SOCIOTECHNICAL ALIGNMENT IN THE INTRA-ORGANISATIONAL DIFFUSION OF INFORMATION TECHNOLOGY

ALFONSO H. MOLINA
TechMaPP,
The University of Edinburgh
Department of Business Studies
50 George Square
Edinburgh EH8 9JY
Scotland, U.K.

Abstract

Generally, 'diffusing' technologies have to gain a legitimate space, often in hard competition for recognition and the resources which will make them part of the overall operation of a company. This raises critical questions such as: how is it that IT technologies become (or not) established as valued ingredients in the life of a company? what factors are likely to influence their acceptance and hence, successful diffusion? is it merely a matter of technical and commercial merit? and what is the role played by the nature of the technology itself?

This paper addresses these particular and general questions by looking in detail at the emergence of a new software technology (i.e., formal-methods) at the British semiconductor company Inmos. The concepts of sociotechnical alignment and 'diamond of alignment' are developed as part of a general theoretical discussion on the nature of intra-organisational diffusion of technology.

Keywords: *technology diffusion / implementation / information technology / sociotechnical constituences / Inmos*

Introduction

The diffusion or implementation of information technology (IT) within a company environment is unlikely to be a simple, linear process involving an unproblematic movement of technology from supplier to recipient. This is well accepted.[1] Generally, 'diffusing' technologies have to gain a legitimate space, often in hard competition for the recognition and resources which will make them part of the overall operation of a company. This raises critical questions such as: how is it that information technologies

---

[1] See below discussion on Diffusion, Implementation, and Sociotechnical Alignment.

A. Inzelt and R. Coenen (eds.), Knowledge, Technology Transfer and Foresight, 125–148.

become (or not) established as valued ingredients in the life of a company? what factors are likely to influence their acceptance and hence, successful diffusion? is it merely a matter of technical and commercial merit? and what is the role played by the nature of the technology itself?

This paper addresses these particular and general questions. It proposes a conceptual framework and applies it to the case-study of the diffusion of a new software technology (i.e., formal-methods) within the British semiconductor company, Inmos. The case is particularly revealing because, in 1990, the prestigious British Queen's Award for Technological Achievement was given jointly to Inmos and Oxford University Computing Laboratory for their innovative development and use of formal methods on the verification of the design of Inmos' microprocessor: the transputer [43]. The award seemed to sanction the definitive diffusion of formal methods within the Inmos' design process.[2] As we shall see, however, the reality behind the news was more problematic and the 'successful' use of formal methods did not really amount to an assured place for the technology within the company's design process. The paper analyses the experience and draws lessons with a view to enriching our understanding of the nature of intra-company diffusion and implementation of information technology.

The next section provides a brief overview of Inmos, formal methods and its relation to the design process of microprocessors. The third section presents a theoretical background and characterises the nature of the processes involved in the intra-organisational diffusion of technology. The discussion pays particular attention to the nature of formal methods technology as well as to the role of social behavioural aspects. The fourth section deals with the case study and the final section discusses the findings of the paper.

1. Formal Methods and Inmos

Formal methods is one of those technologies with no single clear definition. It has been suggested that 'as a whole, [they] are concerned with 'proving' by means of detailed logical steps, that a function or component is demonstrably correct with respect to its specification.' [51, p.248] In addition, although some practitioners do not necessarily equate 'formal' with 'mathematically-based', it is a fact that mathematics has a central role in the technology. Thus, other authors define formal methods squarely as 'the use of mathematics for specifying the properties of a system and the use of mathematical proof for validation.' [12, p.26] As we shall see, until recently, the industrial use of formal methods has been more the province of military and other highly safety-critical

[2] The news also seemed to fit well the pattern of 'user-dominated' innovations described by von Hippel in relation to scientific instruments and certain process innovations [52,53]. Specifically, he found that all the innovations he studied which did not require innovative hardware were 'user-dominated' [52, p. 69]. Fittingly, at Inmos, formal methods were developed for, and used in, the company's design process making use of standard commercial hardware. A major difference, however, is that the Inmos' innovation has not yet evolved out of the company and onto the market (in fact this remains an open question). The diffusion of the technology is thus still at the intra-organisational phase of the overall process described by von Hippel.

applications. This means that they have yet to see a widespread implementation in high-volume industrial environments. In this respect, Inmos is very much pioneering the way, something that has been a characteristic of the company from its birth.

Inmos was established in 1978 by the UK Labour Government with a view to encouraging the development of a British microelectronics capability [33, 34]. The company's strategy was to tackle the market with innovative, high-performance products such as SRAMs (static random access memories) and transputers. In particular, the transputer (the word comes from transistor-computer) was to be the flagship of the British microprocessor industry. It was the first microprocessor in the market with in-built generality of purpose and massively parallel processing capability.

The first transputer was launched in 1985 under the codename T414. By now, there are two additional generations of transputers: the T800 launched in 1987, and the T9000 in 1993. As indicated, the main concern of this paper is with the second generation T800 transputer. A short description of this microprocessor points out that the device is a complete computer on a silicon chip about 1 sq. cm. It combines a 32-bit central processor together with 4 Kbytes of RAM (random access memory) and communication links plus an arithmetic unit (floating point processor) which operates in parallel with and under the control of the central processor. In addition, the transputer is a RISC-like (reduced instruction set computer) device, meaning that it operates only short and simple instructions which can be executed very fast - typically in 50 to 100 nanoseconds. The transputer can be used both as a building block for parallel processing machines and as a single very-high performance microprocessor for computers and embedded applications [34]. The third generation transputer, the T9000, offers substantial increases in performance over the T800. It has 3.3 million transistors with a peak performance of 200 MIPS (millions of instruction per second). This is 10 times the performance of the T800 which had only 300,000 transistors. By 1992, over 750,000 transputers had been sold worldwide. The language of the transputer is Occam which is not only a programming language for implementing concurrent processing systems but also a software design methodology which completely specifies the transputer. In fact Occam predates the transputer in that it was used to design the chip; conversely, the transputer is the best component for implementing Occam as a programming language. The roots of Occam are found in Tony Hoare's work on CSP (Communicating Sequential Processes) at Oxford University. Basically CSP allows for the communication of asynchronously operating program fragments. Occam was developed by D. May of Inmos explicitly to facilitate parallel processing. The name of the language reflects its philosophy which follows that of the 13th century philosopher William of Occam who stated the principle known as Occam's Razor: 'one must not multiply entities without necessity'. [34]

Regarding formal methods, the first generation T414 did not make use of this technology in its design process. It was only with the inclusion of the on-chip floating point unit (FPU) in the second generation T800 that formal methods really entered the picture. The incorporation of an on-chip floating point unit was a response to a market demand from the scientific and engineering community. Previously, the T414 hardware was only capable of performing integer operations and this was a real limitation in the high-performance computer market addressed by the transputer. Figure 1 shows that a T800 is basically a T414 plus a hardware FPU. It also shows that formal methods were used in the design of only part of the new FPU, namely, the microcode.[3]

Nevertheless, it was probably the first time that a microprocessor company anywhere in the world had made use of this emerging technology in the design process of a product for high-volume markets. The results could hardly appear any better. The technology had 'proven' to be an important tool in the design of the T800 and the Award symbolised what looked like their definite diffusion within the Inmos' design process. But, what does the experience tell us?

2. Diffusion, Implementation, and Sociotechnical Alignment

At the beginning, the paper raised a number of questions, among them, how is it that emerging technologies become (or not) established as ingredients in the life of a company? and what role is played by the particular nature of the technology? This section examines a number of theoretical concepts relevant to these questions and to our general understanding of the process of intra-company diffusion or implementation of information technology.

