

Process Algebra as Modelling

Chris Tofts

Hewlett-Packard Laboratories Bristol

Abstract

After 25 years of research (19 personally) into process algebras, I ask what areas of mathematics other than the analysis of concurrent computation could or indeed should its basic approach be applied? In particular, I identify the two areas of reductionism and model comprehension, upon which I believe that process algebra has the potential to have a major impact.

1 Introduction

It is a very dull piece of mathematics which solely addresses the problem it was originally designed for. After 25 years of research effort on process algebra, it is interesting to ask the question ‘what other problems (can) does it solve?’ If we were to assume that an alternative solution to the problem of verifying interacting concurrent systems provided a complete and elegant solution to those issues would there still be a role for process algebra - or should the activity be mothballed?

My personal belief is that the underlying philosophy of process algebra gives us insight into two long standing difficult areas:

- comprehension through composition;
- representing mildly heterogeneous interacting systems for Markov chain analysis.

The first item can be summarised as the ‘reductionism vs holism’ debate. In its simplest terms the proponents of each side of this debate adopt positions of: ‘all systems can be understood in terms of their components’, and ‘some systems can only be understood as wholes’, respectively. Since (almost arbitrary) composition is the *sine qua non* of process algebra, if it cannot add to this debate then it is probably deficient in its fundamental construction. The second problem concerns those systems that are often described as being ‘complex’ or having ‘emergent’ properties. Complexity in these systems does not arise as a consequence of a very large number of interacting components, for if the numbers were that large then statistical physics techniques would apply. Nor are these problems the result of truly vast state spaces, since again the techniques of continuous probabilistic mathematics could be exploited. These problems lie in the *middle*, where there are either a middle sized number (10-1000)² of states in the components or a middle sized number of different components (4-100) and they are present in middle sized quantities (5-1000). It is in these settings that ‘traditional’ applied mathematical techniques tend to suffer.

¹ Email: chris.tofts@hp.com

² These numbers are actually the grossest approximations and are meant to be purely indicative.

2 Reductionism vs Holism

The main confusion in this area seems to lie between the methods of understanding the behaviour of systems and the methods of calculating the behaviour of systems. The fundamental problem being that the two activities should be supported by the same mathematics. As a primary counter example, the behaviour of the transistor was predicted using quantum mechanics, however this is not the approach currently taken to calculating the behaviour of circuits.

If we accept that the physical universe around us is formed from parts and it is the peculiar interaction of those parts which gives rise to the phenomena we see, then an appropriate mathematics would have to have the following properties:

- representation of ‘parts’ with arbitrary behaviour;
- representation of arbitrary interaction mechanisms between the ‘parts’.

The family of approaches that is termed process algebras would at first sight be general enough to cope with both of these requirements. More interestingly, is the application of Robert de Simone’s result on synchronous process algebras in this area. If we can assume a bound on the descriptive complexity of the interaction between parts, say recursively enumerability, then we may be able to give a complete account, at least of the computational aspects, just using calculi of the complexity of Meije or SCCS.

It is important to recognise what this may achieve. We may well be able to give a ‘true’ account of the components and their interactions which give rise to complex physical phenomena. However, this does not imply that we shall be able to calculate effectively with such representations. Finally, as a consequence of the free semantic interpretation of the action within process algebra, we can choose the level of abstraction (and consequent calculational difficulty) which our models will entail. In principle within the same calculi with the appropriate understanding of process abstraction we may be able to give a formal account of the relationships between the necessary abstraction levels. Process algebra may permit us to demonstrate that the whole is never greater than the sum of the parts, if you do the summation correctly.

3 Heterogenous interacting systems

To simplify the following discussion I will use the ‘lazy’ nomenclature of complex to describe the class of systems under discussion. In this setting there are many practical problems for the modeller:

- convincing themselves that they have captured the problem correctly;
- interpreting the results from the analysis;
- presenting the results to those who need them.

Most practical modelling exercises take place in order to support decision making, even in the most abstract academic setting this is often a question as to whether we pursue one line of research or another (or even build one particle accelerator design or another). Given the cost of making incorrect decisions there is a common belief that the models which support these decisions need to be suitably complex and detailed. Since the primary skill of an applied mathematician is the removal of unnecessary complexity and detail this leads almost immediately to a conflict between the model constructor and the model user (or decision maker). So called ‘individual based models’ have become increasingly popular as they appear to provide a solution to the comprehension problem since the fundamental entities in the model are individuals whose behaviour all of the parties to the modelling exercise can agree on. However, this approach to modelling largely engenders a position where the only form of analysis possible is to study the system through simulation.

From a process algebraic perspective we have two advantages:

- we can ‘validate’ the individual components;
- we can track the contribution of component state on the whole system.

Both of these properties permit us a more convincing account as to why the model is ‘correct’ and how to comprehend its predictions. Even if we are forced to resort to simulation in order to analyse systems then the process algebraic view can add to our understanding. For the object oriented class of discrete event simulation languages, which can be viewed as starting from Nygaard and Dahl’s *Simula*, process algebras can provide an elegant and effective semantic basis. Indeed, starting from Birtwistle’s simulation oriented abstraction of *Simula*, *Demos*, it is possible to derive a language which is expressively identical to a value passing version of *SCCS*, whilst retaining the ease of modelling (programming) of the original system. In many cases this permits us to exploit human guided abstraction and comparison through execution until we reach the point where the model can be resolved by analytic techniques, as implemented in *HOLoS*.

Simply viewing process algebra as a notation for complicated Markov chain systems has benefits. One problem with Markov chains is identifying the state space. In any non-trivial context this can be highly error prone. By forming the state space as a composition on interacting processes, this construction is essentially automatic. Indeed, the underlying algebra can be exploited to reduce the state space, via identity, on the fly. More importantly when particular structures are identified within the state space of a large Markov chain it can be difficult to relate this back to the originating system. When the state space is constructed from a process algebraic source, and consequently enumerated by the state names of the components, it is possible to recover all of the underlying component states and thus interpret the behaviour in terms of the original system. This approach has proved extremely powerful in modelling of animal behaviour and evolutionary systems.

4 Conclusions and Problems

Process algebra has significant potential to influence modelling activities outwith theoretical computer science. The two areas I identified above are simply the most immediate ones from my own experience and prejudices. I wonder how the development of process algebra would have proceeded if a view similar to Nygaard’s on *Simula*:

Simula is a problem description language, which with the addition of input and output statements could be executed on a digital computer.

had been applied to the development of process algebra. Combining these views with the (essentially) automata based probabilistic calculational approaches of Marcel Neuts in the mid 1980s may have lead to the primary applications of process algebra being substantively different to the current ones.

It is traditional in time based reviews of a subject area to end with some problems which one considers of sufficient interest to mention and the following are ones I feel are the most interesting:

- (i) Does the physical universe exploit non RE describable combinators?
- (ii) Does the de Simone result hold in synchronous calculi extended with physical phenomena?
- (iii) Is the limiting approximation to asynchrony valid in this space?
- (iv) What is the appropriate notion of abstraction to capture the activities of applied mathematicians?

- (v) Can we exploit the information bound of finite structural state systems mutually observing?
- (vi) What is the relationship between probabilistic process algebra and the effective calculational techniques of Marcel Neuts?
- (vii) What is the appropriate notion of abstraction within probabilistic settings?
- (viii) How do we connect process algebra with dynamical systems theory?
- (ix) Why do theoretical computer scientists concentrate on detail complexity?
- (x) Why is the function the dominant mode of system understanding?