Crafting the Virtual Prototype: How Firms Integrate Knowledge and Capabilities Across Organisational Boundaries

Luciana D’Adderio

Research Centre for Social Sciences (RCSS) and the Division of Informatics

2 Buccleuch Place, The University of Edinburgh, EH8 9LW, UK

Luciana@dcs.ed.ac.uk

*

SPRU, Mantell Building, University of Sussex, Falmer, Brighton, BN1 9RF, UK.
l.d-adderio@sussex.ac.uk

Abstract

This paper examines the introduction of Integrated Software Technologies in Product Development focusing on their influence on organisational Experimentation and Prototyping practices. In particular, it explores the role of ‘virtual prototyping’ techniques, concepts and models in facilitating inter-functional processes co-ordination and multi-disciplinary knowledge integration. It argues that the ability of software to support inter-functional co-operation and the co-ordination of knowledge and activities depends on the organisation’s ability to nurture integrating, ‘translation’ routines which support two-directional translation flows between ‘local’ (function-based) and ‘global’ (computer-embedded) knowledge and activity levels. These mechanisms also lie at the heart of dynamic capabilities creation and maintenance.

Keywords: Knowledge and Capabilities Integration, Integrated Software Systems, Product Design and Development, Virtual Prototypes, **Translation Routines**.

1. INTRODUCTION

Experimentation as a form of problem-solving is fundamental to innovation; it consists of "…trial and error, directed by a certain amount of insight as to the direction in which a solution might lie" (Barron 1988, in Thomke et al.,1998:316). Studies in product and process development have shown that iterative trial and error is a significant feature of design (Wheelwright & Clark 1992, Thomke 1998), technology integration (Iansiti 1997) and manufacturing (Adler 1990, Pisano 1996). According to Leonard-Barton (1995), experimenting and prototyping generate new kinds of organisational capabilities: they help creating 'requisite variety' in products and processes, as well as establishing a virtuous cycle of improvement; they also guard against core rigidities by introducing new sources of knowledge, new channels of information, and new methods for solving problems (ibid).

While the importance of experimentation is widely acknowledged, few authors have so far attempted to characterise the technical and organisational mechanisms that sustain and enable experimental practices. These mechanisms often play a greater role in competitive success than simple mastery of new technologies (Pavitt, 1998). They underpin an organisation's ability to achieve co-ordination and integration of knowledge, activities, skills and capabilities across organisational boundaries. In doing so, they lie at the basis of dynamic capabilities' formation and maintenance, as well as supporting the absorption of organisational tensions and conflicts.

By focusing on the area of product design and development, this paper examines the radical influence of advanced software-based prototyping technologies over the practice and the outcome of experimental and prototyping activities. An example of these advances is the adoption of the Digital Product Model, also known as the Virtual Prototype[1]. The Digital Model is, at the same time, a philosophy, a prototype, and a set of procedures: it is a philosophy, in that it outlines the technological and organisational path that organisations have to follow to integrate their development processes; it is a (virtual) prototype, because, as a digital entity backed by a relational database, it contains links to all data, information and codified knowledge required to design, engineer and manufacture a product; and finally, it contains a set of standard procedures, based on industry best practice.

Specifically, this paper identifies the influence of the Digital Model on the mechanisms underlying the co-ordination and integration of experimental knowledge and activities across organisational boundaries. In doing so, it addresses some fundamental but, as yet, unexplored questions: how do heterogeneous organisational functions or ‘epistemic communities’(Steinmueller, 1998; Knorr-Cetina, 1999), and teams, or ‘communities of practice’ (Brown and Duguid, 1996), manage to co-operate during design and experimentation? What is the role of software-based technologies in supporting inter-functional co-ordination and the resolution of organisational tensions and conflicts? How do virtual prototypes influence the integration of heterogeneous knowledge sources (i.e. from various organisational domains) and types (i.e. tacit/codified, formal/informal, software-embedded/people-embodied knowledge) into product definition? Which are the technological and organisational mechanisms that sustain an organisation’s ability to integrate knowledge and activities across functional boundaries?

---

[1] While Digital Mock-up, Digital Product Model, Virtual Product Model, and Virtual Prototype correspond, in theory, to different levels of simulation capability supported by different stages of software implementation (from less to more advanced), these concepts are often used interchangeably.

In order to answer these questions, a qualitative methodology that attempts to capture the mechanisms of co-evolution of software-embedded and organisational knowledge processes in their complexity and in the context of actual practice has been adopted. Such an approach entails moving beyond a ‘purely cognitive’ and abstract characterisation of knowledge, present in much organisational theory (cf: March & Simon 1958), to study organisational cognition as an ensemble of ‘situated’, collective, and ‘distributed’ processes (Suchman 1987, Hutchins 1995, Tyre & Von Hippel 1997). The research on which this paper is based was conducted at two leading automotive and consumer electronics organisations that were adopting the same integrated software system. The collection of field data involved participant-observation, carried out at various times during a one and a half years’ period, complemented by in-depth, semi-structured interviews. The results are presented in the form of two detailed case studies.

2. EXPERIMENTATION AND THE VIRTUALISATION AND ACTUALISATION OF INFORMATION

2.1 The complementary nature of physical and virtual experimentation

The modes of experimentation in manufacturing organisations are radically affected by the introduction of software-based methods and technologies, such as computer simulations and Rapid Prototyping (Thomke, 1998). To date, the theoretical debate has been centred on a dichotomy between physical and virtual prototypes. Thomke, for example, focuses on learning by switching and choosing between different experimental modes (i.e. physical crash tests vs. computer simulations). He sees computer simulation as a ‘substitute’ for real experimentation in fields ranging from the design of drugs to the design of mechanical and electronic products (1998).

While there is a clear economic rationale in substituting physical prototypes with digital simulations, from a cognitive point of view there is an advantage in combining the learning that derives from both types of experimentation. All of the organisations interviewed as part of this study have emphasised that digital prototypes have not replaced but have been used to complement physical experimentation[2]. In other words, as this paper sets out to show, the most interesting changes are taking place where old and new methods are being successfully combined. Indeed, it appears that the effective utilisation of advanced software technologies is related to the extent of integration of the new with the existing methods, procedures, and technologies (Thomke, 1997). In addition to the learning involved in creating a digital or a physical prototype, a second locus of learning involving learning by ‘hybrid’ (physical and virtual) and cross-functional experimental modes has emerged. Such hybrid modes are intended to facilitate the integration of existing and new technologies, skills, routines and capabilities by providing complementary problem-solving approaches. These methods, therefore, have important organisational implications: organisations who are more apt at adapting their routines and create new procedures which are capable of exploiting the potential designed into digital or software-based technologies are likely to generate competitive advantage. This realisation holds important implications both for organisational learning and practice.

