Domain Theory for Concurrency
—New Categorical Foundations

EPSRC Research Grant Application
Case for support

Glynn Winskel   Marcelo Fiore

Computer Laboratory
University of Cambridge

February 2004

1 Previous track record

Glynn Winskel has been Professor of Computer Science in the University of Cambridge Computer Laboratory since October 2000. For the previous twelve years he was Professor of Computer Science at the University of Aarhus, Denmark, where from 1994 he was Director of BRICS, a research centre, and later PhD School, in Basic Research in Computer Science, funded by the Danish National Research Foundation. Before that he was Lecturer then Reader of Theoretical Computer Science in the University of Cambridge Computer Laboratory, following two years as Research Scientist, at Carnegie-Mellon University, USA. He has previously received research funding from the Royal Society, the EPSRC and various European programmes, and most significantly from the Danish Research Foundation to found BRICS. He is the author of many publications and a successful textbook on the formal semantics of programming languages, published by MIT-Press, which also appears in Italian and Chinese translations. His research has ranged over model checking, domain theory and denotational semantics, models for concurrency, applications of category theory, and the verification of hardware and security protocols. His PhD thesis of 1980 worked out the basic theory of event structures, a fundamental and increasingly relevant model of computation. It is a newly-found confluence of the early work on event structures [38, 28] and later work on presheaf models for concurrency [24] which forms one important axis of this research application. He was a main lecturer at the Fields Institute Summer School, Ottawa, 2003. Last year his paper on "Events in Security Protocols" with his PhD student Crazzolara received the British Computer Society Brendan Murphy Memorial Prize in the field of Networking and Distributed Systems.

Marcelo Fiore has been lecturer in the University of Cambridge Computer Laboratory since October 2000. He holds the EPSRC Advanced Research Fellowship "Semantic Theories for Functional and Concurrent Computation." Before he was lecturer at the University of Sussex. Previously he had been a research fellow for Prof. Gordon Plotkin at the Department of Computer Science in the University of Edinburgh. His main research interests are in mathematical models of computation and interaction, in particular, in developing models for the denotational and operational semantics of languages and systems, and in applying these models to the design of type systems and program logics and reasoning about system behaviour. He has recently branched out into combinatorics and computer algebra. His first research programme initiated the research area of axiomatic domain theory. This project started in his PhD thesis "Axiomatic Domain Theory in Categories of Partial Maps" published in the British Computer Society Distinguished Dissertation series in 1995. He is currently working to extend the foundations of semantics of functional and concurrent computation towards a mathematical theory of general processes encompassing discrete, continuous, and hybrid systems.

Francesco Zappa Nardelli (www.di.ens.fr/~ zappa) has been a PhD student in the Département d'Informatique of the École normale supérieure (ENS) since October 2000. He previously obtained a Master's Degree in Computer Science from the University of Pisa, Italy in 2000. In the same year, he obtained a *Diplome d'études approfondis* (MSc) in Theoretical Computer Science from University of Paris Sud, France. His excellent grades there led to the funding of his PhD studies at ENS by the French government. He obtained his PhD in December 2003. His research interests focus on the semantics of concurrent computation,with special attention to formalisms that include primitives to model distributed and mobile systems. Among the most interesting results of his thesis is the first operational characterisation of behavioural equivalence for Mobile Ambients, a process language widely studied to model mobile computations—this work with Merro solved an open problem. He has also investigated operational and denotational models of higher-order computation with name generation, and, more recently, axiomatic characterisations of behavioural theories of higher-order processes.

Additional expertise

Collaboration has recently begun with Pierre-Louis Curien, his colleagues and students, and Patrick Baillot on aspects of this proposal. This will be furthered through Winskel being invited professor in Paris for a month in the coming academic year. We expect and will encourage close contact with Andy Gordon and Luca Cardelli on the subject of bisimulation for mobile ambients.

The Computer Laboratory at Cambridge has one of the world's leading research groups in theoretical computer science, having Gavin Bierman, Anuj Dawar, Marcelo Fiore, Mike Gordon, Robin Milner (emeritus), Alan Mycroft, Lawrence Paulson, Andrew Pitts, Peter Sewell and Glynn Winskel. There are very good relations with Martin Hyland and associates at the Department of Pure Mathematics, as well as with Luca Cardelli, Nick Benton, Cedric Fournet, Georges Gonthier, Andrew Gordon, C.A.R Hoare, and associates at Microsoft Research Ltd (the European research laboratory of Microsoft Corporation). All these peoples' research interests connect with the areas of this proposal and contribute to a stimulating research environment.