2.1. TECHNOLOGY DIFFUSION AND IMPLEMENTATION

A great deal of the literature on technology diffusion has concentrated on the success or failure of products across inter-organisational and, often, market-mediated interactions, transactions, transfers. Numerous success and failure factors have been identified by the many studies [8, 20, 25, 27, 44, 45, 54] and various lists and tables can be found abstracting and grouping relevant features, requirements, patterns of development (trajectories), as well as limits and opportunities for managing technological innovation. The focus has mostly been on the overall competitive performance of firms and a variety of concepts have been proposed, including Freeman's taxonomy of firm strategies [18]. Complementary taxonomic work has also characterised the type and extent of impact of the diffusion of product and production process innovations. Particularly relevant are the concepts of incremental innovations [19, 42], and Abernathy and Clark's "transilience map" containing architectural, niche, revolutionary, and regular innovations [1].

The term implementation rather than diffusion has been used for cases of intra-organisational adoption, development and use of technologies [16, 17, 23, 24, 54, 55]. For Leonard-Barton, for instance, the implementation of new production technologies means: 'getting them up and running in daily operations.' [24, p.251] In Fleck's (1993) words, implementation is 'the process of getting technologies to work, especially complex ones, as commercially successful operating systems.' [17, p.637]

[3] 'The microcode includes intricate algorithms for floating point multiplication and division, and handles a large number of special cases resulting from denormalized numbers and infinities.' [31, p. 4] The work on the T800 FPU has been described in [3, 31, 32, 48]

Implementation underlines the fact that initial acquisition or transfer of technology does not necessarily imply appropriation and effective usage. The process is rather one of mutual adaptation and development between 'incoming' technology and the user organisation or environment. Indeed, to the extent that implemented technologies are combinations of internally generated and externally resourced subcomponents, the term 'incoming' technology should be understood, more precisely, as 'incoming' technical base or components - that is technology open to further, sometimes significant, development during implementation.[4]

An implicit assumption in the implementation problem is that the implementing party is introducing technology which initially has not been developed by themselves, but by another source who has not fully appropriated or crystallised the contingencies of the user-environment into the technology. This would be especially the case when the developers' initial technology design has taken place with little regard for the sociotechnical reality 'alive' in the user environment. But it could still be the case even when developers belong to the same organisation and have tried to reduce uncertainty by technical iteration and prototyping. The reason is that between a prototype and a routinely operational technology there is always a mismatch or an adaptation gap involving the technology and the user-organisation. Leonard-Barton identifies three critical mis-alignments (poor fits) threatening the success of an implementation. These are between 'the technology and (a) technical requirements, (b) the system through which the technology is delivered to users, or (c) user organisation performance criteria.' [24, p. 252] These mis-alignments turn implementation into a complex process in which further innovation, sometimes major innovations, are commonly required. This is the process of 'innofusion' identified by Fleck [15][5], as well as Leonard-Barton's extension to the invention process [24].

Different technologies are associated with different degrees of implementation difficulties. Immature technologies, for instance, have been less tried and are likely to be more unpredictable in their interaction with the organisation. Configurational technologies, comprising assemblies of technological and non-technological components built up to meet local contingencies, are complex and 'demand substantial user input and effort if they are to be at all successful, and such inputs can provide the raw material for significant innovation.' [17, pp.637-8]

---

[4] I am grateful to J. Fleck for drawing my attention to this point.

[5] Innofusion is a mode of evolutionary innovation in which the processes of innovation and diffusion are collapsed together, and in which significant development takes place during implementation, within the user organisation [16].

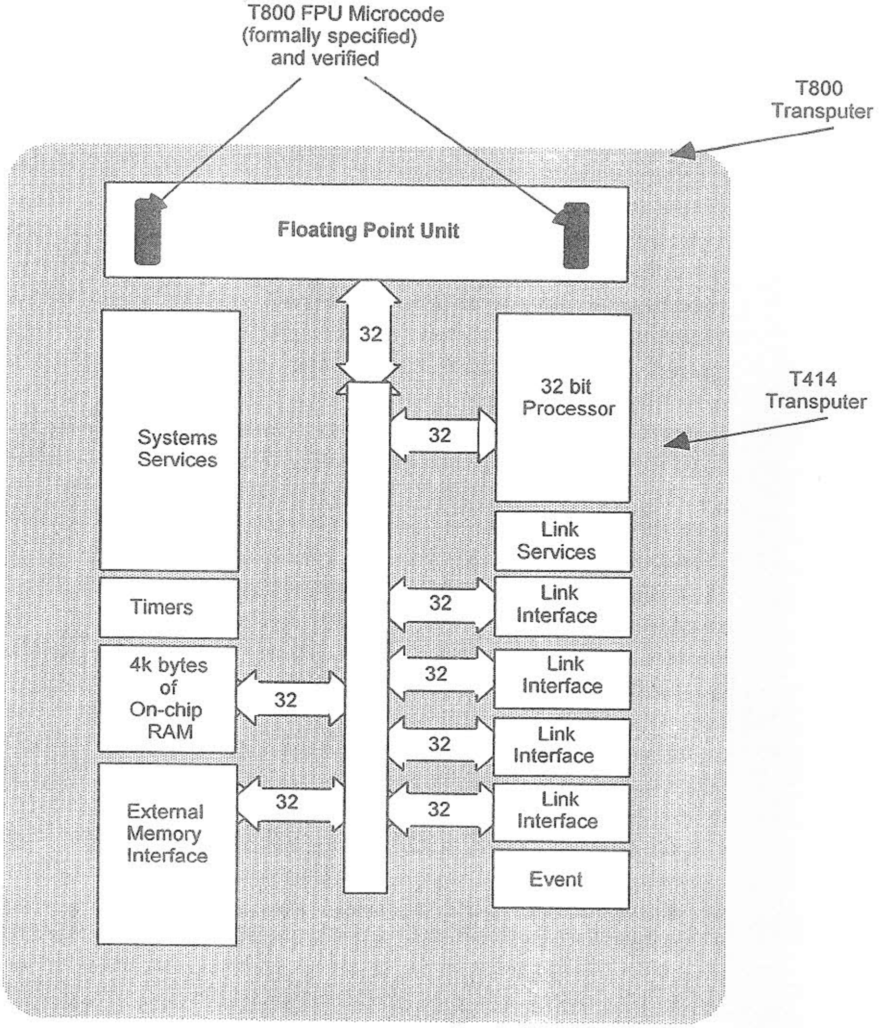

Figure 1: T800 Transputer and Formal Methods

In configurational technologies the processes of innovation and diffusion are collapsed together. Adoption of an integrative approach to innovation is therefore a *necessity* [23]. This would require simultaneous attention to involving users as codevelopers; creating a support system - including a network of supporters and an adequate delivery system for users; and experimentation and planned learning about the integration of the new technology. The type of innovation represented by the technology being implemented is also important. It draws attention to the relationship between the 'incoming' and a possibly established technology [46]. A 'revolutionary innovation', for instance, disrupts and renders established technical and production competence obsolete. Potentially, this means high-level of conflict as 'adaptation' would go hand-in-hand with antagonistic competition, displacement and destruction. In other cases, the relation between the 'incoming' and established technologies may be one of potential complementarity, for instance, adding up to each other in shortening product-development time. The issue here will be much more one of accommodation of roles and rewards, expansion of the skill-base and mutual learning, as well as an all-interacting-parties' recognition of effective and substantial contribution to a common objective.