2.2 The 'virtualisation' and 'actualisation' of information

How can we characterise the influence of digital technologies on experimental and prototyping activities? Drawing loosely from Levy's theoretical framework, we can describe an organisation's experimenting and prototyping activities as involving a series of interactive cycles of 'virtualisation' and 'actualisation' of knowledge, materials and capabilities [1997, 1999][3]. These cycles are supported by the introduction of integrated software technologies across most manufacturing sectors. Analysing the dialectic between the virtual and the actual can help us to understand and characterise the implications of 'virtual' technologies' introduction and use for product development.

Virtualisation consists in the translation of the information related to a physical prototype into digital. An example of this is the technique by which a physical model (i.e. the clay model for a car body, or the epoxy model for a power drill) is 'scanned', or

---

[2] A complementary set of interviews, conducted in other sectors including Formula One and Aerospace, confirms this evidence.

[3] The 'virtual' has been defined by French philosopher Levy as "that which exists only *potentially* and not in *act*" (1997:XIII). In order to become 'actual', the virtual requires a process of transformation, or *actualisation*, which does not simply entail *logic deduction* (i.e. the pure logic working of a software

‘digitised’ and translated into a digital ‘cloud of points’. The inverse procedure, ‘actualisation’, entails a translation from digital to physical. For example, a virtual 3D CAD model can be actualised into a number of different outputs: a Finite Elements computer simulation, a plotted 2D drawing, a 3D rendering image, a CNC-machined physical model, or a rapid prototype (also physical).

The technical ability to virtualise and to actualise a digital model into many different physical and virtual forms, is a form of ‘digital flexibility’ (Levy, 1997 and 1999). The principle behind the creation of a Digital Model is that there is a single, updated source of product data which is available to all development functions and can be displayed according to a function’s specific requirements[4]. Once generated by the Engineering function, for instance, the digital model can (and must) be used as a common reference point by all other organisational functions: the Toolmaker can use it as direct input to the tooling machines, Analysis can use it as input to finite elements or fluid-dynamic simulations, Marketing can use it to obtain customer or executive feedback, etc.

2.3 Virtualisation and actualisation as problematic translation processes

While the processes of virtualisation and actualisation that underpin digital flexibility are becoming a recurrent feature of the experimentation landscape in development organisations, they are far from straightforward. Below, it is shown how actualisation and virtualisation are not simply abstract or ‘purely technical’ but, because they are embedded in an organisational context, they inevitably involve the set up and maintenance of complex and highly problematic organisational routines.

---

programme) but *creative invention* (i.e. the interaction between people and information systems) (1997:7, *my translation*). For a deeper discussion of Levy’s contribution, see D’Adderio (2000 & 2001).

[4] In practice, the extent of this capability depends heavily on specific implementation circumstances.

Firstly, experimentation and prototyping activities involve the interaction of different organisational functions, having different culture, viewpoints, technological awareness, goals, incentives and capabilities. Hybrid prototyping techniques, as observed during our research, testify that experimentation, rather than belonging to a specific department or discipline field, or being confined to abstract calculation by lone inventors/engineers (as portrayed in much innovation literature) is instead a highly interactive activity which spans *discipline-specific* (i.e. Design, Analysis, Manufacturing, Testing, Marketing), '*internal'* (i.e. between Design and Production) and *'external'* (i.e. the toolmakers, the customers) organisational boundaries. The collective, cross-functional side of experimental activities represents one type of complexity.

Secondly, experimentation and prototyping activities involve the co-ordination of different knowledge types and levels, such as formal/informal, local/global, tacit/codified, personal/social, software-embedded/people-embodied. For instance, virtualisation and actualisation require both tacit and codified inputs for the translation to be effective. Both physical and digital prototypes, in fact, embody *the results* of tacit and codified processes: for example, tacit knowledge is required to interpret the results of computer simulations (corresponding to a first movement, from virtual to physical) (Petroski, 1996, Thomke, 1998). However, tacit knowledge is also required for the opposite purpose of translating the digital 'cloud of data' into a finished 3D model (corresponding to a second movement, from physical to virtual). The process of creating a 3D model from the digitisation of a physical model, in fact, does not correspond to an automatic process of translation but involves substantial creative integrative efforts.

At this stage we can put forward two propositions: on one hand, there is a constant tension, flux or ‘state mutation’ between physical and virtual prototypes, for example between 3D physical and 3D digital models. This suggests that there exists no actual dichotomy between physical and virtual prototypes, but rather a continuous flow created by the interactive multidirectional processes of translation between prevalently digital and prevalently physical “states”. Feedback obtained from testing physical models can be used *a-posteriori* to improve the ‘virtual model’; vice-versa, feedback from the digital model can be used to improve *a-priori* the physical model. Whichever the sources or types of knowledge involved, what is clearly important is that hybrid prototyping techniques provide the richest standpoint to support the early integration of heterogeneous knowledge types and sources into product definition.

On the other, hybrid experimentation techniques tend to draw together the inputs from various organisational functions and disciplines, often causing the dissolution of traditional boundaries as well as forcing the organisation to modify its routines in order to facilitate cross-boundary activities. It is therefore not simply an issue of the relative importance of tacit vs. codified knowledge, but an issue of how to integrate different knowledge sources into product definition, and how to co-ordinate the activities of many different organisational functions around the design, testing and production of prototypes. The ‘fitness’ of the final artefact depends heavily on the effective co-ordination of the activities and knowledge inputs as well as the integration of many different iterative variation/selection processes which take place at several different organisational locations and levels. Once we look at experimentation activities not in abstract but in the context of actual practice, we realise the need to unveil the mechanisms that underpin “the ‘artful integration’ of local constraints, received standardised applications, and the re-representation of information" across organisational boundaries, levels and domains (Suchman and Trigg 1993, in Bowker and Star 1999:44). The issues become particularly significant when analysing the influence of integrated information systems on development practices.

Supporting these propositions requires, therefore, an in-depth, 'contextually-situated' examination of experimentation processes; only a process-centred investigation can in fact uncover the evolution of the complex organisational and technological mechanisms which support an organisation's experimentation and prototyping activities. This analysis reveals that the inverse processes of virtualisation and actualisation require processes of translation: virtualisation requires a process of codification and simplification; actualisation requires a process of reinstatement of subjectivity and local knowledge. Due to the technological and organisational complexity involved, the processes of virtualisation and actualisation are not at all straightforward, but involve establishing and maintaining routines that act as knowledge and activities co-ordination and integration devices. The case studies which follow document the processes of building and maintenance of translation routines at a consumer electronics manufacturer. Very similar patterns were observed at a vehicle manufacturing organisation.

3. THE DIGITAL MODEL AS AN INTERFACING DEVICE AND THE EMERGENCE OF TRANSLATION ROUTINES

The R&D and Manufacturing unit of a leading consumer electronics organisation is implementing the Digital Mock-Up strategy and technology, which is intended to provide the organisation with the ability to digitally simulate the characteristics of the product and the production process as well as some features of the development organisation. The DMU implementation involves radical changes in the technologies as well as in the methods and procedures for design, experimentation and prototyping. While radical changes occur at all interfaces between development functions, this paper focuses on two cases, one involving changes occurring at the interface between Industrial Design and Engineering and a second at the interface between Engineering and Analysis[5].