There are good working relations with Gordon Plotkin, John Power, Alex Simpson and Ian Stark at the University of Edinburgh on topics in this proposal, as well as with Matthew Hennessy, Guy McCusker, Julian Rathke and Vladimiro Sassone at the University of Sussex. More broadly afield, there are existing collaborations with researchers at PPS, director Pierre-Louis Curien, École normale supérieure Paris, Université

Paris Nord, a budding collaboration with Benveniste's group at IRISA, Rennes, and through Nigel Walker with British Telecom (both concerned with applying research here in telecommunications), Mogens Nielsen at BRICS, University of Aarhus, Thomas Hildebrandt at the IT University in Copenhagen, André Joyal and Prakash Panangaden in Montreal, all connecting to the proposal topic. Fiore's PhD student Sam Staton and Winskel's former PhD students Mikkel Nygaard (Aarhus), Daniele Varacca (ENS) and present student Lucy Saunders-Evans work on related areas. Consultation is anticipated with researchers at the universities of Birmingham (Escardó, Jung, Kwiatkowska) and Oxford (Abramsky, Ong), Manchester (Schalk, Simmons), and Imperial and QMW Colleges (Robinson, O'Hearn), London. More generally, being part of the ESPRIT Working Group on *Applied Semantics* provides a useful network of expertise on which to draw.

Winskel is invited speaker at several meetings this year including the International Symposium on Domain Theory (ISDT'04) in Xi'an, China, an invitation which should provide excellent opportunities for extending research collaboration.

2 Proposed Research

2.1 Background

Denotational semantics and domain theory, by providing a *global* mathematical setting for sequential computation: place programming languages in connection with each other; connect with the mathematical worlds of algebra, topology and logic; and inspire programming languages, type disciplines and methods of reasoning. In concurrent/distributed/interactive computation that global mathematical guidance is missing, and domain theory has had little direct influence on theories of concurrent computation. One reason is that classical domain theory has not scaled up to the more intricate models used there.

Broadly speaking approaches to theories of concurrent computation are either based on a specific mathematical model of processes or start from the syntax of a process calculus. Among the variety of models for concurrency, one can discern an increasing use of causal/independence/partial-order models (such as Petri nets and event structures) in which computation paths are partial orders of events. Independence models thread through partial-order model checking [14], security protocols [35], nondeterministic dataflow [13], self-timed circuits [15], term-rewriting, game semantics [1], and the analysis of distributed algorithms [25]. There are a variety of process calculi. These are most often based on an operational semantics. Following on from the Pi-Calculus [27, 34], new-name generation is central to almost all calculi of topical interest. Many are higher-order (allowing process-passing) which presents a challenge in understanding suitable equivalences, of which forms of bisimulation are prevalent.

Theories of concurrency form a rather fragmented picture. Relations between different approaches are often unclear; ideas are rediscovered (for example, special event structures reappear as "strand spaces" in reasoning about security protocols). A lot of energy is used on local optimisations to specific process calculi, optimisations that may obscure connections and the global picture. The lessons learnt often remain isolated for lack of the commonality a global framework would provide.

A domain theory which handled independence models, name-generation, higher-order processes and possessed an operational interpretation would provide a global mathematical framework for present theories of concurrency, and would point the way forward to new languages and analytical tools. In case incorporating independence models into a domain theory seems a tall order, there are now arguments (based on event-structure representations of process denotations—see below) that the operational semantics associated with a domain theory for concurrency will involve event structures. It should be remarked that a *traditional* use of powerdomains [33], based on domains of resumptions will fall short because, being based on a nondeterministic choice of actions one at a time, it cannot accommodate the potentially complex structure of computation paths.

How do we work towards such a domain theory for concurrency? The potentially complicated structure of computation paths suggests building a domain theory directly on computation paths. This line has been followed in what seemed originally to be two different directions: that of Matthew Hennessy's semantics for higher-order processes, in which process equivalence is a form of trace equivalence [17]; and categories of presheaf models, in which process equivalence is a form of bisimulation [39].

