Bigraphical Programming Languages for Pervasive Computing

L. Birkedal[1], M. Bundgaard[1], T. C. Damgaard[1], S. Debois[1], E. Elsborg[1],
A. J. Glenstrup[1], T. Hildebrandt[1], R. Milner[2], and H. Niss[1] ⋆

[1] The BPL project, IT University of Copenhagen, Denmark
`bpl@itu.dk`
[2] Computer Laboratory, University of Cambridge, UK
`Robin.Milner@cl.cam.ac.uk`

Abstract. The Bigraphical Programming Language project at IT University of Copenhagen contributes to the UKCRC Ubiquitous Computing Grand Challenge by researching the use of bigraphical reactive systems as a general framework in which to combine theories for design and analysis with techniques, tools and methodologies for engineering and systems building. Initial work has been addressing *Context-awareness, business processes and Reactive XML, axiomatisation and matching,* and *higher-order mobile embedded resources.*

The UKCRC Ubiquitous Computing Grand Challenge (short, UCGC) was formed by merging two proposed grand challenges, one focused on the theory and one focused on engineering and systems building. Recently it has been broadened to include goals for the human experience of ubiquitous computing.

The aim of the Bigraphical Programming Language (BPL) project at IT University of Copenhagen is to contribute to the UCGC by researching the use of Bigraphical Reactive Systems [13, 22, 23, 24] (BRS) as a foundation for the combination of theory and systems building for context-dependent mobile communicating systems.

BRS is a general graphical semantical framework for reactive systems unifying the feature of dynamic communication links introduced with the π-calculus [25] and the feature of mobile nested spatial structures introduced with the Mobile Ambients calculus [8]. Loosely speaking, a bigraphical reactive system consists of set of bigraphs and a set of reaction rules, which can be used to reconfigure the set of bigraphs. BRSs have been developed with principally two aims in mind: (1) to be able to model directly important aspects of ubiquitous systems by focusing on mobile connectivity and mobile locality, and (2) to provide a unification of existing theories by developing a general theory, in which many existing calculi for concurrency and mobility may be represented, with a uniform behavioural theory.

The activities within the BPL project are currently focussed on the following areas: *Context-awareness*, *business processes*, *implementation of bigraphs as Reactive XML*, *axiomatisation and matching*, and *higher-order mobile embedded resources*. The areas are described briefly below.

1 Context-Awareness

A central aspect of pervasive computing is the movement of software and hardware agents in environments equipped with sensors that may also move. Many such mobile systems with particular purposes have been realised in practical experiment, or simulated. One example of a purpose is "sentient computing , defined by Hopper as "Using sensor and resource status data to maintain a model of the world which is shared between users and applications" [21]. Other purposes are possible; for example, sensors (fixed or mobile) may be programmed to collaborate in guiding an agent to a goal. The variety of applications is large, and suggests that a first step is to study and analyse locality and movement per se, without prejudice towards a particular application.

The UCGC foothill project *Analysing Movement in a Sentient Environment* [9] addresses the following problem: What is a fruitful conceptual framework in which to express a variety of rules of motion, allowing systems to be programmed conveniently, simulated (with the help of stochastic information in the rules), and analysed rigorously? An example of rigorous analysis would be that certain invariants are maintained (or maintained with certain probability) by all behaviour allowed by the rule.

To underpin both description and analysis, it is convenient to use a spatial model which can represent both discrete and continuous space, and movement in such space, as well as the usual forms of data and processes involved in traditional computing. Examples of such models are suggested by calculi of interactive computing, such as the calculus of Mobile Ambients or the π-calculus. But the aim of the project is not just to design a calculus; it is to derive from the calculus a programming language, and define a programming methodology, so that the language may be used and evaluated by people whose primary interest is in applications.

The ultimate goal of the project is to unify theory and practice in this basic but challenging facet of pervasive computing. Initial work within the BPL project on providing bigraphical models for context-awareness is presented in [4], in which we propose to model (and ultimately program) a "reflective" building: one equipped with sensors, which continually transmit data of the building's occupancy to a monitor that maintains a data structure which faithfully records the occupancy. The work has led to a more general study of *sortings*, which tailor the description of context-aware behaviour to particular systems while retaining the theory of bisimulation congruences [5].

2 Business Processes and Reactive XML

The vision for the future ubiquitous IT-support at our work places is that we can carry a mobile device anywhere allowing us seamlessly to cooperate, share information, and coordinate tasks with our colleagues. A topical example is the electronic medical journals currently being developed world-wide as part of healthcare and hospital information systems.

Ideas from *Workflow management systems* (WFMS) incorporated in most ERP-systems and *service composition* within service oriented architectures—fueled by business process and service composition languages and notations such as BPEL4WS [12] and the Business Process Modelling Notation (BPMN) [27]—are likely to play a central role for this vision to become true.