As we shall see, the company development and use of formal methods concerning this paper neither match the features of configurational nor incremental or revolutionary type of innovations/implementations. The reported successful implementation seems to have happened without much involvement by the target users of the technology. Nor there seems to have been a situation of real displacement of an established technology by the 'incoming' formal methods. Indeed, if anything, the role of formal methods in verification and validation of microprocessor design appears to be a welcome addition, given both the extremely damaging impact of flaws in microprocessor design[6] and the absence of a single way to prove microprocessors right before they reach the market. On the other hand, formal methods fits the characterisation of new immature technology which is likely to be more unpredictable in its interaction with the organisation [24]. Particularly this means that the requirement for *a process of accommodation of roles and rewards and an all-interacting-parties' recognition of effective and substantial contribution to a common objective remains critical.* Successful implementation is thus never to be taken for granted nor, indeed, as definitive - as formal-methods practitioners found out to their surprise.

This contention brings more into relief the negotiation and behavioural aspects of the process of innovation/implementation. Negotiability in the development of technology has been more the province of sociological approaches such as actor-networks [22] the social construction of technology [4] and the social shaping of technology [26]. Similar emphasis on the political nature of technological processes is found in Badham [2], in the labour process literature [5, 41] and in the processual approach [9]. In turn, the importance of behavioural factors is more the province of social psychology and, particularly, the seminal work on behavioural theory of the firm by scholars such as Simon, Cyert and March [13, 28, 29, 49, 50] Specifically, Cyert and March argued that the organisation should be understood as a 'coalition of individuals, some of them organised into subcoalitions'. [13, p.27] Underlying this view is the premise that 'almost all human behavior consists of sequences of goal-oriented actions.' [50, p.19] They rejected the idea of a single, consistent set of goals

[6] The recent flaw found in the Pentium has forced Intel to offer cutomers the replacement of the chip. Since more than 4 million of the flawed chips are already out, it is estimated that the cost to the company could be as high as $500 million [6, 14].

which members of the coalition share and pursue. 'Basic to the idea of a coalition is the expectation that the individual participants in the organisation may have substantially different orderings (i.e. individual goals).' (ibid.) Conflict is thus of the essence of the organisational process, for, whenever an individual or group experiences difficulty in selecting an action alternative, then there is a conflict [29]. According to Nelson and Winter (1982), however, some organisations clearly develop routinised patterns of behaviour which amount to a comprehensive truce in intraorganisational conflict and tend to stabilise their development [40]. The process is also self-reinforcing because 'fear of breaking the truce is, in general, a powerful force tending to hold organisations on the path of relatively inflexible routine.' [40, pp.110 and 112]

Of course, a major aspect of a new 'incoming' technology is its potential to alter the existing 'truce' by seeking to become part of contexts which already have the presence of other technologies. In these circumstances, there might be plenty of room for misalignments and conflict, even for potentially complementary technologies, and the key factor for successful implementation becomes the quality and effectiveness of what has been referred to as the *process of sociotechnical alignment* [37, 39].

2.2. SOCIOTECHNICAL CONSTITUENCIES AND ALIGNMENT

The concept of sociotechnical alignment is part of the *sociotechnical constituencies* approach [35, 36, 37,38]. This approach propounds that the processes of innovation, implementation and, generally, generation of technological capabilities always entail the build up of sociotechnical constituencies (STCs). *STCs are defined as dynamic ensembles of technical constituents (e.g. machines, instruments) and social constituents (e.g. institutions, interest groups) which interact and shape each other in the course of the creation, production and diffusion of specific technologies such as formal methods.* From this perspective, the emergence of formal methods within Inmos can be treated as an intra-institutional construction of a formal-methods constituency through a process of sociotechnical alignment.

Sociotechnical alignment is what constituents try to do (however consciously, successfully, partially or imperfectly) when they are promoting the development of a specific technology either intra-organisationally, inter-organisationally, or even as an industrial standard. It may be seen as *the process of creation, adoption, accommodation (adaptation) and close or loose interaction (interrelation) of technical and social factors and actors which underlies the emergence and development of an identifiable constituency*. As such alignment should neither be seen as a mere jigsaw-like accommodation of static available pieces nor as complete and permanent, once achieved. For this reason, the term 'alignment' is well supplemented by those of 'mis-alignment' and 're-alignment' which express, on the one hand, situations of tension and dis-harmony and, on the other, changes or re-accommodations in the life of a constituency. Non-alignment may preferably be used for situations in which the parties have not come to each others' attention and is thus less proper to talk of tensions or conflict. Also, alignment between people should not be reduced to consensus. The latter is one possible form of alignment but there might also be 'authoritarian' forms in which alignment is enforced by one party over another through sheer use of power.

In alignment the flow of influences is multi-directional. Indeed, as a sociotechnical process, the interrelations involved are not only among people and institutions but, simultaneously, among people/institutions and technical elements. Thus, when two or more people come together to pursue a common goal then we may talk of alignment between people; when developers shape technologies in accordance with potential users' specifications then we may talk of aligning the technology to people; when people have to learn new skills to be able to use a technology then we may talk of aligning people to technology; lastly, when technologies are shaped in accordance with the features of other technologies then we talk of aligning technology to technology; in practice, all these elements are likely to be present in the development of a constituency at one time or another. Indeed, mis-alignment and re-alignment must also be included since practical constituency-building strategies may sometimes follow these alternatives, for instance, to generate space within an organisation having a strong competing constituency. The central task of a rich strategy is to identify these alternatives and ensure that the most appropriate combinations, emphases and changes are implemented during the life of a constituency. In this respect, it is important to know that the four types of alignment just mentioned are not necessarily complementary. Sometimes alignments in certain directions may actually produce mis-alignment in others. For instance, it is not uncommon for people to reach agreement (alignment) on the basis of expectations (specifications) which then prove mis-aligned with what was actually feasible, given both resources and state-of-development of the technology. This kind of mis-alignment may turn out to be very costly for a constituency and I suspect that it is very much behind the 'over-spending' often bedevilling technical projects.

In previous work, Molina has used the concept of 'diamond of alignment' to illustrate the multiple dimensions of alignment required for successful constituency-building in large-scale inter-institutional initiatives [39]. The emergence of formal methods at Inmos, provides an opportunity to adapt the 'diamond of alignment' to an intra-organisational (company) dimension. This is illustrated in Figure 2 which contains the various dimensions constituency-building strategies might seek to integrate in order to enhance the chances of successful implementation. Leonard-Barton's mis-alignments are pertinent here and they are largely incorporated in the different aspects of the 'intra-organisational diamond'.

At the centre of the diamond is the evolving technology of the constituency. At all times, specific products, solutions and applications are not separate from the constituency. Rather they must be seen as evolving technical manifestations crystallising the state of development of the constituency. In principle, the better the initial alignment of the incoming technology with the various dimensions of the 'diamond', the more effective and easier the implementation is likely to be. A situation of minimal conflict would at least show:

Critically related to this strategic alignment is, second, a clear alignment between the capabilities of the constituency and the technical demands intrinsic to the nature of the problem. In short, the constituency must have the technical capacity to deliver the goods within the resources available at any given time. Capacity and resources, however, are dynamic, mutually influencing, factors and part of the capacity to deliver may imply an ability to expand the resources available. An alignment with well-established industrial standards and trends is often an important factor in this dimension.

Third, a clear alignment between the constituency's developers and the potential or target users and other relevant parties in the process. The aim is for all of them to become members, even developers, of the constituency. This implies various possible directions of alignment, most likely in combinations; (a) expertise-based alignment between developers and users; (b) alignment of the technology to users; and (c) alignment of users to the technology.

