

P-systems and X-machines

Papers dedicated to Mike Holcombe on the occasion of his 65th birthday

Marian Gheorghe · Natalio Krasnogor

Published online: 3 February 2009
© Springer Science+Business Media B.V. 2009

A number of computational paradigms have been inspired by or used for modelling the multifaceted complex phenomena present in biological systems.

One of the most recent and successful of these paradigms, called *membrane computing* or *P systems* (Păun 2002), is a vigorous research field with a significant impact on a variety of disciplines. Indeed, P systems research is a fast growing field—in 2003 the *Thompson Institute for Scientific Information* characterised the initial paper as “fast breaking” and the domain as “emergent research front in computer science”. As the name suggests, a membrane computing system formally captures various mechanisms present in cells, tissues and other more complex organisms. P systems, through complexity and formal language theory constructs, provide a general framework to build up and study nature inspired computational models.

In the more basic P system models metabolites, nutrients and other more complex (macro)molecules, which are normally found inside a cell, are encoded as multisets of simple objects or complex strings, respectively. Different chemical interactions, like transcription, translation and various enzymatic and degradation processes are, in turn, represented by rewriting rules that operate on these multisets of objects of different types. Moreover, the device, as it occurs for example in Eukaryotic cells, can be hierarchically organized by means of a well-defined structure of nested compartments. In other cases, the membranes are organised as a network of compartments in a similar fashion to what is found in a variety of biological tissues in which cells are positioned in a well-defined “matrix” structure. More complex entities, like bacterium colonies or social insects such as ants, bees, etc., are represented by dynamic structure entities that communicate, move

M. Gheorghe (✉)
Department of Computer Science, The University of Sheffield, Regent Court, Portobello Street,
Sheffield S1 4DP, UK
e-mail: M.Gheorghe@dcs.shef.ac.uk

N. Krasnogor
Automated Scheduling, Optimization and Planning Research Group, School of Computer Science
and IT, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, UK
e-mail: Natalio.Krasnogor@Nottingham.ac.uk

around or evolve according to local rules. Other dynamic behaviour is described through membrane creation, dissolution, fusion and fission.

Another computational model, introduced initially as a generic algebraic recogniser, called the *X-machine model* (Eilenberg 1974) has been widely used in specifying, modelling, verifying and testing dynamical systems. A special class of such models, called *stream X-machines*, have been developed to cope with software system specification and testing (Holcombe and Ipaté 1998). In the last decade, special classes of X-machines have been intensively studied and used to capture various processes occurring in biology (Holcombe and Paton 1998). An X-machine is a general state machine that encapsulates the simple structure of a finite state machine and adds on a powerful computational mechanism consisting of a set of functions associated to the finite state transition graph. These functions act upon some symbols taken from an input stream and their associated internal elements, called memory values, and produce output symbols that go onto an output stream and release new memory values. This model and its numerous variants, either sequential or parallel, are very suitable for specifying complex systems consisting of many components represented either as functions in a single machine or as independent machines that cooperate in communicating X-machine systems.

This special issue's goal is to collect papers describing a variety of membrane computing and X-machine models, interactions between them and relationships with other computational paradigms. A big number of research problems are investigated, ranging from theoretical aspects related to complexity, computability and formal semantics, to software engineering specification and testing, modelling biological systems and the use of domain-specific languages. The papers are summarised below.

J. Kleijn and M. Koutny in *A Petri net model for membrane systems with dynamic structure* provide a precise operational model based on Petri nets with localities and range arcs for membrane systems with dynamic structure. Rules using promoters and inhibitors as well as membrane thickening and dissolving are discussed.

The paper *Efficient simulation of tissue-like P systems by transition cell-like P systems* by D. Díaz-Pernil, M. J. Pérez-Jiménez and Á. R. Jiménez shows how a family of recogniser tissue-like P systems with symport/antiport rules solving a decision problem is simulated by a family of basic recogniser cell-like P systems built for the same problem. This is an important result that allows to transfer features regarding the performances and limitations of one model into the other.