In [19, 26] we describe a general approach to define an XML-based exchange and execution format for business process and coordination languages, as well as an XML-based format for the process language semantics based on bigraphical reactive systems. The approach exploits similarities of Bigraphs and XML. We used a subset of WS-BPEL as a case illustrating the idea. The case study suggested two interesting extensions of BRS: The first extension is to allow for (linear) higher-order reaction rules. The second extension is to constrain composition of contexts by tree logics, in this concrete case a subset of XPath, resulting in a kind of *context-dependent* reaction rules. Using respectively a Geometry of Interaction [1, 14, 15] construction and general construction of sortings on bigraphs, we are currently working on describing these extensions as general constructions on the underlying category of bigraphs and show that the general theory of bigraphs, notably the derivation of bisimulation congruences, can be safely extended.

The bigraphical process execution engine has been implemented in a prototype called Distributed Reactive XML [17, 20, 28], which provides an extensible, distributed (peer-to-peer) process execution engine directly based on the formalisation. The implementation of Distributed Reactive XML so far serves as a proof of concept. However, by representing the business process descriptions, their state and semantics of the process languages as XML and implementing it on top of a distributed peer-to-peer XML storage layer allowing concurrent reactions on shared processes and data, we achieve many of the features of the ideal scenario for a coordination middleware as described in [10].

3 Matching and Axiomatisation

The dynamics of bigraphical reactive systems is represented by an abstract definition of reaction rules. As suggested and argued in [2, 4, 22] it would be very useful to have an implementation of the dynamics of bigraphical reactive systems to allow experimentation and simulation. In the BPL project we are working towards such an implementation, which also forms an important part of Distributed Reactive XML described above. The core problem of implementing the dynamics of bigraphical reactive systems is the *matching problem*, that is, to

determine for a given bigraph and reaction rule whether and how the reaction rule can be applied to rewrite the bigraph. The matching problem is analyzed in [3], which provides a sound and complete inductive characterization of matching. This provides a detailed analysis of the matching problem, and paves the way for developing and proving correct an actual matching algorithm on which we are currently working. The inductive characterization is based on normal form theorems for bigraphs [11, 24], which express how general bigraphs may be decomposed into a composition of simpler graphs. Based on axioms for determining whether two bigraphical expressions denote the same bigraph, matching algorithms may operate on normal form representations, and may be founded on principles of equational reasoning.

Of course, the matching problem is closely related to the NP-complete graph embedding problem. Thus we analyze the embedding problem for a restricted class of graphs, and our inductive characterization makes good use of the algebraic presentation of such graphs [11, 24]. One hopes that matching implementations will be efficient in practice since redexes, that is, the precondition patterns of reactions, typically are small.

4 Higher-order Mobile Embedded Resources

As described above Bigraphs have been introduced with the aim to provide a topographical meta-model for mobile, distributed agents that can manipulate their own linkages and nested embedded locations, generalising both characteristics of the π-calculus and the Mobile Ambients calculus. One scenario for pervasive and ubiquitous computing describes active software agents move between service providers, e.g. travel agencies, collecting information about services before placing an order. A serious threat to this kind of agent based service is the ability to copy and execute several copies of mobile software agents. Neither the π-calculus nor the Mobile Ambients calculus supports modeling of such mobile processes that can be copied during their execution, also referred to as higher-order (or non-linear) active process mobility.

In [6] we give a bigraphical semantics for the calculus of *Higher-Order Mobile Embedded Resources (Homer)*[7, 16, 18], which is a higher-order process calculus with nested locations, non-linear active process mobility, and local names. The semantics is based on Milner's presentation of the λ-calculus in local bigraphs. Interestingly, the combination of non-linear active process mobility and local names requires a new definition of parametric reaction rules and a representation of the *location* of names. Current work addresses providing types for non-copyable and copyable mobile embedded resources.

References

[1] Samson Abramsky. Retracing some paths in process algebra. In *Proceedings of CONCUR '96—International Conference on Concurrency Theory*, volume 1119 of *LNCS*, pages 1–17, 1996.

[2] Lars Birkedal. Bigraphical Programming Languages—a LaCoMoCo research project. In *Second UK UbiNet Workshop, Cambridge*, May 2004. Position Paper.

[3] Lars Birkedal, Troels Christoffer Damgaard, Arne John Glenstrup, and Robin Milner. Matching of bigraphs. Submitted for publication, 2006.

[4] Lars Birkedal, Søren Debois, Ebbe Elsborg, Thomas Hildebrandt, and Henning Niss. Bigraphical models of context-aware systems. In *Proceedings of the 9th International Conference of Foundations of Software Science and Computation Structures (FOSSACS'06)*, volume 3921 of *Lecture Notes in Computer Science*, pages 187–201. Springer-Verlag, 2006.

[5] Lars Birkedal, Søren Debois, and Thomas Hildebrandt. Sortings for reactive systems. Submitted for publication, 2006.