Presheaf models form one major starting point (and mathematical weapon [8]) for this proposal. Intuitively, a presheaf over a category $\mathbf{P}$ can be thought of as a process with computation paths which have shapes described by objects in $\mathbf{P}$.[1] In this sense, a presheaf over $\mathbf{P}$ is a process of *type* $\mathbf{P}$. Presheaf categories form a mathematically rich and informative model of processes (they include traditional models like synchronisation trees and event structures [24]), possess a general notion of bisimulation (based on open maps [24]), and have yielded several interesting new higher-order process languages (HOPLA [31] and Affine HOPLA [30]). Presheaf categories play the role of domains of processes. There is a shift from the traditional view of domains as special partial orders to domains as special categories, a view given weight by precursive work on categories of models for concurrency [40].

A guiding principle in designing an operational semantics from a presheaf semantics has been that elements of a presheaf denotation of a process term t at a path p correspond to derivations of judgements $t \xrightarrow{p} t'$ in the operational semantics. (This leads to *strong correspondence* results between operational and denotational semantics [29].) So the methodology is to proceed from the denotational model to an operational semantics. This involves study of the structure of presheaf elements (and their reading as derivations), which has recently uncovered a strong connection with event structures; a definable presheaf can often be represented by an event structure, the finite configurations of which correspond to the presheaf's elements [29]. It is the new-found confluence of the early work on event structures [38, 28] and later work on presheaf models for concurrency [24] which forms a major axis of this research application. The other major axis is the recent uncovering of a highly expressive met-

[1] Mathematically, a presheaf X over a category $\mathbf{P}$ is a contravariant functor from $\mathbf{P}$ to the category of sets, $\mathbf{Set}$; the elements of the set $X(p)$, at a path object p of $\mathbf{P}$, stand for the process's computation paths of shape p, and correspond to the different ways the process can realise p. It's now recognised that Hennessy's domains are analogues of presheaf categories $[\mathbf{P}^{op}, \mathbf{Set}]$, but where the role of the category of sets $\mathbf{Set}$, in the "branching" semantics, is replaced by the simple order $0 < 1$, in the "trace" semantics.

alanguage and operational semantics for name generation and higher-order processes (new-HOPLA) from presheaf models [43]—an encouraging example of the power of presheaf semantics to suggest new primitives, new ways to derive operational semantics, and feed forward into the design and analysis of process languages.

2.2 Programme and Methodology

It is proposed to develop a new domain theory which provides a global mathematical setting for concurrent and mobile computation, a domain theory which is extensible, scales up to the more intricate models needed in distributed systems, and through taking on a mathematical life of its own yields new, perhaps unexpected, ways to structure and analyse processes—as has already happened in HOPLA where prefix is a linear logic exponential, Affine HOPLA with its 'mysterious' tensor, and new-HOPLA [31, 30, 43]. Specifically, the research works towards a domain theory which simultaneously handles causal/independence models such as event structures, name generation, higher-order processes, nondeterministic dataflow, and possesses an operational interpretation. Its methodology and techniques rest on recent successes in extracting process languages, type disciplines and operational semantics from categories of presheaf models of processes (where "domains" are presheaf categories) [30, 31], together with the very recent discovery by Nygaard and Winskel of representations of the presheaf denotations as spans of event structures [29]. The domain theory for concurrency is based on new foundations, (bi)categories forming models of linear logic in which the domains are categories standing for the types of computation paths a process may perform.

The presheaf-based approach reveals a connection with linear logic by uncovering categories which are models of linear logic and at the same time models of processes, in which the type of a process specifies the shapes of computation paths it may perform. Linearity is about how to manage without a presumed ability to copy, so with hindsight can be expected to permeate distributed computation, where the copying of processes is often limited. The categories uncovered are linear/affine/continuous according to whether their maps use (exactly one)/(at most one)/several copies of the input process.

So far, recognition of the place of linearity has led to new languages HOPLA (Higher Order Process Language [31]) and Affine HOPLA [30] designed from the continuous categories and affine categories. Both languages encode directly a rich variety of the known process languages. But they are the beginning rather than the end of the story: we lack a full operational semantics for Affine HOPLA, essentially because of its tensor type and a phenomenon of entanglement; we lack some mathematical underpinnings to justify extensions, for instance of HOPLA, to name generation; the extremely rich affine category of presheaf models, which inspired and interprets Affine HOPLA, supports important operations (for instance those of nondeterministic dataflow and parallel compositions of event structures) which are often not expressible within Affine HOPLA syntax.