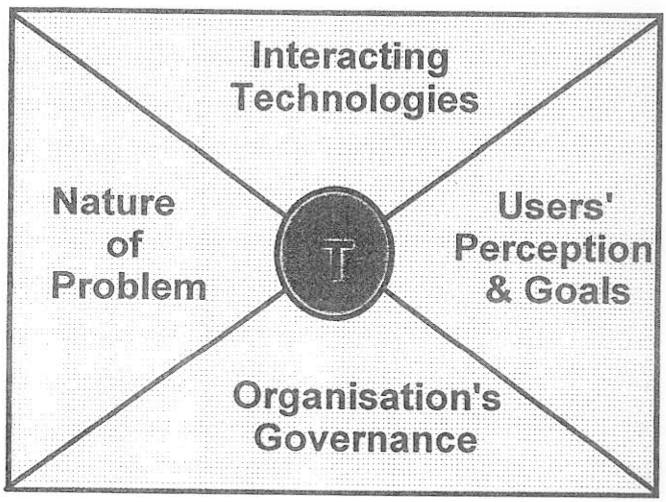

Figure 2: The Diamond of Intra-Organisational Alignment

Fourth, alignment of the constituency's technology with other interacting technologies (hence constituencies) in the creation, production and diffusion system of the firm. This implies a number of possible situations such us:

- *obligatory complementarity* in which the 'incoming' technology requires of others to realise its contribution. In this case, specific solutions will demand strong expertise-based alignment, giving rise to a process akin to what Collinson has referred to as 'knowledge-integration' [10];
- *non-obligatory complementarity* in which the 'incoming' and other technologies may contribute to a common organisational purpose but their interaction is not a pre-condition for one or the other to work;
- *antagonistic competitive* in which the 'incoming' and other technology are disputing the same functional role in the organisation and acceptance of one implies total displacement of the other. The essence of this case is high-degree of conflict;
- *non-antagonistic competitive* in which the 'incoming' and other technology are addressing similar functional roles in the organisation but they can or are allowed to co-exist and compete;

In all cases, the successful implementation of the 'incoming' technology will demand Leonard-Barton's adaptation between the technology and the organisation. The process of constituency-building and sociotechnical alignment implied in such adaptation, however, will vary greatly in terms of complexity and conflicts depending on the nature of the technology and its initial relationship to the various dimensions in the 'diamond' of alignment. In this process, expertise-based mis-alignment certainly plays a most critical part. On its own, however, it does not suffice because sociotechnical alignment is not only about expertise, it is also about other dimensions of human behaviour including uncertainty and elements such as suspicion, fear, resentment, and others which may sometimes have significant effects on the course of constituency-building experiences. True, in technological processes, elements such as fear, resentment, etc. are often inseparable from expertise-based mis-alignment. They are its manifestation. But these elements may also be associated to other sources and, perhaps, themselves underlie what may look like an expertise-based mis-alignment - apart from the fact that people with similar expertise may react quite differently when faced with similar situations.

At any rate, the key point is that, for instance with newly emerging technologies, it is often the case that different organisational players are highly uncertain about what they want or can expect from them, or simply, different players such as developers and users may have problems in understanding each other's goals and point of views. Moreover, perceptions and goals are far from being static and changes may easily lead to mis-alignments. All this points to a process of human interaction which is uncertain, dynamic, and only partly rational. We shall see that this characterised our case of the emergence of formal methods, as players with different disciplinary traditions, expertise and experience came together in the design process of a complex microprocessor.

3. The Rise of the Inmos' Formal Methods Constituency

A brief overview of some of the most relevant factors constituting the initial conditions for the emergence of formal methods at Inmos is necessary.

a) The increasing complexity and costs of microprocessor design seem to be playing in favour of the diffusion of formal methods. At present, microprocessors have reached 9 million transistors with the latest Alpha chip from Digital Equipment Corporation (DEC). The Intel Pentium has 3.1 million transistors and the latest Inmos T9000 has 3.3 million. Intel has suggested that by the year 2000, it is possible that what they have called the Micro2000 may have as many as 100 million transistors. Intel envisages a high-performance option for the Micro2000 which would incorporate 4 CPUs executing instructions in parallel and each running at 700 MIPS to give a total chip performance of over 2 billion instructions per second [7, 21]. On this basis, the point in favour of formal methods is quite straightforward. Microprocessors 'will be far too complex for the design to be tested, and manufacturing volumes will be far too high for the design to be wrong!' [31, p.3]. The conclusion is that formal methods will have to be used in order to cope with this situation.

b) The diffusion of formal methods into the microprocessor industry is an incipient development. There is not much indication of widespread use of formal methods in the microprocessor industry. Companies such as Intel are aiming for a 'zero-defect' goal in their chips, with no mention of using, or even intending to use, formal methods in their design process. Continuous improvements in test and simulation techniques seems to be the path followed by leading players in the industry. However, as chips advance towards 100 million transistors, the 'zero-defect' goal will provide a mounting challenge and will not come for cheap. Simultaneously, the penalty for chip flaws is also increasing. Witness the costly wave of bad publicity attracted by the Pentium following the discovery of a bug affecting complex mathematical calculations.

c) Until recently, formal methods have been growing primarily within the province of academia and research laboratories. Indeed, the Queen's Award was given to both Inmos and Oxford University, highlighting the major role of the university in the development and transference of formal methods to Inmos. In addition, their initial promotion into the outside world has been strongly linked to issues such as the specification and verification of designs for safety-critical and security-critical products. This early promotion and association with safety-critical problems seem to have generated a rather controversial image for formal methods. Broadly, formal methods are perceived as demeaning other methods in verification and testing. As we shall see, the perception exists that the formal methods constituency is promoting mathematically-based methods as the best, if not the only true, approach to design verification and greatest safety. The implication for other less mathematical methods such as simulation and testing is that they are 'imperfect' because they are always non-exhaustive, and hence, they cannot have a claim to total verification of a product. This 'perceptual' devaluation of other methods tends to imply a consequent devaluation of the skills of 'non-formal-methods' practitioners in the design process.

d) In principle, the use of formal methods in microprocessor design and validation does not entail conflictive displacement of other methods. Indeed, the increasing complexity of microprocessor design and consequent validation problem is such that different methods are likely to add up to each other's value. In short, formal methods is not a case of antagonistic competitiveness.

3.1. PROMOTING SUCCESS

Earlier on, we saw how it was only with the second generation transputer (T800) that formal methods actually entered the picture of the Inmos' design process. This development was perceived and, indeed, promoted as a great success, primarily, by formal-methods constituents themselves. Thus, in 1990, an article by Inmos' Chief Architect, David May, suggested that formal methods had been used in the design of the second-generation T800 transputer with significant cost-effectiveness [30]. This assessment was given greater substance in a later article, in which it was stated that formal methods had cut the design time of the FPU to nine months from the two years it would have taken using traditional techniques. In addition, 'all of the microcode written using formal methods worked first time round, but some of the FPU instructions written without using formal methods turned out to be incorrect.' [48, p.63]

This image of major success was obviously reinforced by the formal-methods work being conferred the Queen's Award for Technological Achievement. Indeed, it may have well played a key role in the Award being conferred in the first place. Whatever the circumstances, the formal-methods constituency was apparently securing a definite place in the evolution of the design process at Inmos. Cost-effectiveness, less-time-to-market, higher-certainty-of-correctness, award-winning innovation, this would all be music to the ears of any company's management, and it clearly linked formal methods to very appealing goals for Inmos. Heretofore, one would expect further gains for these methods, especially as the transputer design moved forward into new more complex generations. After all, this is precisely the point of the formal methods constituents, namely, that 'we are very close to the point where some of these designs could not be dealt with any confidence without that technology.'[7] A closer look at the experience of the T800 transputer helps to reveal what actually happened. Moreover, since the third-generation transputer, the T9000, has also been developed, it is possible to look at whether formal methods did really gain an unquestionable place within the company's design process following their T800 success.

3.2. THE T800 EXPERIENCE

A combination of five ingredients created the opportunity for the emergence of formal methods at Inmos. First was a market demand. Soon after the first generation transputer -T414 (see figure 1)- reached the market, Inmos realised they needed to provide a hardware floating point unit (FPU) in order to improve the chip's chance of

[7] Interview with D. May, Inmos, November 1992.

selling [35]. This would be the main new feature of the second generation transputer - the T800. Until then, in the T414, 'floating-point arithmetic was supported primarily by a software package written in occam.' [30, p.111] The second ingredient was a desire to speed up and improve the validation process. Attempts to validate the T414 package through normal validation process of executing a large number of test cases was perceived as too slow. This justified the decision to adopt formal methods for the verification of the microcode in the hardware FPU of the coming T800. The third ingredient was the suitability of the microcode work itself to the application of formal methods. The fourth was the historical relation between Inmos and Oxford University Programming Research Group, in particular through the original language of the transputer, Occam. This leads to the final ingredient, namely, the presence at Inmos of key individuals who understood the potential of the new formal methods technology and were keen to see it implemented and promoted within the company design process. As we shall see, Inmos Chief Architect and Occam developer, D. May, played a prominent role in the emergence of the constituency at Inmos.

The events leading to the formal-methods work in the T800 FPU begun to unfold in 1985 after D. May attended a Royal Society lecture by Donald Good who remarked that constructing a verified program in the Gypsy system took "about 5 times longer than the normal (informal) way." (ibid.) May knew that the Occam floating-point package was in many respects an ideal candidate for a formal correctness proof. It was not very long, it was intricate, and a great deal depended on its being correct. As a result, he reasoned that this sort of time for a formally verified version of their software package was well worth trying. In fact, it was estimated that even if it were to take ten times as long as the original package had taken, it would still reduce the overall length of time through the elimination of time-consuming testing. The issue became what to do, because '[at] the time, we had no clear idea how to proceed with the verification.' [31, p.3] The first step was then to discuss the idea with several members of the Oxford University Programming Research Group (PRG), who have played a leading role in the overall formal-methods constituency-building process. The result was that Geoff Barrett (then working at the PRG, now at Inmos) went to work on the problem and constructed a proof which demonstrated the use of formal methods in this application. Basically, what Barrett did was to take the English-written specification for floating-point arithmetic given by the IEEE-754 (the international standard from the American Institute of Electrical and Electronic Engineers - IEEE) and rewrite it into a formal notation. The logical notation Barrett used for this purpose is known as Z[8], and the formally derived packaged was completed in three months and, as hoped, it overtook the experimental validation (that is, the testing of the original package still in progress).

Formal methods constituents were pleased because, for them, this demonstrated the greater reliability and cost-effectiveness of the new technique for developing algorithms for computer arithmetic. They also saw its immediate application in the rapidly approaching problem of verifying the *hardware* floating point unit of a new transputer,

---

[8] Z is a formal specification language based on mathematical notation that employs set theory and logic to build abstract models. It was inspired by French mathematician Jean Raymond Abrial while working at the Programming Research Group, Oxford University, in the early 1980s [11, p. 36].

the T800. [31, pp.3-4] In effect, while work on the software package was proceeding, the design of the T800 incorporating a hardware FPU had also got underway. Thus, the problem of verification was staring at Inmos' eyes. It was estimated that, with available CAD equipment, the simulation method would take more than a year. As May put it, '[c]learly, it was important to find a way of developing the floating-point hardware formally, possibly starting from the already proven software package.' [30, p.112]

The procedure followed for the verification of the microcode of the T800 FPU has been documented in detail in various papers [3, 30, 31, 32, 47, 48]. Basically, it underlines the existence of a clear alignment between the capabilities of formal methods and the technical demands intrinsic to the nature of the problem (see diamond of alignment). A chain of transformation and equivalencies is involved, First was the IEEE754 / Z equivalence done by Barrett. Second was the equivalence between the high-level and the low-level implementations of the floating point unit microcode, both written in Occam. Finally, closing the gap from IEEE to low-level implementation was the transformation between the Occam high-level representation and Z. The work for the last two was mainly done at Inmos by David Shepherd. Barrett believes that it is justifiable to 'talk about a continuous rigorous development from Z to the microcode.'[9]

Most reports concentrate on the technical steps just described. There is hardly a feel for any difficulty, although in practice what was happening was the intra-institutional emergence of a constituency, claiming to bring substantial advantages into a design process which was already underway using conventional practices. True, formal methods were not brought into the upstream design work itself, but rather to verify what was already designed. It was also selective in the choice of what could be verifiable in accordance with the incipient development of the technology. Thus the choice of microcode reflected both recent experience and the amenability of the problem to formal methods treatment. One might think that this approach would generate no problems for the alignment of the 'incoming' technology. After all, conventional simulation methods can never exhaustively test a microprocessor, and this makes the use of a variety of approaches a very desirable factor in order to reduce uncertainty. In other words, formal methods would play a 'non-obligatory complementary' role (i.e., not necessary for other methods to work but adding to the common purpose of increasing confidence in the results of the validation process).

Looking at the 'diamond of alignment' almost all dimensions seem unproblematic, suggesting effective formal-methods implementation and minimal disruptive impact on the Inmos' organisational 'truce'. In practice, however, there was a mis-alignment problem mainly in the users'-perceptions-and-goals (right hand side) dimension of the 'diamond'. Specifically, the target users of formal methods, that is, the engineers in charge of the FPU design did not share the same enthusiastic perception of success of

---

[9] Notes from an interview with G. Barrett by Donald MacKenzie, September 1992. The use of the word rigorous is interesting because Barrett differentiate 'rigorous' from 'formal'. 'I would tend to use 'formal' for things which are step by step machine checked, and rigorous I tend to use in the way in which a mathematician would use it, in other words, you would actually, given enough information or enough insight, be able to fill in all those intermediate steps. It could be done by hand.' (Interview with G. Barrett, Inmos, November 1992).

formal methods constituents. The reasons were partly rooted in expertise-based incommunication and a reaction to the 'superior' image projected by the overall campaign of formal-methods constituency-building; they were also partly to do with the company's 'governance' because the implementation of the 'incoming' technology revealed a lack of clear and agreed ways to assess the relative contribution of the different methods. Thus, engineers did not follow all the mathematics and felt that formal methods were not adding that much to existing methods. One argument was that formal methods were not concentrating on what was seen as the really hard verification problems. Formal methods stopped at the microcode level and did not reach the hardware which was seen as the difficult part.[10] Hence engineers tended to dislike the trumpeting of formal methods as a major input into their process.[11]

The conflict was not antagonistic, however. It was rather a misalignment of perceptions fuelled by differences in expertise, resource allocation and the rather strong promotional campaign of an emerging constituency trying to justify a place in a company's technological process. Thus, Homewood readily acknowledges that formal methods did help, but he is put off by the great deal of credit and publicity given to the role played by the emerging formal methods constituency. The right balance is difficult to ascertain and this was nowhere reflected more clearly than in the perceptions and apportioning of blame and credit for 'bugs' in the microchip. Formal methods constituents have considered a proof of their value the fact that no bugs have been found in those areas where they were involved. Thus, in 1989, they wrote: 'all of the microcode written using formal methods worked the first time round, but some of the FPU instructions written without using formal methods turned out to be incorrect. In addition, there had been a flaw in the design part of the hardware of the FPU, which has crept in before the team began using formal methods.' [48, p.63] The same argument was reiterated a few years later as the use of formal development techniques was hailed as 'highly successful as no errors have been found in the areas covered by these techniques.' [31, p.9] Most importantly, formal methods constituents felt encouraged to write about their contribution in terms of cost-effectiveness, significant less-time-to-market and a higher-certainty-of-correctness not possible by using other design methods.

---

[10] 'Of course, there were some assumptions made, this is the crucial part, that the hardware existed, the hardware did what it was suppose to do, you know, so when I said that A+B, the hardware was there to do A+B correctly. Formal methods never proved that the hardware did A+B, it was just assumed.' (Interview with M. Homewood, November 1992)

[11] 'Formal methods... never ever formally proved the hardware... I didn't trust formal methods. I thought it was very immature.... [and] ... I had no way of reading the Z specification for the IEEE floating point arithmetic. And when it comes down to it, I didn't consider the microcode to be the hardest part... For me, it gave me some personal problems because at the time I was writing a lot of test code and the test code is the most boring, awful thing to have to write and it all runs on the hardware and it's very low level, horrible stuff. Formal methods didn't get rid of it, formal methods cannot get rid of the test programmes... [But] at the time it was very fashionable within Inmos, [so] I found it a little hard because the test code had to be written and these people say formal methods, and I say, what about a test code? and they can't argue, you've got to have the test code.' (Interview with M. Homewood, November 1992) It is worth noting that, in Homewood's opinion, 'the Z-specification... is not complicated or useful...[thus]...I did not want to learn Z or have the time...' (Personal communication between D. MacKenzie and M. Homewood, March 1993)

Homewood's perception is different, he thinks that 'it was definitely another tool, it was a good tool'[12]. His assessment, however, is that the significance of the contribution of formal methods to the overall process has been somewhat exaggerated. To start, 'the parts of the design tested rigorously by conventional methods were also bug free.'[13] At the same time, any comparison should take into account that 'formal methods had two people on the case, I had one, me, and I had to manage the rest of the design process for the FPU.'[14] This lack of resources for one approach tended to create opportunities for the other to make a better contribution. Thus, in relation to the time saved by formal methods, his viewpoint is that 'they definitely caused a benefit but there was never a formal study or a real quantitative study of how much time it did save and what the cost was, or what the cost of the alternative was... It made a difference but it would have been a week or so difference.'[15] Formal methods constituents estimate that the time saved was more substantial, probably about a month, and that this was a significant saving because time to market was [and is] a big issue. In fact, Shepherd thinks that 'the T800 design was completed a fair amount ahead of schedule. Probably, formal methods helped in that it certainly should have meant that we needed less iterations on the microcode.'

In turn, Barrett's opinion is that 'things were quantified actually. David [May] mentions figures like the fact that the component was actually delivered a month earlier than it was scheduled, and this was largely because much of the testing which would need to have been done on the floating point could have been bypassed because of the techniques that we've used.'[16] In other words, as formal methods were able to prove the correctness of certain design areas, this had the effect of severely curtailing the number of tests which were required in these areas, thus bringing the time down. Barrett agrees that 'at the lowest levels you cannot do anything other than simulate. But actually you can save all your simulation time for doing that rather than simulating the higher levels,'[17] where formal methods are effective. A rough estimate puts at Ł1 million the savings accruing to Inmos for one month ahead of schedule design, a figure that has a lot to say about the argument of formal methods cost-effectiveness. Of course, if one takes down the contribution of formal methods to just a week, this would only be Ł250,000 saved. But, as May points out, the cost of formal methods work in the T800 was relatively low - a fact largely deriving from the labour-intensive nature of the process, which at this early stage is very much about formal-methods constituents writing the tools they need. Indeed, he estimates the cost at probably less than a man year which, in his view, would compare favourably with what competitors were

---

[12] Interview with M. Homewood, November 1992

[13] Personal communication between D. MacKenzie and M. Homewood, March 1993.

[14] Ibid.

[15] Interview with M. Homewood, November 1992. Later on, Homewood has estimated that 'the formal methods probably checked only 5% of the possible errors in the T800 FPU or 1.5% of the whole chip. The other 98.5% were covered by conventional testing, of which probably 90% was tested in the first pass.' (Personal communication between D. MacKenzie and M. Homewood, March 1993).

[16] Interview with G. Barrett, November 1992.

[17] Ibid.

spending on setting up simulations and running test cases in simulation; they are paying both for the cost of computer equipment and you are also paying for somebody's time to do all the simulation testing.[18]

Ultimately, the precise assessment of the relative contribution of formal and conventional methods is virtually impossible. For instance, it is not possible to say whether and when a bug found by one method could be found by another. The penalty cost of a bug is also difficult to estimate given that it could remain undetected for considerable time and the negative impact will depend on whether and how many chips have already been sold. Witness the possible $300 to 500 million cost of the Pentium bug for Intel. Moreover, if a flawed microprocessor is involved in a major catastrophic accident the consequences for a company could be dramatic.

3.3. THE T9000 SETBACK

As we have seen, in the design process there is not an 'antagonistic competitive' situation for formal methods. The nature of the design process is such that there is a clear basis for alignment for a variety of approaches; all converging to make the ideal goal of 'zero defect' a closer reality. 'You need as many angles as possible with these designs.'[19] The constituency-building point then seems to be one of recognising this space for goal alignment, while identifying and treating in a careful way the existing mis-alignments of perceptions. To what extent this was the case with the T800 is difficult to judge. Maybe the answer lies in the extent to which the success claimed by formal methods constituents did really lead to the generation of a stable alignment with other players in the Inmos' design process. The start of the design process of the third generation transputer, the T9000, is very revealing here.

In 1989, Inmos started the design of the T9000, a chip over 10 times more powerful than the T800. This transputer reached the market in 1993, having been announced during 1991. In fact, Inmos designed it under a great deal of pressure, as the T800 began to look somewhat obsolete in the face of competing microchips in the market. Time was of the essence, and this seemed to offer a great opportunity for formal methods to expand their presence on the silicon geography of the transputer. After all, a shorter time-to-market and verified correctness had been the avowed contribution of formal methods to the T800. In practice, the design of the T9000 started without using formal methods and the area of the FPU (which had given formal methods a seemingly strong foothold) would not use them at all. Why?

Part of the answer lies in the disbandment of the Inmos' design team following the completion of the T800. Many of the people who had work in the T800 -and had been exposed to formal methods- left the company altogether. For the T9000, Inmos was obliged to assemble a brand new design team, with the result that any cumulative experience beyond the formal methods constituents simply vanished. On the other hand, it is not altogether clear whether such cumulative experience would have automatically translated in cumulative support on the part of T800 engineers. After all,

[18] Interview with D. May, November 1992.

[19] Interview with M. Homewood, November 1992.

one of those who left was Homewood and, interestingly enough, his perception of the exaggerated credit given to formal methods had some part in his decision to leave.[20]

In any event, formal-methods players soon realised that their constituency-building process was back almost to square one. As Barrett reflects, 'it started with good intentions, I think the idea was there in the beginning that it should happen, but actually there were key [social] constituents in the T9000 which militated against formal methods actually becoming part of it.'[21] Nothing was more revealing than the case of the floating point unit. Here, D. May did remind engineers of the successful use of formal methods in the T800 FPU. This was not taken up however. Given their responsibilities, the members of the new team tended to stick to the conventional methods they understood well, and formal methods were not part of this context. The microcode of the T9000 FPU was simpler than that of the T800, since operations such as multiplication and division were now implemented in hardware. This then facilitated the use of conventional methods which were perceived as perfectly adequate for the task in hand. Indeed, for Roger Shepherd (chief architect for the T9000), the use of the word 'formal' may mislead people to think that the conventional methods used in the T9000 FPU were less effective and rigorous. 'What our engineers did with the T9000 FPU hardware was both sophisticated and rigorous. With limited resources, we had to concentrate formal verification work where it was most needed.'[22] But it was also the case that, for engineers, the use of formal methods -in their present state of development- requires a great deal of work and mathematical skill. In other words, if these techniques could have been applied just by pushing a button, then they would have been used. The techniques have to be simplified before they can be applied widely.[23]

In short, lack of tools, the mathematical language, and even, suspicion for the unfamiliar, were all back to exclude formal methods from the early design process once again. In a way, this was hardly surprising. The problem, May reflects, 'is that telling people that the tools they are familiar with are not the best ones is very, very difficult. What we were faced with the T9000 was a substantial part of the design group who had no experience that would cause them to believe there was a need for these methods and, in some cases, a definite sort of resistance to really wanting to get involved in view of the fact that they clearly got this steep learning curve to go through.'[24]

Faced with this lack of understanding and 'rejection', there was little that formal methods constituents could do but to adopt a 'wait and see' attitude.[25] The hope was that

---

[20] 'The reason I left Inmos was mainly cash but also the way the company rewarded individuals in terms of merit (i.e., pats on the back). The formal dogma devalued my work enormously. I meet people and they say: "the T800 FPU that was designed by formal methods"... [T]he engineers at the front end and putting long hours and really worried about whether the chip worked or not, [they] weren't credited.' (Personal communication between D. MacKenzie and M. Homewood, March 1993).

[21] Interview with G. Barrett, November 1992.

[22] Notes from a telephone conversation between Roger Shepherd and Donald MacKenzie, March, 1993. See note 28 for areas where formal methods were eventually used in the T9000.

[23] Ibid.

[24] Interview with D. May, November 1992.

[25] Ibid.

the complexity of the T9000 would eventually expose the limitations of conventional methods, thus 'opening the minds' of members of the design team to other methods. 'It's not at all obvious how you overcome it [the resistance], other than letting them get to the point where they can start to see real difficulties appearing, the deadlines are approaching, and they've no idea how are they going to actually sign on the dotted line to say that the piece of design I've done actually does what it's supposed to.'[26]

In other words, formal methods constituents begun to count on the intrinsic uncertainty of conventional methods for an eventual increase of their appeal to rather worried engineers. Formal methods were eventually introduced, in one degree or another, in the verification processes of important areas of the T9000 (i.e., Virtual Channel Processor, the Processor Pipeline, the Address Generators and the Control Links[27]). These were areas in which the potential for arbitrary events severely tested the confidence of test-engineers in using simulation methods alone. These problem-areas provided formal methods with a new opportunity to continue the build up of the constituency.[28] It is beyond the purpose of this paper to discuss in detail the experience of the T9000. Suffice to note the fact that formal methods were effectively excluded from the initial decision-making on the design of the T9000. Once again, they were only called upon to work on problems appearing in a design process which was already under way. This means that, for all its high-profile image of success, the experience of the T800 had not given the constituency enough clout to secure a permanent place among those shaping the design process of the transputer from the start. The alignment process had not been as successful as the public portrayal of formal-methods achievements with the T800 suggested.

4. Conclusion

This paper started by raising a number of questions concerning the nature of the processes of intra-company diffusion or implementation of information technologies. It looked at the emergence of formal-methods technology in a microprocessor company, scrutinising in detail what was the reality behind the news of almost instant successful implementation of this technology. The theoretical and empirical discussion revealed that intra-company diffusion of technology is essentially a process of sociotechnical alignment in which the specific nature of the technology is deeply inter-twined with perceptual and behavioural aspects. In this context, it was shown that the technical and commercial merit of a technology is not only insufficient, it is indeed an area of

---

[26] Ibid.

[27] *Virtual Channel Processor* is a device which allows any number of logical connections between two processors to be implemented by a single physical connection. *Processor Pipeline* is a device where instructions get executed in a sequence of stages; several instructions are executed at the same time in different stages of pipeline. *Address Generator* is the first stage of the pipeline; it does some simple operation on the register to generate addresses (indexing operations). *Control Link* is the mechanism which regains control of a processor which is running a bad program.

[28] For a detailed analysis of the experience of formal methods in the T9000, see [37].

contention, especially in situations where the organisation's 'governance' lacks agreed ways to assess the relative contribution of the different methods. This understanding was encapsulated in the 'diamond of intra-organisational alignment.'

More specifically, the examination of the experience of the Inmos' formal methods constituency showed that the T800 target users never really embraced the technology. Contrary to what would be expected, for instance in Von Hippels 'user dominated' innovation, in the transputer case, target-user participation had little to do with the implementation and subsequent image of rapid success of formal methods. The technology was actually implemented by the developers themselves, a fact that depended largely on the emerging and 'non-obligatory complementarity' features of formal methods. On the one hand, they are potentially a welcome addition in microprocessor design validation; on the other, they are a new technology still requiring a great deal of development before becoming more accessible to target users. This underlays a process of sociotechnical alignment in which most dimensions in the 'diamond of alignment' exhibited no major barriers. Thus, no antagonistic displacement of established technologies nor any major technical or organisational re-adaptations inside the company were necessitated. Formal methods were effectively implemented in the Inmos' design operation with no visible disruptive impact on the existing organisational 'truce'. True, they were helped by the support of key personalities who were able to mobilise resources for their implementation. Also, this implementation was highly selective. The technology was used to verify only what was already designed and the choice of the problem (i.e., verification of FPU microcode) was clearly aligned with the incipient development of the technology.

Under these circumstances, the critical 'alignment' dimension for the long-term establishment of formal methods became the process of accommodation of roles and rewards between formal-methods developers and target-users. This implied an all-interacting-parties' acknowledgement of effective and substantial contribution to a common objective. Here, we found important areas of misalignment. In particular, we saw that the perception of major success promoted by formal-methods constituents following the experience of the T800 transputer was not unanimously shared within Inmos. As a result, in spite of (or maybe because of) the high-profile external image, formal methods did not really gain an unquestionable place within the company's design process. The technology was plainly left out at the start of the T9000 design process.

Interestingly, the start of the T9000 actually revealed a 'setback' case in which a reported successful adoption/use of an emerging technology was followed by a dramatic loss of acceptance. This type of cases are more difficult to find in the literature of technology diffusion and implementation. One reason is that most analyses tend to concentrate on one-cycle products/processes, or, the long evolutionary diffusion of industrial processes. Cases of implementation along several generations of product-cycles have received much less attention. The latter would be product-families experiencing more than one major design phase of most likely increasing complexity in their lives. This is exactly the case of microprocessors and Inmos' implementation of formal methods in their design process. The important implication of these product-families for 'incoming' technologies is that success in one generation of product does not necessarily imply assured acceptance for the next. In principle, the start of every new product generation carries the opportunity for a setback and players promoting the new technology would do well to be aware of that.

Finally, formal-methods constituents did eventually find a place in the T9000 design process. Perhaps indicative of more favourable times, it was the complexity of this third generation transputer that provided the opportunity for them to re-enter the design process as important players. Thus, if the predictions of ever-increasing microprocessor complexity are correct, then, formal-methods constituents may well be proven right in their contention that their methods, as they continue to progress, will become indispensable. As this happens, the formal-methods constituency might truly become a permanent force in the design process of future microprocessors.

5. References

1. Abernathy, W. and Clark K., Innovation: Mapping the Winds of Creative Destruction, *Research Policy* (14) 3-22 (1985).
2. Badham, R., Introduction: New Technology and the Implementation Process, *The International Journal of Human Factors in Manufacturing* 1(1) 3-13 (1993).
3. Barrett, G., Formal Methods Applied to a Floating-Point Number System, *IEEE Transactions on Software Engineering* 15(5) 611-21 (1989).
4. Bijker, W., Hughes, T. and Pinch, T., eds., *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*, MIT Press, Cambridge, 1987.
5. Braverman, H., *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century*, Monthly Review, NY, 1974.
6. *Business Week*, Bare Knuckles at Big Blue, 26 December 1994, 26-28.
7. *Byte*, A Talk with Intel April 1991, 131-140.
8. Calantone, R. and Cooper R., New Product Scenarios: Prospects for Success, *Journal of Marketing* (45) 48-60 (1981).
9. Clark, J., McLoughlin, I., Rose, H. and King R., *The Process of Technological Change: New Technology and Social Choice in the Workplace*, Cambridge University Press, Cambridge, 1988.
10. Collinson, S., Managing Product Innovation at Sony: The Development of the Data Discman, *Technology Analysis and Strategic Management* 5(3) 285-306 (1993).
11. *Computing*, An Unlikely Couple, 26 November 1992, 36.
12. *Computing*, Better Safe than Sorry, 3 March 1994, 26.
13. Cyert, R. and March, J., *A Behavioral Theory of the Firm*, Prentice-Hall, Englewood, Cliffs., NJ, 1963.
14. *Financial Times*, Costly Miscalculation, 21 December 1994, 16.
15. Fleck, J., Innofusion or Diffusation? The Nature of Technological Development in Robotics, *Edinburgh University PICT Working Paper No.4*. RCSS, University of Edinburgh, Edinburgh, 1988.
16. Fleck, J., Configurations: Crystallizing Contingency, *The International Journal of Human Factors in Manufacturing* 3(1) 15-36 (1993).
17. Fleck, J., Learning by Trying: The Implementation of Configurational Technology, *Research Policy* (23) 637-52 (1994).
18. Freeman, C., *The Economics of Industrial Innovation*, Penguin Books, Harmondsworth, 1974.
19. Freeman, C., The Economics of Innovation, *IEE Proceedings* 132, A(4) 213-221 (1985).
20. Globe, S., Levy, G. and Shwartz, M., Key Factors and Events in the Innovation Process, *Research Management* (16) 815 (1973).
21. Intel, *The Computer Inside: Micro2000*, Intel International (UK), Swindon, 1990.
22. Law, J. and Callon. M., The Life and Death of an Aircraft: A Network Analysis of Technical Change, in *Shaping Technology/Building Society: Studies in Sociotechnical Change*, W. Bijker and J. Law, eds., MIT Press, Cambridge, 1992.

23. Leonard-Barton, D., The Case for Integrative Innovation: An Expert System at Digital, *Sloan Management Review* 29(1) 7-19 (1987).
24. Leonard-Barton, D., Implementation as Mutual Adaptation of Technology and Organization, *Research Policy* (17) 251-267 (1988).
25. Lilien, G. and Yoon E., Determinants of New Industrial Product Performance: A Strategic Reexamination of the Empirical Literature, *IEEE Transactions on Engineering Management* 36(1) 3-10 (1989).
26. MacKenzie D. and Wajcman, J., eds. *The Social Shaping of Technology*, Open University Press, Milton Keynes, 1985.
27. Maidique, M. and Zirger B., A Study of Success and Failure in Product Innovation: The Case of the U.S. Electronics Industry, *IEEE Transactions on Engineering Management* EM-31(4) 192-203 (1984).
28. March, J., The Business Firm as a Political Coalition, *The Journal of Politics* (24) 662-678 (1962).
29. March, J. and Simon, H., *Organizations*, John Wiley, NY, 1967 (1st edition, 1958).
30. May, D., Use of Formal Methods by a Silicon Manufacturer, in *Developments in Concurrency and Communication*, C. Hoare, ed., Addison-Wesley, Reading, Mass., 1990.
31. May, D., Barrett, G., and Shepherd, D., Designing Chips that Work, in *Mechanized Reasoning and Hardware Design*, C. Hoare and M. Gordon, eds., Prentice Hall, NY, 1992.
32. May, D. and D. Shepherd., Towards Totally Verified Systems, in *Mathematics of Program Construction*, L. van de Snepscheut, ed., Springer, Berlin, 1989.
33. McLean, M. and T. Rowland., *The Inmos Saga: A Triumph of National Enterprise*. Pinter, London, 1985.
34. Molina, A. H., *The Transputer Constituency: Building Up UK/European Capabilities in Information Technology*, RCSS, Edinburgh University, Edinburgh, 1989.
35. Molina, A. H., Transputers and Transputer-Based Parallel Computers: Sociotechnical Constituencies and the Build Up of British-European Capabilities in Information Technology, *Research Policy* (19) 309-333 (1990).
36. Molina, A. H., In Search of Insights into the Generation of Techno-Economic Trends: Micro- and Macro-Constituencies in the Microprocessor Industry, *Research Policy* 22(5/6) 479-506 (1993).
37. Molina, A. H., *The Formal Methods Constituency: Diffusion of an Emerging Technology into a High-Volume Industrial Environment*, RCSS, University of Edinburgh, Edinburgh, 1993a.
38. Molina, A. H., The Generation of Large-Scale Capability-Building Initiatives - Extending the Sociotechnical Constituencies Perspective, in *The Management of Information and Communication Technologies - Emerging Patterns of Control*, R. Mansell, ed., ASLIB, London, 1994.
39. Molina, A. H., Technology Diffusion and RTD Programme Development: What Can Be Learnt from the Analysis of Sociotechnical Constituencies? *DGXII Occasional Paper XII-378-94*, C.E.C./DGXII-A/5, Brussels, 1994a.
40. Nelson, R. and Winter, S., *An Evolutionary Theory of Economic Change*, Harvard University Press, Cambridge, 1982.
41. Noble, D., *Forces of Production: A Social History of Industrial Automation*, Alfred Knopf, NY, 1984.
42. Ray, G., Full Circle: The Diffusion of Technology, *Research Policy* (18) 1-19 (1989).
43. Roscoe, A., Occam in the specification and verification of microprocessors, *Phil. Trans. R. Soc. London* (339) 1371-51 (1992).
44. Rothwell, R., Freeman, C., Horsley, A., Jervis, V., Robertson, A. and Townsend, J., SAPPHO Updated - Project SAPPHO Phase II, *Research Policy* (3) 258-291 (1974).
45. Rubenstein, A., Chakrabarti, A., O'Keefe, R., Souder, W. and Young H., Factors Influencing Innovation Success at the Project Level, *Research Management* (9) 15-20 (1976).
46. Sahal, D., *Patterns of Technological Innovation*, Addison-Wesley, Reading, Mass., 1981.
47. Shepherd, D., The Role of Occam in the Design of the IMS T800, *Technical Note 47.*, Inmos, Bristol, 1988.
48. Shepherd, D. and G. Wilson., Making Chips that Work, *New Scientist* 13 May, 61-64 (1989).
49. Simon, H., *Administrative Behavior*, The Macmillan Co., NY, 1957.
50. Simon, H., Human Nature in Politics: The Dialogue of Psychology with Political Science, in *Between Rationality and Cognition: Policy-Making under Conditions of Uncertainty, Complexity and Turbulence*, M. Capanella, ed., Menier, Torino. 1988.
51. Tierney, M., Software Engineering Standards: The Formal Methods Debate in the UK, *Technology Analysis and Strategic Management* (4)3, 245-278 (1992).