[6] Mikkel Bundgaard and Thomas Hildebrandt. Bigraphical semantics of higher-order mobile embedded resources with local names. In *Proceedings of the Graph Transformation for Verification and Concurrency workshop (GT-VC'05)*, Electronic Notes in Theoretical Computer Science. Elsevier, 2005. To appear.

[7] Mikkel Bundgaard, Thomas Hildebrandt, and Jens Chr. Godskesen. A CPS encoding of name-passing in higher-order mobile embedded resources. *Electronic Notes in Theoretical Computer Science*, 128(2):131–150, 2005.

[8] Luca Cardelli and Andrew D. Gordon. Mobile ambients. In *Proceedings of FoSSaCS'98*, volume 1378, pages 140–155. Springer-Verlag, 1998.

[9] Dan Chalmers, Matthew Chalmers, Jon Crowcroft, Marta Kwiatkowska, Robin Milner, Eamonn O'Neill, Tom Rodden, Vladimiro Sassone, and Morris Sloman. Ubiquitous computing: Experience, design and science. Ubiquitous Computing Grand Challenge Manifesto, February 2006. ⟨`http://www-dse.doc.ic.ac.uk/Projects/UbiNet/GC/Manifesto/manifesto.pdf`⟩.

[10] P. Ciancarini, R. Tolksdorf, and F. Zambonelli. Coordination middleware for XML-centric applications. In *Proc. ACM/SIGAPP Symp. on Applied Computing (SAC)*. ACM Press, 2002.

[11] Troels C. Damgaard and Lars Birkedal. Axiomatizing binding bigraphs (revised). Technical Report TR-2005-71, IT University of Copenhagen, 2005.

[12] Tony Andrews et al. Business process execution language for web services (BPEL4WS). ⟨`http://www-128.ibm.com/developerworks/library/specification/ws-bpel/`⟩, 2003.

[13] Ole Høgh Jensen and Robin Milner. Bigraphs and transitions. In *Proceedings of the 30th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 38–49. ACM Press, 2003.

[14] J.Y. Girard. Geometry of interaction I: interpretation of system F. In R. Ferro, C. Bonotto, S. Valentini, and A. Zanardo, editors, *Proceedings Logic Colloquium '88*, volume 127 of *Studies in Logic and the Foundations of Mathematics*, pages 221–260. North-Holland, 1989.

[15] J.Y. Girard. Geometry of interaction II: deadlock free algorithms. In *Proceedings of COLOG 88*, number 417 in Lecture Notes in Computer Science, pages 76–93. Springer-Verlag, 1989.

[16] Jens Chr. Godskesen and Thomas Hildebrandt. Extending Howe's method to early bisimulations for typed mobile embedded resources with local names. In *Proceedings of the 25th Conference on the Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05)*, volume 3821 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[17] Thomas Hildebrandt and Jacob W. Winther. Bigraphs and (Reactive) XML. Technical Report TR-2005-56, IT University of Copenhagen, 2005.

[18] Thomas Hildebrandt, Jens Chr. Godskesen, and Mikkel Bundgaard. Bisimulation congruences for Homer — a calculus of higher order mobile embedded resources. Technical Report TR-2004-52, IT University of Copenhagen, 2004.

[19] Thomas Hildebrandt, Henning Niss, and Martin Olsen. Formalising business process execution with bigraphs and reactive xml. In *Proceedings of Coordination '06*, 2006. To appear.

[20] Thomas Hildebrandt, Henning Niss, Martin Olsen, and Jacob W. Winter. Distributed reactive XML. In Lubos Brim and Isabelle Linden, editors, *Proceedings of the First International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord'05)*, volume 150 of *Electronic Notes in Theoretical Computer Science*, pages 61–80. Elsevier, 2006.

[21] Andy Hopper. Sentient computing. *Philosophical Transactions Royal Society London*, 358(1773):2349–2358, 2000. In the Clifford Paterson Lecture.

[22] Ole Høgh Jensen and Robin Milner. Bigraphs and mobile processes (revised). Technical Report UCAM-CL-TR-580, University of Cambridge, Computer Laboratory, February 2004.

[23] Robin Milner. Bigraphical reactive systems. In *Proceedings of CONCUR '01—12th International Conference on Concurrency Theory*, pages 16–35, 2001.

[24] Robin Milner. Axioms for bigraphical structure. *Journal of Mathematical Structures in Computer Science*, 15(6):1005–1032, 2005.

[25] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, I. *Information and Computation*, 100(1):1–40, 1992.

[26] Martin Olsen. Encoding mobile workflows in Reactive XML. Master's thesis, IT University of Copenhagen, 2006. In Danish.

[27] Stephen A. White. Business process modeling notation: Specification. ⟨`http://www.bpmn.org/Documents/BPMN V1-0 May 3 2004.pdf`⟩, 2004.

[28] Jacob W. Winther. Reactive XML. Master's thesis, IT University of Copenhagen, 2004.