In the search for a domain theory for concurrency, one accompanied by an expressive syntax and operational reading, the affine and continuous categories of presheaf models are an informative place to begin. This is both because of *strong correspondence* results showing how elements of presheaf denotations correspond to derivations in an operational semantics [29] and the operational reading provided by representations in terms of event structures [29]. The representations suggest variations, in particular new (bi)categories of event structures which already model higher order processes and promise to give semantics to features currently beyond the reach of traditional domain theory. By working towards a domain theory which subsumes the intricate models used in analysing distributed computation we expect to uncover new fundamental insights into the design and analysis of process languages. Here there are good grounds for optimism; for example, early work of Nielsen, Plotkin and Winskel on unfoldings of Petri nets formed the basis of several analytical tools including the early model checker of McMillan and tools for self-timed circuits [2, 14, 15], while event structures have been widely applied, recently in the analysis of security protocols [35, 42].

We will achieve these aims through the following research tasks—further technical details can be found in recent papers at www.cl.cam.ac.uk/˜ gw104.

Operational semantics from presheaf models

Following [24], a process may be viewed as a presheaf over a category with objects paths (or computation-path shapes) and maps saying how one path extends to another. Presheaf models are very informative mathematically. Their mathematical structure can guide us towards an operational semantics. A presheaf X over a path category $\mathbf{P}$ is associated with its category of elements which we can view as a form of transition system, but where transitions have the more general shape of objects of $\mathbf{P}$. Given a specific language like Affine HOPLA [30] and its presheaf denotational semantics we can hope for a corresponding operational semantics and a coinductive characterisation of open-map bisimulation. Progress has been made here [30]. The guiding principle in designing an operational semantics has been that elements of a presheaf denotation of a process term t at a path p correspond to derivations of judgements $t \xrightarrow{p} t'$ in the operational semantics. (Proof of adequacy now involves a novel generalisation of logical relations to an analogue based on sets of realisers rather than simple truth values.) But it has proved very challenging to get an operational semantics for affine processes at higher-order. Linearity obliges us to work with rather complicated environments, and a form of "entanglement" in the execution of processes—choice in one component of a term of tensor type generally affects choice in another. It is the interaction of the environments with higher-order processes which has been problematic in giving an operational semantics to full Affine HOPLA. These complications largely disappear when arbitrary copying is allowed (cf. the simple operational semantics of HOPLA [31]) but at the loss of the tensor operation around which so many useful constructions and questions centre, in particular the relation with event structures and the "relational" treatment of nondeterministic dataflow [21].

A major challenge is to present a coinductive characterisation of open-map bisimulation at higher-order. There are currently only patchy results. One inroad is to design a process language which more adequately reflects the mathematical structure present in the models, in particular the duality between input and output present there. The mathematics for this enterprise is developed in [8].

The same mathematical foundations provides for the introduction of internal actions and weak bisimulation in the general context of presheaf models [7]; an aim is to arrive at a metalanguage which supports the general abstraction and hiding operations of that paper.

Event-structures as higher-order processes

A way to understand the operational content of presheaf models is through representation theorems relating presheaf denotations to more traditional process models. It has come as a recent surprise that the presheaf denotations of first-order processes in Affine HOPLA can be represented by event structures; the elements of definable presheaves can be understood as finite configurations of an event structure. (In more detail, maps definable in Affine HOPLA by open terms can be represented by certain spans of event structures with composition given by using pullbacks.) This sheds light on the affine tensor and the form of entanglement associated with it, revealing the affine tensor as a form of parallel composition of event structures and entanglement as a pattern of concurrency/conflict. It's been noticed by Winskel that the event-structure semantics extends to all types, so higher-order processes [29]. Though, as one would expect, the event-structure semantics diverges from the presheaf semantics at higher-order; the event-structure semantics is analogous to stable domain theory [3]. One can hope to recover an extensional semantics by using bistructures [12].