3.1 The Digital Model as interfacing device between Industrial Design and Engineering

The first case study analyses the changes occurring at the interface between Industrial Design (ID) and Engineering. As a result of the Digital Mock-Up implementation, new procedures are being introduced in both organisational functions; these are aimed at facilitating the translation and transfer of industrial designers' knowledge from solid clay models and paper- (or computer-) supported[6] sketches into a 3D Digital Model; the Model represents the major deliverable by the ID department, and is normally handed over from ID to Engineering after 'model sign-off'.

---

[5] Additional evidence shows that similar radical changes occur at the interfaces with other functions, including 'external' enterprise organisations such as the Toolmaker.

[6] i.e.: ALIAS or Macintosh photo-realistic conceptual renderings.

The new design and prototyping procedures, supported by a number of advanced software-based technologies, are especially devised to ‘capture’ and embed some of the industrial designers’ and engineers’ knowledge into the Digital Product Model during the early stages of PD. The managers, designers and engineers interviewed have emphasised the difficulty of transferring the designer’s knowledge from a clay model, or a hand sketch, into a digital CAD model; they have also highlighted the difficulty of embedding the designer’s intention into the Digital Model. This difficulty is essentially due to the visual and kinaesthetic nature of engineering knowledge (Vincenti, 1990, Ferguson, 1993a,b, Henderson, 1991 & 1995).

3.1.1 Creating and modifying the 3D (SURFACE) CAD model

The principal output of the Industrial Design (ID) function is the 3D digital SURFACE model, which represents the outer part of the final product, commonly named ‘skin’ or ‘shell’. ID creates the 3D model by combining mechanical functionality and Marketing requests and “…injecting them with the company style” (Interview/BK). The ID Surface Model is designed in CATIA, a leading software application[7]; when it is ready, it is signed-off to Engineering where the ‘internals’ are designed. Later in the process, the model returns to ID where, through various iterations, it is modified according to the changes introduced by downstream development activities (i.e. Engineering, Analysis, Production). The process of translating ID’s sketches and clay models into a digital surface model is, in contrast to what is commonly suggested by much of the technical literature, highly problematic. Despite this, it is essential that efforts are made to ensure that as much as possible of the designers knowledge and intent is incorporated into the digital product definition as early as possible in the development process. This in fact

---

[7] CATIA is the leading application in the world-wide automotive and aerospace CAD markets.

represents an important means to maximise the input of the ID function into product definition.

There are several methods available to create the Digital CAD Model. For example, the model can be created interactively in CATIA starting from sketches made by hand or by using ALIAS, another widespread software application; alternatively, it can be obtained by scanning hand-refined clay models into the computer system[8]. The latter technique, involving the translation of a physical clay model into a Digital CATIA Model, can be executed either manually, by measuring sections of a clay model and typing the data into the computer, or digitally, by ‘scanning’ the clay form. In both cases, “…the digital data obtained *has to be resurfaced in CAD by experienced CAD engineers* to become a quality surface model” (Interview/BK).

The scanning methodology, an example of ‘reverse engineering’, employs techniques which enable the machine to ‘read’ the surface of the clay model and translate it into digital data. The methodology used at the observed organisation involves two steps: 1)scanning the model; 2)reconstructing the surfaces. The first step involves mechanical measurement which produces an observational ‘cloud of points’ of digital data. The second step involves constructing lines and surfaces that closely approximate but, nonetheless, simplify or idealise the original clay model (Interview/BR).

This process is often compared to taking a snapshot of the clay model: theoretically, a simple and *automatic* process. Through deeper analysis, however, the procedure appears far from being automatic and seamless. Indeed, several designers and managers have

---

[8] Often a combination of the two methods is used.

described it as a highly problematic portion of the process: "…the second part of the methodology is to reconstruct the surfaces and that's where the problem stands at the moment …*It isn't a press-button process*: for example, if you scan this clay prototype, you are picking up all the little problems, the draft angles go out of control…So, whatever you scan, it cannot be the finished product" [9] (Interview/AS). And also: "…what's important is that this representation *isn't easily turned into the form of the model; so, the cloud of points is not necessarily the product*" (Interview/BK).

Some critical contributions on the use of Computer-Aided technologies in creative design have emphasised the lack of an interactive tool able to assist the transition phase from 2D sketching to an accurate CAD model: "In early phases of design, the progress of the design process is documented by sketches, package drawings and tape drawings. Increasingly, computer-aided systems are being introduced to replace and complement conventional design development steps. As a result of this development, *numerous media gaps appear, characterised by multiple changes between various two- and three-dimensional representational media as well as physical and computer-internal models.* In addition, the transition during the early phases of industrial design from the sketching design themes to the preparation of a CAID reference model *of spline quality* is not supported by computer-aided systems" (Boniz & Krzystek, 1996:162-3, *emphasis added*).

At our consumer electronics organisations, the transition and translation of the design from a physical clay model into a Digital CAD Model is made problematic by what one engineer calls 'breaks in the process': "…you get the *break in the process* which is

---

[9] The finished product, or deliverable, for ID is the signed off 3D surface model.

where you are trying to go from the [physical] models to the CATIA form, because CATIA doesn't lend itself to complex free-formed surfacing. And then when you try to go back and make modifications again you get *another break in the system,* because the system does not allow you to make modifications. You really have to rework it” (Interview/AS).

Whichever technique is used for translating the surface data from both sketches and clay models into a CAD model, a careful reconstruction process is required. This process must ensure that the surfaces of the resulting digital model are high quality, because the quality and effectiveness of the work performed by downstream functions (i.e. engineering, tool-making, manufacturing) depends heavily on upstream surface quality: “The aim is to create *high quality* surfaces. If you create good quality surfaces you can then use them in making your 3D model. If you don't pay enough attention to surfaces you get a lot of problems downstream, you have difficulty to machine it, and in using SOLIDS. *It's just like laying the foundations* (Interview/AS). A quality SURFACE model is in fact mandatory to allow engineers to create a quality 3D SOLID[10] model; this is then circulated around the whole enterprise and becomes an intermediary among functions in all subsequent stages of development.

3.1.2 Visual and kinaesthetic knowledge

High quality design surfaces allow for smoother machining (and, therefore, actualisation) of the 3D model into a physical shape at the Production stage. In order to obtain quality surfaces, the experience of the designer is mandatory: “In CATIA terms this is where the skill is...The problem with surfaces is that you do not only need to be

---

[10] A ‘SOLID’ model is a 3D virtual prototype created in CATIA from a SURFACE model.

able to use CATIA, you need to be able to use forms, to understand forms, to understand what this [surface] is doing, to understand the term ‘fair’. ‘Continuity’ is one thing, ‘fair’ is another: [a surface] can be totally continuous but not fair. *If you look down the surface you will see it*”(Interview/AS). Surface quality is therefore a property which is difficult to capture geometrically and can only be assessed by building a physical prototype.

