Prologue

This is the Prologue to **The Space and Motion of Communicating Agents***, published by Cambridge University Press in March 2009.*

The informatic challenge

Computing is transforming our environment. Indeed, the term 'computing' describes this transformation too narrowly, because traditionally it means little more than 'calculation'. Nowadays, artifacts that both calculate and communicate pervade our lives. It is better to describe this combination as 'informatics', connoting not only the passive stuff (numbers, documents, ...) with which we compute, but also the activity of informing, or interacting, or communicating.

The stored-program computer, which sowed the seeds of this transformation sixty years ago, is itself a highly organised informatic engine specialised to the task of calculation. Computers work by *internal* communication among their parts; no-one expected that, within half a century, most of their work—not counting highly specialised applications—would involve *external* communication. But within twenty-five years arose networks of interacting computers; the control of interaction then became a prime concern. Interacting systems, such as the worldwide web or networks of people with phones, are now commonplace; software takes part in them, but most prominent is communication, not calculation.

These artifacts will be everywhere. They will control driverless motorway traffic, via communication among sensors and effectors at the roadside and in vehicles; they will monitor and treat our health via communication between devices installed in the human body and software in hospitals. Thus the term 'ubiquitous computing' represents a vision that is being realised.[1] In 1994 Mark Weiser, a pioneer of this vision, wrote[2]

> *Populations of computing entities will be a significant part of our environment, performing tasks that support us, and we shall be largely unaware of them.*

[1] The terms 'ubiquitous' and 'pervasive' mean roughly the same when applied to computing. I shall only use 'ubiquitous'.

[2] Citations of related work will be found in Chapter 12.

This suggests that informatic behaviour is just one of the kinds of phenomena that impinge upon us. Other kinds are physical, chemical, meteorological, biological, ..., and we have a good understanding of them, thanks to an evolved culture of scientific concepts and engineering principles. But understanding still has to evolve for the behaviour of a population of informatic entities; we have not the wisdom to dictate the appropriate concepts and principles once and for all, however well we understand the individual artifacts that make up the population.

This understanding is unlikely to evolve in large steps. The qualities we shall attribute to ubiquitous systems are extraordinarily various and complex. Such a system, or its component agents, will be *self-aware*, possess *beliefs* about their environments, possess *goals*, enter *negotiation* to achieve goals, and be able to *adapt* to changing circumstances without human intervention. Here is an incomplete list (in alphabetical order) of concepts or qualities, all of which will be used to specify and analyse the behaviour of ubiquitous systems:

> agent, authenticity, belief, connectivity, continuous space, data protection, delegation, duty, encapsulation, failure management, game theory, history, knowledge, intelligence, intention, interaction, latency, locality, motion, negotiation, protocol, provenance, route, security, self-management, specification, transaction, trust, verification, workflow.

Much has been written about principles and methods of system design that can realise these qualities, and much experimental work done in that direction. That body of work is one part of the background for this book, and is discussed in greater detail—with citations—in Chapter 12.

The design task for ubiquitous systems is all the harder because they will be at least an order of magnitude larger than present-day software systems, and even these have often been rendered inscrutable by repeated adhoc adaptation. Yet ubiquitous systems are expected to *adapt themselves without going offline* (since we shall depend upon their continuous operation). It is therefore a compelling scientific challenge to understand them well enough to gain confidence in their performance. This has been adopted as one of the Grand Challenges for Computing Research by the UK Computing Research Committee.

Looking at our list of system qualities in greater detail, we notice that some are more sophisticated, or 'higher-level', than others. Some, such as trust, are properties normally attributed to humans, not to artifacts. But when an assertion such as 'A trusts B' is made at a high level of modelling, we expect it to be realised at a lower level by A's behaviour; for example, A may grant B's requests on the basis of evidence of B's past behaviour.[3] If a stratification of modelling can be achieved by such realisations, then the task of description and design of ubiquitous systems will become tractable.

To model ubiquitous systems of artifacts will be hard enough. But, as the reader may already be thinking, such systems will also contain natural organisms. They will occur at dramatically different levels; we already mentioned people with phones, and we should also include more elementary biological entities. We should seek to model not only interactive behaviour among artificial agents, but also interaction with and

[3] A behaviourist philosopher might insist that this is the *meaning* of 'A trusts B', even for humans.

among natural agents. Ultimately our informatic modelling should merge with, and enrich, natural science.