Within Affine HOPLA we can define a semantics for Milner's CCS. Unfortunately, one can show the event-structure denotations of Affine HOPLA are too impoverished to coincide with the standard "true concurrency" semantics of CCS (as e.g. given in [40]). A language must go beyond Affine HOPLA if it is to express such semantics (and the trace of nondeterministic dataflow—see below). Guidelines on what's lacking in Affine HOPLA can be got from work on presheaf models for concurrency [11], where the ingredients of product of presheaves, pomset augmentation and cartesian liftings (extending the match operators of Affine HOPLA) all play a critical role. This work suggests other event-structure representations, based on more general spans of event structures, specifically based on maps which support the augmentation of causal dependence, or are partial so allowing hidden events and leading to a treatment of weak bisimulation in the manner of [7]. A syntax for higher-order processes as event structures, supporting the range of parallel compositions on event structures, must lie in the algebra of the bicategories of spans.

Nondeterministic dataflow has stood apart from traditional domain theory because of a negative observation, the "Brock-Ackermann anomaly" [4], that its denotational semantics was not amenable to a relational treatment, nor consequently to an approach based on traditional powerdomains. Once "relations" are understood more broadly as maps in the affine category of presheaf models now representable as spans of event structures, the incompatibility with domain theory disappears [21].

Name generation and higher-order processes

Path-based models, either the domain models or presheaf models, can be extended to models of the Pi-Calculus [18, 10]. Processes are made to depend functorially on the set of current names. The domain model [18] supports a form of trace equivalence while the presheaf model [10] supports the traditional strong bisimulation of the Pi-Calculus. However, in extending this work to allow process passing there are fundamental questions regarding the existence of function spaces in suitable categories. The categories exist in domain and presheaf variants as well as stable domain theory analogues, so there is lots of room for experimentation.

Another direction in which to proceed (and where a good start has been made with Francesco Zappa Nardelli) is to make use of the informative presheaf model [10] as a guide in building a metalanguage, new-HOPLA supporting both name generation and higher-order processes. Its recursive types are often parameterised by a signature of actions, and this seems to provide fruitful links to action structures and bigraphs [22] and answer the challenge of how to handle locality without its being a primitive notion. Work has been begun on combining independence and name-generation [9].

An interesting question is whether the metalanguage supports bisimulation for ambients (that discovered by Merro and Zappa Nardelli [26]), and if not, this enterprise should inform us as to what's missing. Zappa Nardelli has begun a preliminary implementation in Fresh ML.

Beyond linearity

Code can be copied, and this generally leads to maps which are not linear. According to the discipline of linear logic, nonlinear maps from $\mathbf{P}$ to $\mathbf{Q}$ are introduced as linear maps from $!\mathbf{P}$ to $\mathbf{Q}$—the exponential ! applied to $\mathbf{P}$ allows arguments from $\mathbf{P}$ to be copied or discarded freely. Different ways to make assemblies of processes lead to different choices of exponential, the nonlinear maps of only some of which respect bisimulation [30]. There are several mathematical issues here. The exponentials are instances of what can claim to be "pseudo comonads" on a 2-category (or even bicategory) a rich and pivotal area which is under-developed. Nonlinear maps can be seen as generalised polynomials as well as generalised analytic functors [23] and there are unexplored connections with combinatorics.

We have not yet considered any but the simplest comonads as event-structure operations; the lifting comonad from which the affine categories are derived is easily seen as an operation on event structures. Given that copying involves a form of juxtaposition we expect that parallel compositions of event structures are involved. Quite possibly in representing the comonads one will be forced beyond event structures to a more general model.

The mathematical framework suggests we back-up from Affine HOPLA to a more neutral language based on linear maps with a comonad, though this has the disadvantage of taking us to less familiar territory as regards process languages—we know of no concurrent process interpretation of the linear tensor, for example.

Logic

Each presheaf category possesses a logic, generalising Hennessy-Milner logic, which is characteristic for open-map bisimulation (see [24]). In general the modalities are based on maps in the path category. Types denote path categories in languages such as Affine HOPLA. We can hope that by structural induction on types to obtain a workable syntax for the modalities and the logic at each type and include operations (a sort of weakest precondition) for moving assertions across from one type to another. This programme is similar in general outline to that of "logic of domains". A logic for Affine HOPLA is an obvious place to start.