The possibility to ‘look’ and ‘touch’ a prototype is at the very basis of the processes of elicitation and capture of design knowledge. The continuing need for building physical versions of prototypes is related to the visual and kinaesthetic nature of engineering knowledge, as documented, for example, by Ferguson: “…the eyes and the fingers -the bare fingers- are the two principal *inlets to trustworthy knowledge* in all the materials and operations which the engineer has to deal with” (1993b:50, *emphasis added*). Interestingly, the requirements for the use of visual and kinaesthetic knowledge by engineers are not eliminated by the introduction of recent digital technologies, but only transferred to more subtle design knowledge niches. The automatic scanning of the clay model leaves in fact *‘pockets of ambiguity’* that only experienced and able engineers and designers are able to fill.

This view is shared by the practitioners interviewed: “Industrial designers are very skilled in creating this model by hand, and this is the most effective way of achieving the design shape…*The problem with that* [computer representation] *is that every time you make a complex shape on the computer,* [in order] *to verify it, you need to have it cut.* Because of the ergonomic requirements of the shape, you want to make sure that it *feels right* in the hand. So you always have to have the shape cut…With the guys working in foam, it means that you get instant feel if you are going in the right direction. *The problem then comes to get this complex [physical] shape into the digital model. That is difficult.”* (Interview/AS).

Existing software-based technologies struggle to ‘capture’ the designer’s subjective intent into the physical prototype: “…What you need is an easy method to create and then modify surfaces. When I say easy, it is still a skilled job. It is always going to be difficult to represent that [clay surface] in the computer, because *you can't take the computer to be subjective, or to understand that there is actually a change in curvature along here*. That needs to be defined by the eye, it is not possible for the computer to define the form on its own. You can have a start by scanning it and giving it its best fit, *but it won't be how the designer intended it to be”*(Interview/AS).

Working with a physical model, either hand-made or obtained by CAD-printing, is, therefore, a fundamental means to incorporate the designer’s knowledge into the artefact: “The thing is *you can't judge a form on the screen.* You always need a clay, a cut piece. Now, whether you make it in a block using surfacing and give it to the machine tool to cut it, or whether you get a model maker to cut it, you need to handle it, to see it...*At the end of the day you have to cut it*” (Interview/AS). It is only by ‘cutting’, i.e. by using NC machining to generate a solid physical model out of the digital one, that some mistakes or imperfections can be detected and corrected: “So you get this [computer] representation. Then you can cut. *Once you’ve cut, you can see immediately where the problem was: this* [model] *is not really what the designer wanted”* (Interview/AS).

Building a physical prototype therefore helps capturing the designers' intent into the digital model, as efficiently as possible; it also helps to ensure that several knowledge sources (from different organisational functions) can enter the product definition process early, i.e. by circulating the resulting physical prototype around the organisation; finally, it can be used for validation and verification of the digital model, that is the trustworthiness and adequacy of the model from several viewpoints (i.e., executive management, customers, engineers, marketing etc.).

3.1.3 Translating into the CAD language

At a deeper level of analysis, software-based technologies reveal limitations in capturing the designers' knowledge and intent. This problem partly stems from the fact that such technologies are designed by engineers, with engineers in mind: "…It is basically *an engineering way to describe surfaces*. Suppose you have one section here, another here and another here, the [resulting digital] surface will go through those sections. That is the nearest you could go to deriving a surface from a hand sketch. *That's how CATIA describes it and the way an engineer would describe it*" (Interview/BK).

The software demands a *translation* from the designers language into the engineers and CAD language, in order to ensure the best possible approximation of the designer's intent within the final model and, therefore, to maximise the industrial design input into product development. The engineering bias in CATIA, however, creates significant problems to the designers[11]: "...We could get these things [the surfaces] created in CATIA, we knew we could do it. But, initially, we couldn't get the designers to do it. *Because of the engineering paradigm that is in CATIA*, the industrial designers did not want to use CATIA to create these things. It is a culture mismatch, and also a mismatch of skills and interest. That's not the way they [the designers] like to work. They don't actually mind computers, they like the Mac, for instance, because that does the things they want it to do. But *CATIA gives engineering what engineering wants, it gives the toolmaker what the toolmaker wants*, but you don't actually want to do the surfacing in CATIA" (Interview/BK).

---

[11] Similar biases have been reported in several other CAD applications.

The methodologies used by industrial designers to define, construct and modify surfaces differ substantially from those of an engineer, and so does the industrial designer's way to understand surfaces. In order to clarify this difference, an industrial designer recalls the example of the construction of *spline*[12] curves in aerospace: "[For us] …it is the same as in the aircraft and the shipbuilding industry. They have applied similar methodologies in their 'lofting departments', which is where for years and years guys have been trained to generate smooth forms. They where initially doing it full size, using spline curves, and then taking sections from the spline curves…They draw the curve they want, then they lift the sections from there using two-dimensional geometry techniques. It used to be done with a very long piece of wood, which had a number of weights on. These weights would create a naturally smooth surface. Then they would create sections…[or]…'frames', and then they would stick those to the side which will hold them together. If you try to build the sections first, [there is] no way you would get a smooth shape" (Interview/BK).

These observations mainly refer to the *making* of a CAD digital surface. A similar problem, however, arises in relation to *modifying* those surfaces (i.e. after receiving feedback from other functions): "…for example, CATIA surfaces…are made of constrained patches. The geometry is based on a number of control points. *In the engineering paradigm* you modify the surface by modifying those control features that were used to define the surface. *This is not the way a designer wants to work.* It restricts your ability to modify it, which is a problem because changes are inevitable, there is nothing more certain than changes" (Interview/BK). In this context, "The way CATIA works… is still not satisfactory, because it doesn't allow [you] to modify surfaces the

---

[12] "A *spline* is a flexible strip that can be made to go through a set of points in such a way as to produce a

way a designer would. The other methods they [software developers] have introduced is to use ‘canonic’ forms, circles on it, *but the only people that use* [canonic] *circles and lines are engineers, the designers don't like it*”(Interview/AS).

The ID’s way to modify surfaces differs substantially from the engineers’ approach: “…once the surface is there in 3D, you can see the way the light fall on it, the way it interacts with surfaces...you want to be able to make changes, that's what the software is not good at. Once a surface is modified, the surface doesn't allow to maintain the relationship with the surrounding surfaces. I can't say: I would like to maintain continuity there, a curvature, or a specific style line (i.e., I want that always to be at a certain number degrees). You can't describe that” (Interview/BK). This problem is significant as errors in surface continuity will generate problems later on, in the process of making a SOLID model: “…The boundary conditions are very difficult to control. I.e., if you have a gap between two surfaces, you have a tangency condition that is not respected, then when you offset these surfaces along the normal, so that you make a solid out of it, then those conditions get much worse, and you go outside your tolerance for closing the volume, and that requires rework. So *if you get it right from the beginning then it gets easier downstream*” (Interview/AS).

The process of generating a Digital Model is, in synthesis, anything but straightforward. It involves two major kinds of difficulties: 1)*capturing and translating* prototypes and other *knowledge inputs into a digital form*; this is difficult because it requires that industrial designers adapt to an engineering way to define and design surfaces; 2)once a digital surface is obtained, the difficulty resides in *incorporating feedback and changes*

---

smooth curve (Onwubiko, 1989).

to it; this entails a method of modifying and controlling surfaces and their parameters that is, again, typical of an engineering approach.

3.1.4 The 3D CAD model as a standardising device

A legitimate question may arise at this point: if the translation of the designers' intent into a digital model is so problematic, would this not offset the advantages of using the CAD tools? Why bother with adopting the software tools when their use appears so controversial and counterintuitive? A mechanical engineer provides an answer: "The problem is we need CATIA for *achieving the benefits downstream in engineering:* the higher quality the surfaces are, the easier it is downstream, the easier the engineering, the easier the data transfer" (Interview/AS).

The operation of translating physical prototypes and sketches into a digital model, involving a process of reduction of ambiguity and of codification of designers knowledge, is useful in allowing for part of the *knowledge to be shared* and passed on to engineering and other downstream development functions. The operation of codification and reduction of ambiguity allows for a different type of experimentation and flexibility to be generated: *digital flexibility*. Digital flexibility, in our context, consist in the fact that, once the model is translated into digital data, this can be extracted (or actualised) under many different forms, as well as updated. CAD-printing is one example, but there are many other ways of reproducing a digital model. Such 'digital' flexibility, however, is achieved only by reducing ambiguity and subjectivity; this is obtained by translating the language of each function into the CAD language, which is biased towards the model of the engineering language. In order for them to become a constituent part of the process, inputs into product definition must be therefore translated into the CAD 'language', and then back into the 'local language' during successive iterations.

The 3D CAD Model (SURFACE) thus becomes a *'collector'* of codified design knowledge coming from different functions and prototyping activities. The process of codification implies a strong 'reduction of ambiguity'. While, in fact, a certain level of ambiguity can be beneficial at the upstream development stages (i.e., as a means to convey the designer's intent and as an inlet for tacit knowledge), ambiguity often creates serious problems downstream at the Production and Manufacturing end. The difficulties encountered in transferring surface models into a common sharable CAD model are therefore justified in view of the *'downstream benefits'*; these include the ability to create a 3D Model that can be later used as an interface between all development functions. The 3D digital surface model is used in fact by Engineering to create the 3D SOLID Model, the main engineering deliverable. Once the 3D SOLID model is created, this can be used by all development functions as a basis for successive iterations. The CAD SOLID Model becomes therefore an intermediary for all subsequent interactions between 'upstream' and 'downstream' development functions.

Conceptually, the 'released' CAD model behaves as a *standardising device* (cf. Star & Griesemer, 1989, Henderson, 1991, Fujimura, 1992, Bowker & Star,1999): in the name of increased flexibility, it demands the translation of all codifiable development inputs into a language that is mostly familiar with engineers, while not necessarily with other development functions. This flexibility, however, only works if the model is re-appropriated and reinterpreted by each development functions. The evidence collected here suggests that *local re-appropriation practices* are required in order to facilitate *the elicitation and input of local and tacit knowledge into the codified product definition*. An example of local re-appropriation of the global CAD model is the CAD-printing technique, whereby the data that represents the 3D CAD model is sent to NC machining whereby is ‘printed’ into a physical prototype. A similar result can be obtained via stereolithography or other ‘fast prototyping’ techniques. These technologies generate a physical prototype out of digital data; the prototype is then used by designers to verify that the shape visualised on screen is effectively what they had intended. Any changes are later re-incorporated into the single, common ‘digital product definition’. This emphasises that there is a continuous flow and constructive tension between physical and digital states, as well as between local and global knowledge and activity levels. Only the local re-appropriation of the CAD model will allow the results of local tacit knowledge processes to be embedded into the artefact definition *(local adoption of the global CAD model)*[13].

While the function of the local departments therefore remains fundamental, a translation process must be in place which supports continuous knowledge exchanges between the local and the global (CAD) level; this should ensure that as many different knowledge types and sources as possible are incorporated into product definition, as early as possible. This especially includes heterogeneous, non-articulable and, therefore, uncodifiable knowledge sources which are not easily captured by software. *These translation processes are at the heart of co-ordination and communication among various development functions.* Inter-functional communication and co-ordination depend heavily upon the effectiveness of these translation and integrative procedures. To borrow a term from the Sociology of Technology, the released 3D CAD model can be described as an ‘*obligatory point of passage’* (Latour, 1987) between different development functions. Inter-functional exchanges and knowledge transfers take place

---

[13] Re-appropriation procedures represent, therefore, an important means to ensure the integration of ‘sticky information’ (Von Hippel, 1994), tacit knowledge, local idiosyncrasies, requirements and viewpoints into global (software-embedded) product definition.

via translation and re-appropriation routines that *connect local and global knowledge and activity levels as well as integrating heterogeneous (specialised) knowledge sources.* These routines help to accommodate the tensions arising from the dialectic between standardisation and non-standardised practices and codes as well as between *local* and *global* levels. This "…permanent tension between the formal and the empirical, the local/situated and attempts to represent information across localities..." has been so far under-explored and under theorised (Bowker and Star 1999:44).

It appears clear that part of the challenge that the organisation faces concerns the ability to strike a balance between the global and the local knowledge and activity levels; this balance should ensure that 'central' control and co-ordination through standardisation is achieved, while allowing for the incorporation of rich, formal and informal, local knowledge into product definition; this is important especially at the early development stages where the product specification is not yet clearly identified. The *interactions between local and global levels* should support the early incorporation of heterogeneous knowledge types (i.e. tacit and codified, internal and external) and sources (i.e. discipline-specific) into product definition. These routines also support the integration between existing and new technologies, skills, and capabilities.

3.2 The Digital Model as interfacing device between Engineering and Analysis

The first case study has illustrated how the CAD (SURFACE) Model becomes a principal intermediary between Industrial Design and Engineering. The example that follows captures a similar trend occurring at the interface between Engineering Design and Analysis; the intermediary, in this case, is the CAD (SOLID) Model.

3.2.1 A single common model

The Digital Mock-Up implementation has introduced radical changes also at the interface between Engineering and Analysis; these include the introduction of integrated software modules that enable engineers to perform analyses directly on CAD geometry. The assumption behind this is, analogously, that, once a 3D Digital Model is produced, it can be used as an interface for all subsequent interactions between development functions. Prior to the introduction of integrated software, "…A volume model had to be built *specifically* for analysis; then we would construct from the volume model the surfaces, and all this was very time consuming. Previously, you had too many things to do before you got the answer*, and you had to do them in order to get the answer*…In the new paradigm*, you build a Solid Model*. This has to be built anyway, it is not built especially for the analysis. *Then we can do the analysis directly on the solid model*" (Interview/BK). The main idea is that, by adopting the 3D (SOLID) Model, analysts gets to work on the same model as the engineers: "So, where there were several models and duplication, there is now *a single common model* which every function has to refer to" (ibid).

Once the Model is generated, the software allows engineers to apply 'loads' and 'boundary conditions' *directly on their design geometry*, to automatically mesh the model, run an analysis, and post-process the results without exiting the integrated software environment. Following the introduction of integrated software technologies, first-order analysis work is shifted from the analysts to the designers, who can perform simple analysis iterations which help improve their design and identify the 'best part model'. After an 'optimal' model has been identified, the first-order analyses can be sent directly to an analyst, which can set up higher-order analyses, such as Finite

Elements Analysis (FEA), using a specialised analysis tool, without having to rebuild the geometry. The analysts' FEA model can then be sent back to the designers, enabling them to visualise the CAD geometry and automatically update the master assembly model in the software database (Deitz, 1997).

3.2.2 The integration of Design and Analysis

The aim of integrated software systems is, therefore, to help practitioners "…*close the design loop,* by providing bi-directional associativity so that CAD geometry flows directly *without translation* to the pre-processor, analysis engine, post-processor, and then back to the master assembly model in the CAD system's database" (Deitz, 1997:95, *emphasis added*). The integrated software's philosophy is to create a *seamless two-way information flow* between Engineering Design and Analysis, thus completing the communication and information loop between the two functions. Authors have named this phenomenon as 'the convergence of Design and Analysis'(ibid).

Such integration and 'seamless communication', however, is not automatically achieved as a consequence of software implementation. The convergence of Design and Analysis in fact involves the *a-priori translation of the Analysis language into CAD language.* This translation enables the analysts to perform iterations directly on the 3D geometry model created by the Engineers. As with the previous case study, the adoption of the 3D CAD model by Analysts is enabled by the prior migration of the FEA language to a way of representing the model data which is familiar to Engineering.

The language migration, in this case, is accomplished by linking the features found in parametric, feature-based CAD systems, familiar to engineers, with components of finite-elements models, familiar to Analysis. This corresponds to translating the language of Analysis, which is based on abstract finite elements, into the engineers' language of geometric entities. This language translation is meant to render analysis more intuitive for engineers; the software, for example, "…is capable of presenting the results of an analysis *in terms of geometry and shape, which a designer can understand* *more easily*, rather than, say, stress plots" (Palframan, 1999:16, *emphasis added*). In order to integrate design and analysis, the latter is therefore made more intuitive, thus enabling engineers to pose problems and interpret the results in an engineering context.

3.2.3 The Analyst's specialised expertise

The engineers are therefore developing some analysis skills: they are increasingly using automated analysis software as an aid to identifying the critical design parameters, evaluating their interactions, and eventually determining the best overall design approach. The analysts' skills, however, are not being made redundant. Even though advanced software systems may automate many routine tasks, "…they can't replace engineers with specialised expertise" (Deitz, 1997:100). *Specialised expertise* is, in fact, required at all stages of analysis. During pre-processing, for example, the body of the design is divided into elements; deciding the most appropriate shape and size of these elements requires a sound understanding of the physical model and the Finite Elements procedure (Onwubiko, 1989). While the pre-processing stage requires knowledge about expected stress patterns, the post-processing stage requires experience to interpret the results (ibid).

As a consequence of the introduction of integrated software technologies, practitioners have therefore divided Analysis-related activities in two types. The first type of activity is concerned with an increase in the automation of the finite elements solution process using *linear elastic static analysis* in which the engineer's input only involves the choice of the appropriate mathematical model for analysis. The second type, which includes the *non-linear or dynamic analysis* of structures and fluids, still requires a greater degree of involvement and expertise in finite elements methods by the engineer

(Bathe, 1996). In our consumer electronic organisation: “We have just purchased a modeller from the CAD provider which does the analysis. We intend to use this in two ways: firstly*, to do FEA of simple parts to be used by the engineers. The engineers now will be taught how to use the analysis: they will build a SOLID model, and then will use the analysis on the SOLID model.* The engineers have got a reason to build the model for FEA, and that also gives them skill and *the skill is used in preparing the right model for analysis*; secondly, because now we have these things [the 3D SOLID models] so much earlier than a number of years ago, we are now in the position of being able to give that model to the Moulder *who has got experience with plastics*. We can say: this is what the form is, this is what we want, tell us if we are right or wrong. We still use the expertise outside, it is an extended enterprise” (Interview/BK). The 3D CAD model is therefore sent as information input to the Moulders who hold specialised knowledge and can perform higher-order type of analysis: “The moulders…have this sort of system [Mould-Flow], they would do the analysis for you, but then they need the information. So now we intend to use CATIA' s data preparation capabilities to interface with Mould-Flow. *The analyses are done by specialists* because you need somebody who knows how to get the information from CATIA, knows the package, knows how to do the analysis and how to interpret the analysis” (Interview/BK).

The 3D Model is therefore used to communicate the design intent and to collect specialised knowledge from other development functions such as Analysis and the Mould-makers, allowing to input these specialised knowledge sources back into digital product definition (the common database, and therefore the single Model)[14]. The nature

---

[14] While our example centred around the interfaces between Design and Engineering and Engineering and Analysis, the same type of pattern emerges at other critical interfaces, such as the one between Engineering and the Toolmaker and Mouldmaker, whereby the solid model is used as information input to these ‘external’ organisations (Interview/BK); similar translations also occur at the interface between

of the exchanges between Engineers and Analysts is also evolving: Engineers are able now, in turn, to prepare their 3D model in a way that is most useful to analysts. This, again, improves co-ordination in the opposite direction and promotes an inverse translation flow, this time from Engineering to Analysis. Meanwhile Analysis, as a development function, retains its fundamental role by ‘specialising’ on higher order, complex analysis problems.

The CAD (SOLID) model becomes, therefore, an intermediary between Engineering and Analysis: created by engineers, it is passed on and used by analysts for their trials. Again, similarly to the previous case study, the analysts’ input into product definition and development is mediated by the CAD model which, at the end of the analysis process, embodies some of the analyst’s knowledge. The model becomes the intermediary for all subsequent exchanges between Engineering and Analysis. As seen above, the improvement in co-ordination and knowledge transfer depends upon the adoption of a common product model, which in turn requires the partial migration of the analysts language towards the language of engineers, or the CAD language. As seen in the first example, co-operation in experimentation entails the construction of specific experimentation procedures which support the translation of knowledge flows between functions and levels. While, on one hand, the Analysts’ language is modified to become familiar to Engineers (*translation from local to global*), on the other hand, the engineers’ 3D model, is ‘locally’ modified in order to be adjusted to meet the requirements of the Analysts (*translation from global to local*).

3.3 The emergence of translation routines

Design and Prototyping where the 3D solid data is used as input to the Stereolithography and Laser-

In conclusion, the above evidence has shed light on the techno-organisational processes of selection and variation. We have seen that an interesting feature of the ‘virtualisation of information’ is that, once digitised, information can be processed in many different ways as well as represented in many different guises. The digitisation of physical prototypes allows for their early translation into 3D digital data models; these then become intermediaries in the subsequent interactions among organisational functions, an obligatory point of passage for the knowledge that is to flow from local functions into global product definition and vice-versa. From an organisational standpoint, the adoption of the 3D modelling technology corresponds to the creation of one of a series of important interfacing devices. Now each function draws directly from the ‘released’ computer Model, performs its prototyping iterations and simulations, and then inputs the results back into the Digital Model.

While the 3D Model as a standardising device performs an important co-ordinating function, we have also highlighted that this would not be effective unless mechanisms are set up which ensure that the globally collected knowledge, elaborated into the form of the Digital Model, is re-interpreted and re-appropriated at the local level by each development function. At the level of the individual function or department, the digital models are actualised in many different digital or physical prototypes; the process of actualisation or local re-appropriation is achieved by re-embedding into the digital model the context-, person- and department-related knowledge and contingencies. During these processes, ambiguity is restored, codified knowledge re-interpreted, local meaning is re-constituted, as the ‘local’ community of practice ‘appropriates’ the Digital

---

synthering machines which produce ‘fast’ prototypes by printing CAD data (Interview/KT).

Model. We have also analysed the emergence of ‘translation procedures’ that support these two-directional translation flows.

These translation procedures support inter-functional co-operation by allowing multiple translations between the CAD language (and the language of engineers) and the ‘local’ discipline- and function-specific languages. Co-ordination between heterogeneous development functions is promoted by the creation and maintenance of these two-directional translation processes which help the process of embedding local knowledge into global product definition as well as supporting the appropriation of the global model by local development functions. The existence of translation routines is also attributable to the need to draw from local and informal knowledge sources and input these as much as possible, into the software-embedded global product definition. Indeed, if non-articulable knowledge was as uninteresting as supposed by economists, who focus principally on the margin between codifiable and codified knowledge (i.e. Cowan, David and Foray, 1998), there would be no need for organisations to set up and maintain these translation procedures.

4. CONCLUSIONS AND POLICY IMPLICATIONS

This paper has analysed the influence of integrated software-based technologies on design and experimentation activities conducted across inter-organisational and inter-functional boundaries. In particular, it has focused on the way such technologies reshape the mechanisms by which heterogeneous organisational knowledge sources (i.e. from various functions and domains) and types (i.e. tacit, articulable and codified) are transferred within and across organisational boundaries and the way these are integrated into virtual and physical artefacts. In observing the day-to-day co-evolution of technological and organisational practices, our paper has also addressed the implications of software introduction on collective knowledge-building activities, focusing on the way software systems affect the patterns of collaboration and knowledge sharing among heterogeneous organisational 'epistemic communities'(Steinmueller, 1998; Knorr-Cetina, 1999), and 'communities of practice' (Brown and Duguid, 1996). In this context, we have observed the role of virtual and physical prototypes as 'points of obligatory passage', 'knowledge repositories' and 'intermediaries' among different communities and as loci where organisational conflicts are absorbed, and temporary truces can be reached.

We have argued that the introduction of integrated software tools radically reshapes these fundamental processes, for example by introducing the so-called 'virtual prototyping' techniques and concepts. While emphasising that the *Digital*, software-embedded *Model*, shared among all functions involved in PD and beyond, can act as reference point for all organisational functions and therefore help co-ordinating their differing viewpoints, we have also highlighted its limitations. We have argued that, rather than being straightforwardly supported by the introduction of standardising software tools and models, inter-functional co-operation in experimentation and design involves the local appropriation of the digital model by individual development functions, according to their specific requirements, knowledge and objectives. While the digital model can potentially act as a standardised interfacing device and facilitate the transfer of knowledge across functional boundaries, this works only to the extent that it is supported by the construction of local appropriation routines.

We have therefore demonstrated that effective co-ordination entails the formation of *translation routines*, that integrate formal and informal, tacit and codified, local and global, software-embedded and people-embodied, heterogeneous and standardised knowledge sources and levels. Such integrative routines support two types of knowledge flows: first, they facilitate the translation of local into global knowledge as well as facilitating the early embodiment of local multidisciplinary knowledge into global product definition; second, they support knowledge translation from a global back to a local level, facilitating the re-appropriation of the digital model by each development function, according to their specific requirements and viewpoints. This continuous process of routine building and maintenance is therefore absolutely crucial for the software-embedded philosophy to work in practice. Formal and informal co-ordination mechanisms embedded in routines represent the principal means to repair discontinuities and bridge the techno-organisational gaps between heterogeneous functions, knowledge types and sources, and old and new technologies.

The notion of translation routines provides a more satisfactory characterisation of how software technologies are able to co-ordinate actions and knowledge across different communities of practice and epistemic communities; it also explains how heterogeneous inputs from specialised organisational functions are integrated and how these are able to collaborate while maintaining divergent knowledge, interests and viewpoints[15]. The notion of translation routines helps to explain how organisational tensions and conflicts, due to the coexistence of heterogeneous, specialised functions, are absorbed[16]. Indeed we have demonstrated that such heterogeneity (a consequence of the increase in division of labour and specialisation) can enrich the development process by providing specialised knowledge inputs. Among these inputs is that portion of knowledge which is impossible to articulate but which, nevertheless, is required to improve the product definition process. The emergence of translation routines testifies that, contrary to the codification debate's belief, this portion of knowledge is not simply relevant but essential to the development process.

Further, we have importantly emphasised that the emergence of integrative routines also underlies the organisation's ability to create and maintain *dynamic capabilities* and therefore, ultimately, to sustain its innovative, adaptive and competitive potential. Uncovering these mechanisms represents an important theoretical and empirical step as the existence of dynamic capabilities has been assessed so far only ex-post, as a measure of improved performance. This finding also provides empirical support to the suggestion that the concept of dynamic capabilities is far from 'an empty box', and that it can, and indeed must, be explored in order to improve our understanding of organisational behaviour and performance (Winter and Zollo 1999, Winter 2000).

---

[15] Drawing from the Engineering Epistemology and the Sociology of Technology debates, D'Adderio (2000 & 2001) provide a deeper discussion of the significance of translation routines than it is possible here.

[16] In this sense they perform a governance role similarly to Nelson and Winter's routines (1982).

The above results have been obtained by moving past an abstract conceptualisation of the experimental activities performed by development organisations and by studying knowledge as situated, that is integral to and not separated from knowledge-production activities; the insights provided by our case studies could only be acquired by treating Vincenti's conceptual separation between 'what engineers *know'* and 'what engineers *do'* as an indistinguishable whole (1990). We believe that our approach represents the way forward for organisational studies centred around knowledge issues: understanding organisations as knowledge-repository and -creating systems cannot, and should not, be viewed as separate from understanding how knowledge processes work in practice. In a similar way, we have opted for a characterisation of the knowledge creation and transfer processes as highly interactive, in the belief that to study them as processes typically performed by individuals could imply at best overlooking important issues, and at worst acquiring findings which are irrelevant or altogether misleading.

Finally, we would like to conclude by drawing some speculative policy implications. A first issue concerns the need for software user firms to account for the difficulties involved in appropriating standard, software-embedded procedures and models. On one hand, software-embedded 'best practice' procedures and models can be relatively easy to acquire[17]. Acquisition can be facilitated, for example, by purchasing a software system 'solution' that embodies industry best practice and by exploiting the technology transfer expertise of software producers and consultants to implement it. On the other hand, our evidence demonstrates that such standardised, 'coded' procedures and models are of little use unless they are locally appropriated and effectively transformed into actual routines and prototypes. It is only by locally appropriating and adapting the

---

[17] The ease of appropriation, however, will vary among firms, this being a function of the relative 'fit' between the standardised practices and the existing organisational processes and structures.

software-embedded, standardised practices, models and methodologies to the new idiosyncratic organisational context that these can in fact acquire value.

In this respect, our investigation strongly indicates that a diffusion approach runs the risk of seriously mis-stating the organisational costs and productivity effects of software adoption processes. It emphasises that, in evaluating the costs and benefits of Information Systems implementation, the ‘invisible work of customisation’ that is required to translate (or ‘actualise’) software-embedded procedures (and models) into actual routines (and artefacts) must also be taken into consideration. The standardisation of practices does not in fact unequivocally imply reduced costs of knowledge acquisition, as knowledge, models and practices have to be ‘recreated’ at each new organisational location. These implications should also be taken into account when drafting public policies that are aimed at supporting the transfer of best practice across firms, sectors and countries.

The above inferences are also relevant from the viewpoint of software systems producer organisations. As integrated, enterprise-wide systems become more and more generic, following the producers’ intent to devise software solutions that can be applied to an increasingly wider range of firms and sectors, these are also likely to require an increasing amount of customisation by the user organisation. As a consequence, software producers need to build greater flexibility and customisation potential into their systems in order to facilitate the process of adaptation of generic systems to local, context-specific, circumstances and requirements. This involve designing systems that, while embodying standardised practices and models, are also flexible enough to allow for extensive local customisation and adaptation.

References

Adler, P., 1990, *Shared Learning*, Management Science 36:938-957.

Bathe, K. J., 1996, *Finite Elements Procedures*, New York: Prentice Hall International.

Boniz, P., Krzystek, P., 1996, *Reverse engineering in combination with digital photogrammetry*, in: M. J. Pratt, R. D. Srinam and M. J. Wozny (eds.), Product modelling for computer integrated design and manufacturing, London: Chapman & Hall.

Bowker, G. C., Star, S. L., 1999, *Sorting things out: classification and its consequences*, Cambridge (MA): The MIT Press.

Brown, J. S., Duguid P., 1996, *Organisational Learning and Communities-of-Practice - Toward a Unified View of Working*, Learning and Innovation, in: M. D. Cohen, L. Sproull, Organizational learning, Thousand Oaks: Sage Publications, 58-82.

Cowan, R., David P. A., Foray, D., 1998, *The Explicit Economics of Knowledge Codification and Tacitness*, TSER/TIPIK.

D'Adderio, L., 2000, *Inside the Virtual Product: The Influence of Integrated Software Systems on Organisational Knowledge Processes*, PhD Thesis, Brighton: University of Sussex, forthcoming in:

D'Adderio, L., 2001, *Managing Information Systems in Product Development*, Book Series on Industrial Informatics and Integrated Manufacturing Business Systems, RSP.

Deitz, D., 1997, *The Convergence of Design and Analysis*, Mechanical Engineering, March 1997: 93-100.

Ferguson, E. S., 1993a, *How Engineers Loose Touch*, Invention and Technology, Winter 1993:16-24.

Ferguson, E. S., 1993b, *Engineering and the Mind's Eye*, Cambridge (MA): The MIT Press.

Fujimura, J. H., 1992, *Crafting science: standardised packages, boundary objects, and "translation"*, in: A. Pickering (ed.), Science as Practice and Culture, Chicago: University of Chicago Press, 168-211.

Henderson, K., 1991, *Flexible sketches and Inflexible databases: visual communication, conscription devices, and boundary objects in design engineering*, Science, technology and Human Values 16(4): 448-473.

Henderson, K., 1995, *The visual culture of engineers*, in: S. L. Star (ed), The culture of computing, Oxford: Blackwell, 196-218.

Hutchins, E., 1995, *Cognition in the Wild*, Cambridge: MIT Press.

Knorr-Cetina, K., 1999*, Epistemic cultures: how the sciences make knowledge*, Cambridge (MA): Harvard University Press.

Iansiti, M., 1997, *Technology Integration: Making Critical Choices in a Turbulent World*, Boston (MA): Harvard Business School Press.

Latour, B., 1987, *Science in action*, Cambridge (MA): Harvard University Press.

Leonard-Barton, D., 1995, *Wellsprings of Knowledge - Building and Sustaining the Sources of Innovation*, Boston (MA): HBS Press.

Levy, P., 1997, *Il Virtuale*, Milano: Raffaello Cortina Editore.

Levy, P., 1999, *Cybercultura - Gli usi sociali delle nuove tecnologie*, Milano: Feltrinelli.

March, J., Simon, H., 1958, *Organizations*, Cambridge (MA): Blackwell.

Nelson, R. R., Winter S. G., 1982, *An Evolutionary Theory of Economic Change*, Cambridge (MA): Harvard University Press.

Onwubiko, C., 1989, *Foundations of 'Computer-Aided Design'*, New York: West Publishing Co.

Palframan, D., 1999, *Computer-Aided Engineering: The Trick's in the Training*, The Engineer 7:15-6.

Pavitt, K., 1998, *Technologies, Products and Organisation in the Innovating Firm: what Adam Smith Tells Us and Joseph Schumpeter Doesn't*, Industrial and Corporate Change 7(3): 433-451.

Petroski, H., 1996, *Invention by Design - How Engineers Get from Thought to Thing*, Cambridge (MA): Cambridge University Press.

Pisano, G., 1996, *The Development Factory*, