification.
- Development of semi-automated procedures for the behavioural model extraction of distributed components.
- Efficient tools for the verification (either using temporal logic formulas, behavioural equivalences, or behavioural typing) of the hierarchical compositions of components from their behavioural specifications.

We would like the collaboration to result in a software prototype applicable to realistic applications.

3 Partners and Participants

Partner	Funded by partner	funded by Fiacre
Oasis	Eric Madelaine (CR1) Rabéa Boulifa (PhD, ending oct 2004) Tomás Barros (PhD, ending nov 2005) Antonio Cansado (PhD, starting oct 2005) Marcella Rivera (internship 2005/2006)	Walid Belkhir (master 2005) Hejer Rejeb (IE, starting 2/2006)
Vasy	Hubert Garavel (DR2) Frederic Lang (CR1) Radu Mateescu (CR1) Gwen Salaun (PostDoct)	IE, starting 9/2006
Feria	François Vernadat (Pr) Mamoun Filali (CR1) Bernard Berthomieu (CR1) Jean-Paul Bodeveix (Pr) Tarek Sadani (PhD)	Rodrigo Tacla Saad (internship 9/05-¿2/06)
ENST	Elie Najm (Pr) Sylvie Vignes (Pr) Irfan Hamid (PhD)	

4 Results of first period

4.1 T1: Specification Language and Intermediate Model

In this task we commit to develop in a coherent way both a behavioural specification language for the component developer, and an intermediate model that can be used as intermediate representations for connecting our tools. The specification language covers the behaviour of primitive components and the architecture of composite components, and should include high level notions able to express the specific concepts of distributed component systems.

The common basis for the specification language and the intermediate representations is the *pNets model* that represents the behaviour and interactions between components as parameterized synchronization networks of parameterized labelled transition systems (LTSs). This model has been defined in [3]. It is powerful enough to be an easy target for automatic model generation from static analysis of source code, but it is also a low level semantic representation that is easy to translate to the verification tool input languages.

At the intermediate format level we use FC2Parameterized [2] as the concrete syntax for our formalism. This language is powerful enough to represent all features of the model including parameters, hierarchy and dynamism. It is suitable for interfacing with the various verification tools used or developed by the Fiacre partners. As in the short term we target classical checkers, manipulating finite structures, the FC2Parameterized format also encodes the definition of domain instantiations; the instantiated models are represented in the non-parameterized subset of FC2. We have shown that this format can be easily translated into the input formats of the CADP toolset (bcg and exp). We are also working on mappings towards the input languages of other tools, including Cotre for Tina, and Altarica for the infinite state tools of the Persee ACI.

At the specification level the user can define parameterized LTSs in either FC2 or LOTOS syntaxes. We have proposed a graphical language covering a static subset of this formalism [1]. It was successful in the specification of a complex case study involving hierarchical parameterized distributed processes. We use the Fractal component model for specifying the architecture of our component systems. We are proposing an extension to the Fractal architecture description language to handle behaviour specifications in our format [12]. In particular, this extension will be used for distributed components in the ProActive implementation of Fractal, including specific high level primitives for group communications. Our extended distributed Fractal model will be the basis for the Grid Common Model (GCM) of the CoreGrid network of excellence. Several tracks of research are opened for further development: A specification language that would incorporate both the architectural aspects (logical topology of the distributed component assemblies) and the behavioural aspects is still missing. Also missing is a "programmer-level formalism", that would be easily usable by application developpers, usually not specialists in formal methods.

Studying the relations with existing work on behaviour typing, we have adapted the concept of behaviour contracts previously developed by the ENST team to the framework of ProActive distributed components (encoding of ProActive asynchronous request and management of future values). In contrast with previous works, our behavioural types specify whole components rather than the protocol at a given interface [9]. This will allow us to define a notion of sound hierarchical composition.

4.2 T2: Automatic Model Generation

Based on the generic model [3] defined in task 1, we have defined methods for constructing behaviour models of components by static analysis of their source code, and we have released recently a first version of analysis tools.

These methods and tools apply to the ProActive implementation of Fractal, in which the code is a combination of Java/ProActive code for primitive components, and Fractal ADL for the description of the architecture of composite components and ultimately of the whole application.

Parameterized LTS are used to describe the external behaviour of primitive components (interaction at the interfaces, protocol at the interfaces). Synchronization networks are used to describe the interactions between components at each level of the hierarchy. The model is general enough to tackle synchronous communication as well as asynchronous request mechanisms used in distributed component systems [5, 6, 7]. Our parameters represent both value passing, data flow relations and indexed families of components.

This approach is not totally automatic, because it requires at least that users provide adequate abstractions of the data domains used in their code. Idealy, those abstractions should be proved correct in the sense of abstract interpretation theory, in order to preserve the properties we prove; this may require using (user-guided) theorem provers. Furthermore, we require the user of the analysis tools to provide a proper instantiation of the parameters domains.

Our current analysis platform [4] includes a first version of the ADL2N tool, that analyses the ADL files, and generates the synchronisation networks for composite components. It takes into account only the functional behaviour of the components; the next version will produce also controllers encoding the component management features of Fractal, and the asynchronous communication of ProActive.

The platform also includes tools for instantiating the parameterized systems, and for producing intermediate files in the input syntax of the CADP toolset. This has been described in [2].

We have recently started working on the tool that will perform static analysis of the ProActive code and build the corresponding pNets.

4.3 T3: Verification

This task relies on the existing toolsets CADP (http://www.inrialpes.fr/vasy/cadp) and TINA (http://www.laas.fr/tina) developed by VASY and Feria respectively. Part of this task has consisted of exchanges between the users and the developers belonging to Fiacre. These exhanges have allowed to transfer expertise in the use of advanced verification tools, such as the distributed state space generation tool Distributor of CADP, that also includes a simple variant of partial order reduction. The use of Distributor has allowed to generate state spaces corresponding to large compositions of components, which could not have been generated using sequential tools.

Beyond this collaboration, most of the efforts have led to the integration in each toolset of the ideas and techniques developed by the other partners.

These exchanges have led to several enhancements in some CADP tools, Bisimulator [10, 13], and Distributor [14]. A new tool named Reductor 4.0 has been developed to permit on-the-fly reduction of state spaces. Exp.Open 2.0 [15] supports several new means of synchronisations, including the synchronisation vectors of the *pNets model* from task 1.

The TINA toolbox has also been enhanced along the following lines :

- A new model checker for State-Event LTL was developed [8]. This tool can be used to model check labelled transition systems represented either in the TINA format or in the BCG format of CADP.

- TINA was extended to handle TTS (Timed Transition Systems), which are a variant of "predicate transition nets" with time constraints. This gives TINA the capability of building abstract state spaces for systems with both data and time. A new input format in two parts was defined for TTS : the first part describes control as a Time Petri net and the second part describes data as a set of C functions and predicates.

- Two compilers from RT/LOTOS [16] (a variant of LOTOS with real-time capabilities) and V-Cotre (a component based language initially targeted to real-time avionic applications defined in the RNTL Cotre project and constituting the basis of the behavioural annex of AADL [11] proposed for standardization) have been developed. Both compilers generate a TTS corresponding to the input model.

Another part of the task has consisted of developing links to combine the toolboxes when this was meaningful. So far, translators have been developed between the formats used to represent Labeled Transition Systems in the respective toolboxes. In addition, the design of a common intermediate semantic model that will combine the best features of the V-Cotre format and the NTIF format has been undertaken.

5 Other Results and Dissemination

In terms of behavioural model of components, the Fiacre preliminary results have been presented both to the Fractal community (Fractal workshops, Grenoble, sep. 2005 and Nantes, jul. 2006), and to the CoreGrid network of excellence (GridCoord conf., Sophia-Antipolis, oct. 2005). In the first case, this has lead us to submit a proposal for an extension of the Fractal ADL definition, incorporating elements for the behaviour specification. In the second case, our results will be integrated in the "Grid Component Model" (GCM) defined by the CoreGrid WP3 partners.

The Fiacre ACI was used as a basis to launch perennial collaborations between FERIA and VASY. We can mention two important successes:

- The national RTNL platform project "OpenEmbedd", where FERIA and VASY are in charge of the crucial semantic aspects, especially to define the common semantic format of OpenEmbedd for asynchronous systems,

- The "Topcased" project belonging to "Pôle de Compétitivité" AESE ("Aéronautique, Espace et Systèmes Embarqués") where FERIA and VASY are applying the Fiacre results to develop an integrated verification chain for avionics applications written in the AADL standard extended with appropriate behavioural annotations. In particular, they are defining an intermediate language for verification called Fiacre ("Format Intermédiaire pour les Architectures de Composants Répartis Embarqués") based on the Fiacre developments.

Software

- Beta-version 2005-c of CADP was released in may 2006, including the improvements and the new tools mentionned in task 3, and in particular the new Exp input format that is used by the Vercors platform. http://www.inrialpes.fr/vasy/cadp
- Tina version 2.7.4 was released in nov 2005, and Tina 2.8.0 (beta) in may 2006., including support for the PNML format, and the SE-LTL checker. http://www.laas.fr/tina
- Tools composing the Vercors platform are available at http://www-sop.inria.fr/oasis/Vercors, including FC2Instantiate, FC2Exp, and ADL2N v0.8

References

[1] I. Attali, T. Barros, and E. Madelaine. Formalisation and proofs of the chilean electronic invoices system. In *in proc. of the XXIV International Conference of the Chilean Computer Science Society (SCCC'04)*, Arica, Chili, November 2004. IEEE.

[2] T. Barros. *Formal specification and verification of distributed component systems.* PhD thesis, Université de Nice - INRIA Sophia Antipolis, November 2005.

[3] T. Barros, R. Boulifa, and E. Madelaine. Parameterized models for distributed java objects. In *Forte'04 conference*, volume LNCS 3235, Madrid, September 2004. Spinger Verlag.

[4] T. Barros, A. Cansado, and E. Madelaine. Model-checking distributed components: The Vercors platform. In *submitted*, 2006.

[5] T. Barros, L. Henrio, and E. Madelaine. Behavioural models for hierarchical components. In Patrice Godefroid, editor, *Model Checking Software, 12th International SPIN Workshop*, volume LNCS 3639, pages 154–168, San Francisco, CA, USA, August 2005. Springer.

[6] T. Barros, L. Henrio, and E. Madelaine. Behavioural models for hierarchical components. Technical Report RR-5591, INRIA, June 2005.

[7] T. Barros, L. Henrio, and E. Madelaine. Verification of distributed hierarchical components. In *International Workshop on Formal Aspects of Component Software (FACS'05)*, Macao, October 2005. Electronic Notes in Theoretical Computer Science (ENTCS).

[8] B.Berthomieu and F. Vernadat. Réseaux de Petri temporels : méthodes de vérification et analyse avec Tina. In Nicolas Navet, editor, *Vérification de systèmes Temps Réel Tome I*, Traité IC2. Hermes, 2006.

[9] Walid Belhkir. Behavioural typing for proactice distributed components. Technical report, Mastère recherche Univ. d'Aix-Marseille, June 2005.

[10] D. Bergamini, N. Descoubes, C. Joubert, and R. Mateescu. Bisimulator: A modular tool for on-the-fly equivalence checking. In Nicolas Halbwachs and Lenore Zuck, editors, *Proc. of the 11th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems TACAS'2005.* LNCS 3440, Springer Verlag, apr. 2005.

[11] J.P. Bodeveix, P. Dissaux, P. Farail, M. Filali, P. Gaufillet, and F. Vernadat. Behavioural descriptions in architecture description languages: Application to AADL. In *Europ. Congress: ERTS2006: Embedded Real Time Software, Toulouse.* SIA, SEE, AAAF, jan 2006.

[12] A. Cansado, L. Henrio, and E. Madelaine. Towards real case component model-checking. In *5th Fractal Workshop*, Nantes, France, July 2006.

[13] C. Joubert and R. Mateescu. Distributed local resolution of boolean equation systems. In Francisco Tirado and Manuel Prieto, editors, *Proc. of the 13th Euromicro Conf. on Parallel, Distributed and Network-Based Processing PDP'2005*, Lugano, February 2005. IEEE Computer Society.

[14] C. Joubert and R. Mateescu. Distributed on-the-fly model checking and test case generation. In *Proc. of the 13th Int. Workshop on Model Checking of Software SPIN'2006 (Vienna, Austria)*, 2006.

[15] F. Lang. Exp.open 2.0: A flexible tool integrating partial order, compositional, and on-the-fly verification methods. In Jaco van de Pol, Judi Romijn, and Graeme Smith, editors, *Proc. of the 5th Int. Conf. on Integrated Formal Methods IFM'2005 (Eindhoven)*. LNCS 3771, Springer Verlag, nov. 2005.

[16] T. Sadani, P. De Saqui Sannes, and J.P. Courtiat. Validation de spécifications rt-lotos. une interface vers l'outil tina. In *"Modélisation des Systèmes Réactifs", MSR'05*, Grenoble, October 2005.

6 Annex: Agenda of Fiacre meetings

Plenary meetings

- **1st Fiacre meeting, INRIA Sophia Antipolis, November 22-23, 2004**

  E. Madelaine: “Modèles paramétrés pour les applications ProActive”

  F. Lang: ”Activités de recherche du projet Vasy, outils CADP”

  F. Lang: ”Compositional Verification Using Cadp of the ScalAgent Deployment Protocol for Software Components”

  M. Filali: “Vérification des systèmes paramétrés et infinis”

  B. Berthomieu: “Outil TINA”

  E. Najm: ”Behavioural Contracts for a Sound Assembly of Components”

  T. Barros: “Intermediate Format for Hierarchical Systems”

- **2nd Fiacre meeting, LAAS/CNRS, Toulouse, March 18-19, 2005**

  F. Vernadat: “Méthodes ordres partiels implantées dans TINA”

  B. Berthomieu: “Démo TINA, nouveaux développements (stepper, SE-LTL model-checker)”

  F. Lang: ”Démo CADP (génération distribuée d’espaces d’états, Exp.Open 2.0)”

  T. Barros: “Behavioural Models for Hierarchical Components”

  G. Salaun: ”Describing and Reasoning on Web Services using Process Algebra”

  G. Salaun: ”Formal Coordination of Distributed Entities described with Behavioural Interfaces”

  T. Barros: “Démo ProActive”

  I. Hamid: “Sémantique de AADL”

  E. Najm: “Assemblage Sain pour Composants Distribués”

  E. Najm: “Component Connectors”

- **3rd Fiacre meeting, INRIA Rhone-Alpes, Montbonnot, September 26-27, 2005**

  E. Madelaine: “L’outil ADL2N: conception”

  T. Barros: “Behavioural Models for Distributed Hierarchical Components”

  T. Barros “Démos ADL2N, FC2Instantiate, FC2Exp, CADP

  I. Hamid: “Composants et Connecteurs: vers une construction formelle d’un middleware temps-réel”

  F. Vernadat: “L’outil TINA: construction d’espaces d’états abstraits pour la vérification de systemes critiques; derniers dévelopements”

  M. Filali: “Raffinement contextuel de composants paramétrés”

  F. Lang. ”Exp.Open 2.0: un outil flexible intégrant réductions d’ordres partiels, vérification compositionnelle et vérification à la volée”

  H. Garavel. “DISTRIBUTOR and BCG_MERGE: Tools for Distributed Explicit State Space Generation”

  Invited Speaker: J.-B. Stefani (Sardes project, INRIA Rhone-Alpes). ”Une introduction à Fraktal”

- **4th Fiacre meeting, ENST Paris, February 13-14, 2006**

  J. Hugues: “Modélisation Comportementales d’intergiciels (temps réel réparti embarqué)

  F. Vernadat: “Derniers dévelopements de l’outil TINA”

  Invited Speaker: Pascal Poizat (LAMY, Evry): “Adaptation Logicielle”

  F. Lang: “Propositions d’extensions temporisées pour NTIF”

  A. Cansado: “[progress on] Verification of Distributed Components”

  groupe de travail “Génération de mod‘eles Lotos” (Oasis, Vasy)

  groupe de travail “Modèle intermédiaire” (Vasy, Feria)

Technical meetings

- Oasis - ENST, Paris, june 2005: “Behavioural typing of components”
- Oasis - ENST, Sophia-Antipolis, july 2005: “Behavioural Typing for ProActive distributed objects”
- Vasy - Feria, Toulouse, 20-21 mars 2006: “Langage intermédiaire pour les outils de vérification”
- Vasy - Feria, Toulouse, 20 juin 2006: “Langage intermédiaire Fiacre”