The adjunctions associated with traditional powerdomain monads provide models of linear logic [19]. Presumably there are full abstraction results for HOPLA, companion to that in [31], based on detecting the "must" as well as "may" behaviour of processes. Just as there is an abstraction function from the presheaf semantics of HOPLA to its path semantics, so can we expect other abstraction functions from the presheaf semantics to other powerdomain semantics induced by the powerdomain logics.

Probability and interaction

Are the path-based models of processes extensible to probabilistic and continuous processes? The work on presheaf models suggests probabilistic analogues, in which the category of sets is replaced by a category built on the unit interval; nondeterministic processes as presheaves over paths are replaced by probabilistic processes as valuations on paths. This idea is in tune with Katoen's probabilistic event structures. Though it is not immediately clear from Katoen's definition [16], probabilistic event structures are essentially concrete data structures in which probability distributions are assigned to the values at cells; this results in a valuation where probabilities are assigned to configurations (read "paths"). We intend to keep an operational grip on matters through representations of valuations and maps e.g., in terms of probabilistic event structures. Through event-structure representations, one can expect to understand the relation between causal independence and conflict, on the one hand, and probabilistic independence, on the other. Significant preliminary work has been done here by Varacca and Winskel, Voelzer, and Abbes [36, 37], as yet unconnected to the very general Markov processes studied, e.g. in [32]. A language of new-name generation (often used as an idealisation from random generation) should be open to a probabilistic interpretation. How do we combine probability with nondeterminism? An obvious idea is to adjoin probability distributions to the sets of possibilities in the presheaf semantics of processes. Another is to form presheaves over probabilistic processes. Both options should be explored. Possibly related topological considerations appear in extensions of presheaf semantics to fairness [20].

Applications and techniques

Although this is primarily a theoretical project we intend the theory to feed forward through suggesting new process languages and analytical tools. We expect practical benefits, just as 'theoretical' work on unfolding of Petri nets and event structures [28] has become important in several tools and analysis techniques [2, 14, 15, 35, 42]. We are in communication with Benveniste's group, where ideas from independence models are being used in research on distributed fault management [2]. We are also finding common ground on networking with Nigel Walker of British Telecom. Having to perform certain calculations with categories routinely is leading to some streamlining in mathematical techniques, useful in a broad context [5, 6].

3 Detailed programme of research

The following plan is likely to evolve as research progresses. Funding is requested for a research associate (F. Zappa Nardelli) who will initially work on the domain theory of name generation, its metalanguage and operational semantics, and study the extent to which it subsumes existing process languages and their equivalences. In addition, funding for a PhD studentship is requested. Appropriate topics for PhD theses, drawn from the workplan, are:

- Event-structure semantics of higher-order processes;

- Name generation in path-based models of processes;
- Probability and interaction;
- Logic for higher-order processes.

All are rich areas for theoretical work and pilot studies into applications and implementations, quite naturally starting with the experimental application of HOPLA, affine HOPLA and new-HOPLA;

Below, we present an itemised breakdown of the research tasks and a diagram of their dependencies and time-plan in the "Diagrammatic workplan".

1. Event-structure semantics (with Hildebrandt, Curien, Nygaard)
   (a) Event-structure semantics of affine language Affine HOPLA (trace and branching variants); relation with presheaf semantics,
   (b) Event-structure semantics of nondeterministic dataflow,
   (c) Experiments towards reproducing existing independence semantics of process calculi (in particular the event-structure semantics of CCS) within domain theory,
   (d) Bistructures semantics (trace and branching variants); relation with presheaf semantics.
2. Operational semantics of presheaf models (with Nygaard and Baillot)
   (a) Operational semantics of first-order Affine HOPLA as a transition system with independence,
   (b) Operational semantics of full Affine HOPLA,
   (c) Develop an operational understanding of bisimulation from open maps at higher order,
   (d) Weak bisimulation from open maps—examples.
3. Nonlinearity (with Hyland)
   (a) 2-Categories and bicategories of non-linear processes,
   (b) "Pseudo comonads" on them,
   (c) Realising exponentials as event structure operations.
4. Name generation (with Zappa Nardelli)
   (a) A domain-inspired language and operational semantics for higher-order processes with name generation,
   (b) Encodings of existing calculi within the language,
   (c) Investigation of its model theory; the question of existence of function spaces in suitable functor categories,
   (d) Name generation and independence,
   (e) Its bisimulation operationally and denotationally.
5. Logic for presheaf models
   (a) Logic for each type of Affine HOPLA,
   (b) The translation of logic along maps between presheaf categories.
6. Probability and interaction (with Varacca)
   (a) Probabilistic processes and independence,
   (b) Probability and nondeterminism.
7. Experimental applications and techniques (in particular for new-HOPLA, Affine HOPLA, and a probabilistic language).