Space

Where can we start, in building a stratified model of ubiquitous systems? The key term here is 'stratified'. The agents of a ubiquitous system stand to it in the same relation as musical instruments stand to an orchestra. Instruments existed long before orchestras; how to combine them in groups and then into the whole would have puzzled the early virtuosi of each instrument. It would have gradually emerged how the physical qualities of each instrument would combine to realise qualities of the group; for example, how the tone-colours of different wind instruments would yield the more abstract quality of tenderness, or of humour, in a wind quartet. Thus gradually emerges the huge spectrum of qualities of a whole orchestra.

Where this analogy becomes strained is in the brute fact of *size*; a ubiquitous system will involve millions of agents, whereas an orchestra has a mere hundred instruments.

Let us return to stratification. In a ubiquitous system, a quality attributed to a larger subsystem must be realised by simpler properties of smaller subsystems or of individual components. This realisation, in turn, surely depends on how the system and its subsystems are constructed. So, to realise system qualities, we must first understand possible structures for ubiquitous systems. We may be grateful for this conclusion; it poses a challenge more accessible than that of realising human-like qualities in a machine. Structure is itself difficult, especially for systems that will reorganise their own structure. But one can at least make proposals about the possible ingredients of structure, without being bewildered by the immense range of behavioural qualities that it will support.

This book works out such a proposal. It starts from the recognition that a notion of *discrete space* is shared by existing informatic science on the one hand and imminent ubiquitous systems on the other. This space involves just three of the concepts listed above: *agent*, *locality* and *connectivity*. When we come to reconfiguration of the space we must consider two more of those concepts: *motion* and *interaction*.

At this point, the reader may object: "How can you be sure that we can base our understanding of system behaviour on these concepts? You aim to explain systems that have some of the intelligence of humans, and these chosen concepts are at the level of the basic structure of matter! Your proposal is analogous to claiming that we can base our understanding of the brain on chemistry." The simple answer is: I am *not* sure that these concepts are sufficient; but I do claim they are necessary. Brain researchers are faced with a task harder than ours in many ways; but they are fortunate that much chemistry was known before brain research began. We, on the other hand, have work to do to formulate the analogue of chemistry for ubiquitous systems.

Let us now turn to discussing a space of agents, based upon locality and connectivity. Since these ideas pervade the whole book, we shall denote them by the simpler words *placing* and *linking*. It is instructive to reflect how placing and linking run through existing informatics. Even before the stored-program computer, calculation depended on ways to organise space—not the space of Euclidean geometry, but a dis-

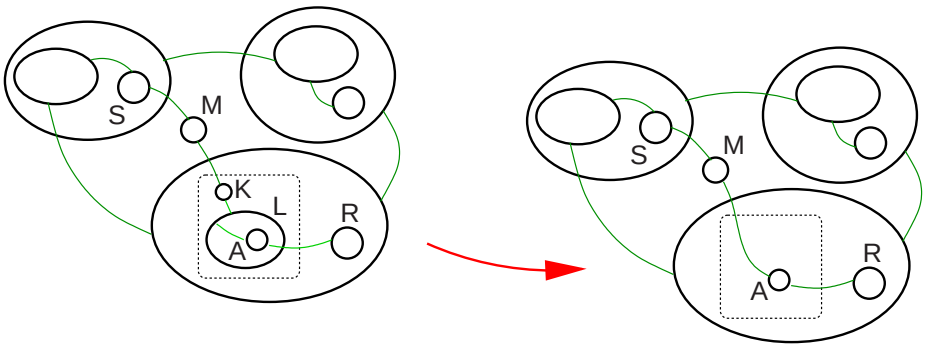
crete space involving properties like adjacency and containment. Arabic numerals use one-dimensional placing to represent the power of digits; this allows two-dimensional placing to be used to arrange data in the basic numerical algorithms—addition, multiplication, and so on. Algorithms for solving differential equations with a manual calculator deployed the use of placing for data and calculation in sophisticated ways.

In stored-program computers the space became more refined. Programs use one storage register to 'point at' another; that is, an integer variable is used to index through a sequence of elements (where previously a human calculator would run his or her finger through the sequence). Thus linking became distinct from simple properties of placing, such as adjacency or containment. Placing and linking became independent; for example, an element *placed* within an array can be *linked* to something else occupying a distant place.

It is striking that wireless networks allow us similarly to think of linking as independent of physical placing in ubiquitous systems. We assume this independence when we describe the internet. Moreover placing and linking can be either physical or virtual; we even mix the two within a single system, using the relationships of physical entities as metaphors for relating the virtual ones. These metaphors abound in our vocabulary for software: flow chart, location, send and fetch, pointer, nesting, tree, etc. Concurrent computing expands the vocabulary further: distributed system, remote procedure call, network, routing, etc.

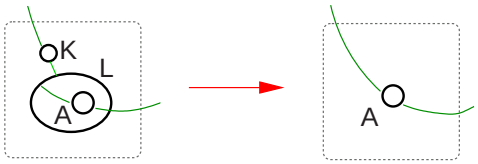
Motion

Any model of ubiquitous systems based on placing and linking, whether of physical or virtual entities or both, must accommodate motion and interaction. In fact it is unsatisfactory to separate these two concepts, so I tend to conflate them. (In moving into a room, I can be said to interact with the room.) The picture below illustrates a mixture of the physical with the virtual; it also shows how a system may reconfigure itself.



It represents a change of state in which a message M moves one step closer to its destination. The three largest nodes may represent countries, or buildings, or software agents. In each case the sender S of the message is in one, and the receiver R in another.

The message is en route; the link from M back to S indicates that the message carries the sender's address. M handles a key K that unlocks a lock L, reaching an agent A that will forward the message to R. This unlocking can be represented by a *reaction rule*; such rules define how a part of the system may change both its placing and its linking. A rule that defines the above reconfiguration is as follows:



Here, both key and lock are virtual; but of course physical reconfiguration can happen in the same system. For example, at any time the (physical) receiver R may move away from her location. Can the message chase R and catch her up? Perhaps some interaction between her and the forwarding agent A makes this possible. Indeed, as she goes, R may construct an informatic record of her (physical) journey, and send it back to assist the forwarding agent. So there is no doubt that a model of space and interaction has to coordinate informatic and physical entities.

I shall show that these diagrams, and their reconfiguration, are a presentation of a rigorous theory. I aim to develop that theory to the point that it can begin to underlie experiments with real systems, and so form the basis for theories that deal with the more subtle notions mentioned above, such as beliefs, self-awareness and adaptability.

The bigraph model

The graphical structures we have just illustrated will be called *bigraphs*. Like an ordinary graph, a bigraph has nodes and edges, and the edges link the nodes. But unlike an ordinary graph, the nodes can be nested inside one another. So a bigraph has *link* structure and *place* structure; hence the prefix 'bi' in bigraph.[4] Bigraphs will be introduced with more detail in Chapter **??**, but a few comments will be helpful here:

- The two structures—placing and linking—will be treated independently in the basic theory of bigraphs. This accords with our observation that both pointers in computer programs and wireless links in the real world can arbitrarily cross place boundaries. This independence property has another benefit; when first introduced, it was found to simplify the theory of bigraphs dramatically.

- The reader may ask "What is the space in which bigraphs live and move?" The answer is that bigraphs themselves *are* the space of the model. My proposal is that this notion of space is enough to represent an enormous range of structures. Experiment with this simple space will reveal whether and when a more complex space is required.

[4]The term 'bigraph', as used here, was introduced in 2001. I recently found that the term was already used then as a synonym for 'bipartite graph', a well-established notion in graph theory. The meanings differ, but the use of the same term is unlikely to cause confusion.

- A single bigraph may represent both virtual and physical entities (a country, a message, ...). This may seem surprising, but creates no difficulty; indeed, it is very convenient. To push our example a little further, imagine that the receiver R is a traveller who carries a laptop in which she makes a schematic map of the places she visits. This physical laptop is then represented by a node in the bigraph, and the virtual structure (the map) it contains may be represented by the contents of that node.

Generality

Let us now discuss the degree of generality achieved by bigraphs. Will they serve as a platform for building ubiquitous systems? To answer this we must present the bigraph model as a design tool, to be used not only for analysis but even as a programming language; then experiments can be done to reveal its power and generality.

But to establish the model as a candidate for this long-term role, we must first make sure that it accommodates, or generalises, already existing theories of interactive agents. This shorter-term challenge is more well-defined. We must encode each previous model—including its rules of interaction—into bigraphs. Indeed bigraphs should not only represent the agents and reactions of previous models; they should also provide theory that applies uniformly to those models. In other words, bigraphs should tend to unify theories of processes.

This book gives priority to the latter challenge: to generalise existing process models. Therefore in Chapter 12, the final chapter, I explain how bigraphs have drawn ideas from preceding models, and were developed in order to strengthen and generalise their theory. The result has been positive. To give perspective, I give a brief summary here. (A little familiarity with process models will be helpful in the next paragraph, but it can be skipped.)

Each process model (for example Petri nets, CSP, mobile ambients, π-calculus) defines processes syntactically, and then presents its rules of interaction. Thus each model is represented in bigraphs by two parameters: a *sorting discipline*—which includes a *signature*—that make the bigraphs represent the model's formal entities, and a set of *reaction rules* to represent their behaviour. These two parameters yield a *bigraphical reactive system* (BRS) that is specific to the model. BRSs for several process models are presented in the book. Often the agreement with the model is exact; in other cases nearly exact. It is worth making specific points:

- For the purpose of both analysis and programming, many existing models have a convenient algebraic (i.e. modular) representation of processes. In bigraphs there is a uniform algebraic presentation, and this bears a close relation to that of existing models. Thus bigraphs contribute uniformity of expression.

- Some calculi, including CCS and the π-calculus, define what it means for two processes to behave alike. This is called *behavioural equivalence*. A typical example is bisimilarity. Such an equivalence is usually a congruence—i.e. it is preserved by insertion of the processes into any environment. The proof of congruence has typically been somewhat ad hoc. Bigraphs provide a degree of

uniformity here; in bigraphs not only do we treat bisimilarity uniformly across process calculi, but we also provide a uniform proof of congruence.

- For most of the book we retain the full independence of placing and linking; this yields most of the results. However Section **??** defines uniformly a way to relax this independence; it defines how to localise a link and thereby to represent the *binding* of a name; this has allowed us to to handle (for example) the π-calculus.

Thus the aim to generalise or subsume existing process calculi serves as a focus for developing our model. But these very calculi do not only aspire to an engineering role, as a means to express and analyse the design of complex systems; they also aspire to advance the fundamental science of informatics. They represent a challenge to the models of computation that were dominant in the twentieth century. By exposing computation as an especially disciplined form of informatic behaviour, they have opened the way to a science of such behaviour in which the determinacy and hierarchy found in traditional computing are the exception, not the rule. They replace calculational structure with communicational structure.

This book can therefore be seen as advancing the science of communicational structures. For example, I carry out much of the work of the book at the level of *wide reactive systems*, more general than bigraphs. But by working in the explicit space of bigraphs I attempt to bridge between the engineering and the science of communication. Indeed, such a model extends the repertoire of models available to natural scientists. For example, with the help of a stochastic treatment of interaction we are able to apply the bigraphical model to the predictive analysis of biological systems. This application lies beyond the scope of the present book, but is explained a little more in Section **??**.

Rigour

Working at a broad frontier of informatics, spanning science and engineering, demands prioritisation; as I have already stated, it lies beyond the scope of a single book both to explore all possible applications (natural and artificial) and to establish a model in full detail. I have chosen to do the latter because, as we saw in the preceding section, there already exist many precise models in the form of process calculi, and they pose an accessible challenge—to recover them as instances of a more impartial study. This challenge, to establish commonality among existing formal models, must itself be addressed formally if we are to make it a firm platform on which to tackle a still wider range of applications. But I have interleaved formal development with discussion, and have not relied on previous knowledge of any particular mathematical theory.

I use the medium of category theory, but the level at which I use it is elementary, and I define every categorical concept that I use. Large informatic systems are complex, and any rigorous model must control this complexity by means of adequate structure. After many years seeking such models, I am convinced that categories provide this structure most convincingly. It is true that they can also express deep mathematical abstractions, many of which at present lie beyond the interest of informatic scientists. But there is a sharp division of motive between pursuing these abstractions per se and using categorical primitives as a means to understand informatic structure. The work

in this book is of the latter kind. Readers familiar with categories will follow their use here without difficulty; others who wish to tame informatic structure may find this work a pleasant way to learn some mathematics suited to that purpose.

Models are built to aid people's understanding, and different people seek different levels of understanding. Engineering scientists seek a rigorous model; software designers seek something softer, but with equal intuitions, and this is even more true for their client companies and for end-users. So we would like to know that softer models of communicating agents can arise from our rigorous model. Fortunately, by their very nature these systems involve a concept of space, which is reflected in the idea of bigraphs and lends itself to informal understanding based upon diagrams. Throughout the book I work as much as possible with bigraphical diagrams; they express the rigorous ideas but do not replace them.

Deployment

It is one thing to develop a rigorous model; quite another thing to bring it into use by those concerned mainly with applications. But this usage is a primary goal for our model; moreover, it is only by deploying the model in applications that we can subject it to stringent testing.

Even protypical applications tend to be complex; one need only think of phenomena in ubiquitous computing and in biology. It follows that software tools are essential for exploring the efficacy of the model, both for scientific analysis and for advanced software engineering. Such tools have several roles: in *programming* and *specifying* complex systems; in *simulating them*, with the help of stochastic dynamics; and in *visualising* them at various levels of abstraction, exploiting the graphical presentation inherent in the model.

Work in these directions is under way at the IT University (ITU) in Copenhagen, as outlined in Chapter 12. A strategy exists for modular tool development, which can proceed in collaboration among different institutions. I would be glad to hear from anyone willing to contribute seriously to this development.

Outline of the book

Bigraphs are developing in various ways. All these developments are based upon *pure* bigraphs: those in which the independence of placing and linking is strictly maintained. So most of the book is devoted to pure bigraphs, whose theory is more or less settled. Part I presents their structure; Part II handles their behaviour; and Part III deals with their development, past and future.

In Part I, Chapter 1 introduces bigraphs starting from standard notions in graph theory. The main idea of bigraphs is to treat the placing and the linking of their nodes as independently as possible. Chapter 2 defines bigraphs formally, together with the operations that build them; it then introduces various kinds of category that will help to develop their theory. Chapter 3 develops the algebra of bigraphs, with operations for both placing and linking; it also derives operations familiar from process calculi.

Chapter 4 defines relative pushouts, a categorical tool for structural analysis. Chapter 5 applies this tool to bigraphs, preparing for the later derivation of transitions. Chapter 6 develops a sorting discipline for bigraphs that is reminiscent of many-sorted algebra.

In Part II, Chapter 7 defines the notion of a wide reactive system (WRS), more general than bigraphs. For such systems it defines reaction rules and derives (labelled) transition systems; it then obtains important results such as the congruence of bisimilarity. WRSs have an abstract notion of space, enough to allow reaction to be confined to certain places. Chapter 8 specialises this work to bigraphs, yielding the more refined notions of a bigraphical reactive system (BRS) and its transition systems; it also identifies certain well-behaved kinds of BRS. Chapter 9 uses link graphs, a simplified version of the theory, to analyse behaviour in arithmetic nets and Petri nets. Chapter 10 applies bigraphs to CCS, and recovers its original theory.

In Part III, Chapter 11 discusses several developments beyond pure bigraphs. First, it examines how to *track* the identity of agents through interaction; this would allow one to express, and to verify, assertions about a BRS such as "Each agent receives each message at most once" or "Mary has visited three rooms since she entered the building". Second, it proposes a generic way to represent agents with infinite behaviour using finite bigraphs, with the help of rules for structural *growth*. Third, it discusses how to constrain placing and linking so that certain links have *scope*, or are *bound*, in the familiar way that variables in a programming language have scope or are bound as formal parameters of a procedure. Finally, it summarises recent work on the *stochastic* interpretation of bigraphical systems; this is essential for simulating nondeterministic systems, in particular in biological applications, where the more likely of two possible reactions is that which is attributed the higher rate in an exponential distribution.

Chapter 12 outlines how bigraphs have developed, and discusses related work with full citations. These show how much the work of this book owes to my close colleagues, as well as to influences from other research initiatives.

Using the book

The chapters need not be read in strict sequence. Mostly, later chapters point back to what they need from earlier ones. Figure 1 gives a guide to the dependency among chapters. For example if you reach Chapter 8 by going down the left side, you read about bigraphs and then get the theory when you need it; if you reach it down the right side you stay at the general level of reactive systems as long as possible. Leaping ahead may also be useful; for example, those who know something of process calculi may leap from Chapter 1 to Chapter 10, to gain motivation for returning to the intervening chapters.

The book is suitable for teaching yourself; there are many exercises, and solutions to all of them. The book is suitable for a Masters' course, where the amount of theory included can be adapted to the students' knowledge. Parts of the book can be used for an optional final year Undergraduate course.

The book can also serve as the foundation for a lecture course that concentrates upon the intuition of bigraphs and their experimental use. I have designed such a course; from my website, `http://www.cl.cam.ac.uk/~rm135`, the reader may

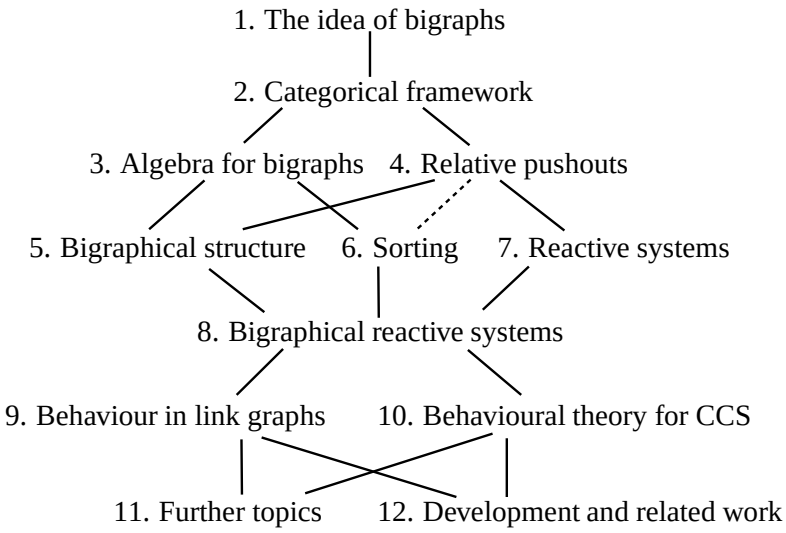


Figure 1: Dependency among the chapters

---

download a sequence of seventy or more slides that I have used. Accompanying them is (or, at the time of writing, will soon be) a slide-by-slide narrative, linking the slides together and making copious reference to this book—especially for locating the underlying rigorous development. This combination of slides and narrative will evolve in response to my own experience, and to the experience of others who use them. I shall be delighted to receive comments by email (`rm135\@ccam.ac.uk`) from anyone, based on such experience; thus I hope to improve the slides, the narrative and ultimately the book itself.

Acknowledgements

I owe much to early collaboration on bigraphs with Jamey Leifer and Ole Høgh Jensen. I am most grateful for their creative insights. I thank Philippa Gardner and Peter Sewell for important contributions in work that led from action structures (a previous model) to bigraphs. Several people have generously given time to careful reading, helping me to express things better: Samson Abramsky, Mikkel Bundgaard, Troels Damgaard, Marcelo Fiore, Sam Staton and David Tranah.

I also thank warmly all those I have worked with, or learnt from, in this subject over nearly thirty years, in particular: Martín Abadi, Samson Abramsky, Jos Baeten, Martin Berger, Jan Bergstra, Gérard Berry, Lars Birkedal, Clive Blackwell, Gérard Boudol, Mikkel Bundgaard, Ilaria Castellani, Luca Cardelli, Adriana Compagnoni, Troels Damgaard, Vincent Danos, Rocco De Nicola, Hartmut Ehrig, Marcelo Fiore, Philippa Gardner, Arne Glenstrup, Andy Gordon, Matthew Hennessy, Thomas Hildebrandt, Jane Hillston, Yoram Hirshfeld, Tony Hoare, Kohei Honda, Alan Jeffrey, Ole

Høgh Jensen, Jan-Willem Klop, Jean Krivine, Cosimo Laneve, Kim Larsen, Jamey Leifer, Alex Mifsud, George Milne, Kevin Mitchell, Faron Moller, Ugo Montanari, Uwe Nestmann, Mogens Nielsen, Catuscia Palamidessi, David Park, Joachim Parrow, Carl-Adam Petri, Benjamin Pierce, Gordon Plotkin, John Power, Sylvain Pradalier, K.V.S. Prasad, Corrado Priami, Michael Sanderson, Davide Sangiorgi, Vladi Sassone, Peter Sewell, Mike Shields, Sam Staton, Bernhard Steffen, Colin Stirling, Chris Tofts, Angelo Troina, David Turner, Rob Van Glabbeek, Bjorn Victor, David Walker, Glynn Winskel, Nobuko Yoshida.

I acknowledge the Préfecture of the Île-de-France Region for the award of a Blaise Pascal International Research Chair, which enabled me to advance this work during a recent year in Paris. I warmly thank Jean-Pierre Jouannaud and Catuscia Palamidessi, who were my welcoming hosts at École Polytechique at Saclay during this period.

Robin Milner
Cambridge, June 2008