In *The evolution of maximum cooperative P systems is NP-complete*, G. Ciobanu and A. Resios introduce a new class of P systems, called maximum cooperative P systems, which is a basic cell-like P system evolving at each step by consuming the maximum number of objects available in each compartment. The computational complexity of this model is investigated by referring to the integer linear programming problem.

P. Kefalas, I. Stamatopoulou, I. Sakellariou and G. Eleftherakis, in *Transforming communicating X-machines into P systems*, present an algorithm to transform a restricted class of communicating X-machines, with finite memory, into tissue P systems, with equivalent behaviour. The paper provides an illustrative example and investigates different available tools built for such models that make their use attractive and robust.

Finite state based testing of P systems, by F. Ipaté and M. Gheorghe, addresses the problem of testing based on P system models with non-cooperative transformation and communication rules. The approach based on the use of a minimal deterministic finite cover automaton provides a suitable approximation for the computation of a P system and relies on a method that generates the test set which is widely used in the context of finite state machine and X-machine models.

V. Manca, R. Pagliarini and S. Zorzan, in *A photosynthetic process modelled by a metabolic P system*, define a modelling strategy based on a metabolic P system specification framework with its dynamics based on log gain theory. This is then applied to model an important photosynthetic phenomenon, namely non photochemical quenching, that exhibits a very good approximation to the known experimental behaviour of the system.

Finally, in *Rule-based programming for biological modelling. Application to the modelling of the λ phage genetic switch*, by O. Michel, A. Spicher and J.-L. Giavitto, it is addressed the problem of introducing spatial information in modelling complex biological systems. Using a rule-based approach within a domain-specific language, namely MGS, the authors analyse a model of the genetic switch of the λ phage.

The papers collected in this special issue are important not only because of the significant results they present, but also for the clear future research roadmap they define. On the other hand, the type of computational models illustrated by these papers represents an appropriate opportunity to mark the 65th anniversary of Professor Mike Holcombe's birthday and to honour through this special issue his outstanding contributions to many fields in computer science. In his research, Prof. Holcombe has covered numerous fields, from algebra and theoretical computer science (Holcombe 1982) to pragmatic software engineering, formal specifications and testing (Holcombe and Ipate 1998), and modelling complex biological systems (Holcombe et al. 1998). The content of many of the papers in this special issue, dealing with new computational models, covering a large variety of research problems, investigating the interface and interactions of various formalisms and application domains, etc., reflects very closely Prof. Holcombe's original, non-standard and eclectic research interests and insights. Through his drive and vision he has challenged the international scientific community with important research questions, e.g., regarding the type of computational paradigms used in software engineering or in modelling biological systems, the robustness and completeness of the models used, the relevance of different classes of models with respect to the problems they aim to solve, the use of agile methodologies for building software systems and much more. His colleagues and collaborators acknowledge, through this dedicated special issue, Prof. Holcombe's ability to lead, challenge and produce exciting research. Happy Birthday, Mike!

Acknowledgments The editors, MG and NK, would like to express their gratefulness to the Natural Computing journal for accepting to publish this special issue, to thank from the bottom of their hearts to Professor Grzegorz Rozenberg for encouraging, supporting and helping throughout the entire process of elaborating all the materials for publication and to acknowledge a very pleasant and fruitful collaboration with Ms. Naomi Portnoy and Ms. Maria Irish Aloha Cubillan from the Journals Editor Office. NK would like to acknowledge the EPSRC for funding projects EP/E017215/1 and EP/D021847/1.

References

- Eilenberg S (1974) Automata, languages and machines. Academic Press, New York
- Holcombe WMLH (1982) Algebraic automata theory. Cambridge University Press, London
- Holcombe M, Ipate F (1998) Correct systems—building business process solutions. Springer, Berlin
- Holcombe M, Paton R (eds) (1998) Information processing in cells and tissues. Plenum Press, New York
- Păun Gh (2002) Membrane computing. An introduction. Springer, Berlin