3.1 Relevance to beneficiaries

A domain theory for concurrency would suggest new (meta)languages for processes, concepts, constructs and reasoning methods related within a global framework. The strategy proposed here of obtaining process languages from suitable categories, equipped with an operational interpretation, would help systematise language design, operational semantics and process equivalences—issues of broad relevance. Presently, independence models (like event structures and Petri nets), though widely used, stand beyond the range of traditional domain theory and denotational semantics. Their incorporation in a domain theory has potential benefits for communities working on partial-order model checking, security protocols, nondeterministic dataflow, self-timed circuits, term-rewriting, game semantics, Petri nets and the analysis of distributed algorithms, as well as broadening the scope of these areas and their techniques. We are establishing collaboration with IRISA Rennes (via Albert Benveniste) and British Telecom (via Nigel Walker) on applications to telecommunications.

3.2 Dissemination and exploitation

Our work will be disseminated via selective international conferences and workshops and their proceedings, as well as refereed journals. HOPLA has already found its way into teaching material at Cambridge; its path-based domain theory and operational semantics form a model full-abstraction result. In the longer term a book is planned to follow up on the handbook chapter [40].

3.3 Justification of resources

The total cost of the project will be 252,669 pounds, to be spent over 3 years. Given the ambition and scope of the project this is good value for money. Joint collaboration with researchers (Baillot, Curien, Nygaard, Varacca, Hildebrandt) is already planned.

Staff Funding is requested for one full-time Research Associate (Zappa Nardelli) for three years. One project studentships is requested. Contribution to the salary of a Computer Officer is requested for the additional installation, maintenance and support required for print, file access and authentication via Mac's.

Travel Support is requested for conference/research trips within the UK (the research trips to visit Abramsky, Hennessy, Jung, Plotkin, Power, Rathke, Robinson, Sassone, Stark), conference/research trips in Europe (the research trips to visit groups around Benveniste, Curien, Hildebrandt, Nielsen) and outside Europe (the research trips to visit groups around Brookes, Joyal, Panangaden, Zhang).

Equipment Funding for one notebook computer for use by project members is requested for off-site working and presentations. Funding is requested for one workstation and one notebook computer for the RA and PhD student.

References

[1] Abramsky, S., Game semantics for programming languages. In proc. MFCS, LNCS 1997.

[2] Benveniste, A., Haar, S., Fabre, E., Jard, C., Distributed monitoring of concurrent and asynchronous systems. CONCUR'03, LNCS 2761, 2003.

[3] Berry, G., Modèles complètement adéquats et stables des lambda-calculs typés. Thèse de Doctorat d'Etat, Université de Paris VII, 1979.

[4] Brock, D., and Ackermann, W., Scenarios: a model of non-determinate computation. ICFP, LNCS 107, 1981.

[5] Caccamo, M., and Winskel, G., A higher-order calculus for categories. TPHOLs 2001, LNCS 2152, 2001.

[6] Caccamo, M., Hyland, M., and Winskel, G., Notes on Category Theory. From www.brics.dk/~ mcaccamo.

[7] Cattani,G.L, Fiore, M., and Winskel, G., Weak bisimulation and open maps. LICS'99, 1999.

[8] Cattani, G.L, and Winskel, G., Profunctors, open maps and bisimulation. Available from (www.cl.cam.ac.uk/~ gw104).

[9] Cattani, G.L, and Sewell, P., Models for name-passing processes: interleaving and causal. LICS, 2000.

[10] Cattani, G.-L., Stark, I., and Winskel, G., A presheaf model for the Pi-calculus. CTCS, LNCS 1290 1997.

[11] Cattani, G.-L., and Winskel, G., Presheaf models for concurrency. In *Proc. of CSL' 96*, LNCS 1258, 1997.

[12] Curien, P.-L., Plotkin, G., and Winskel, G., Bistructures, bidomains and linear logic. In the festschrift celebrating Robin Milner's 60'th birthday.

[13] Dennis, J.B., Control flow and data flow: concepts of distributed programming. Vol. 14 of NATO ASI Series F, Springer, 1984.

[14] DIMACS Workshop on Partial Order Methods In Verification, Princeton, July 1996. In DIMACS Series in Discrete Mathematiocs and TCS, 29, AMS, 1997.

[15] Eighth International Symposium on Asynchronous Circuits and Systems, UK, IEEE, 2002.

[16] Katoen, J-P., Quantitative and qualitative extensions of event structures. PhD thesis, Twente, 1996.

[17] Hennessy, M., A fully abstract denotational model for higher-order processes. Inf. and Comp., 112, no.1, 1994.

[18] Hennessy, M., A fully abstract denotational semantics for the pi-calculus. TCS vol.278, 2002.

[19] Hennessy, M., Plotkin, G., Full abstraction for a simple parallel process language. MFCS, LNCS 74, 1979.

[20] Hildebrandt, T., Towards categorical models for fairness: fully abstract presheaf semantics of SCCS with finite delay. CTCS'99, LNCS, 1999.

[21] Hildebrandt, T., Panangaden, P., Winskel, G., Relational semantics of nondeterministic dataflow. CONCUR'98, LNCS, 1998.

[22] Jensen, O.H. and Milner, R., Bigraphs and mobile processes. (www.cl.cam.ac.uk/~ rm135/), 2003.

[23] Joyal, A., Foncteurs analytiques et espèces de structures. Springer Lecture Notes in Math. 1234, 1986.

[24] Joyal, A., Nielsen, M., and Winskel, G., Bisimulation from open maps. Inf. and Comp., vol. 127, no.2, 1996.

[25] Lamport, L., Time, clocks, and the ordering of events in a distributed system. CACM 21(7), 1978.

[26] Merro, M., and ZappaNardelli, F., Bisimulation proof methods for mobile ambients. ICALP 2003.

[27] Milner, R., Communicating and mobile systems: the pi-calculus. Cambridge University Press, 1999.

[28] Nielsen, M., Plotkin, G., Winskel, G., Petri nets, event structures and domains. TCS, vol. 13, 1981.

[29] Nygaard, M., Domain Theory for Concurrency. PhD thesis (www.brics.dk/~ nygaard), Nov. 2003.

[30] Nygaard, M., and Winskel, G., Linearity in process languages. LICS'02, 2002.

[31] Nygaard, M. and Winskel, G., Full Abstraction for HOPLA. CONCUR'03, LNCS, 2003.

[32] Danos, V., Desharnais, J., Panangaden, P., Conditional expectation and the approximation of labelled Markov processes. CONCUR 2003.

[33] Plotkin, G.D.P., A powerdomain construction. SIAM Journal on Computing 5(3), 1976.

[34] Sangiorgi, D., and Walker, D., The pi-calculus: A Theory of Mobile Processes, CUP, 2001.

[35] Thayer, J., Herzog, J., and Guttman, J., Strand spaces: why is a security protocol correct? IEEE Symposium on Security Protocols, 1998.

[36] Varacca, D., Two Denotational Models for Probabilistic Computation. BRICS PhD thesis (www.brics.dk/~ varacca/), Nov. 2003.

[37] Voelzer, H., Randomized non-sequential processes. CONCUR'01, LNCS, 2001.

[38] Winskel, G., Events in Computation. PhD thesis, University of Edinburgh, 1980. Available from www.cl.cam.ac.uk/~ gw104.

[39] Winskel, G., A presheaf semantics of value-passing processes. CONCUR'96, LNCS, 1996.

[40] Winskel, G. and Nielsen, M., Models for concurrency. Foundations of Computer Science IV, OUP, 1995.

[41] Winskel, G., Linearity and nonlinearity in distributed computation. Invited chapter in Book on Linear Logic. See www.cl.cam.ac.uk/~ gw104.

[42] Winskel, G., and Crazzolara, F., Events in security protocols. 8th ACM Conf. on Computer and Communication Security, 2001.

[43] Zappa Nardelli, F., De la sémantique des processus d'ordre supérieure. PhD Thesis, Paris. Dec.2003.

Diagrammatic workplan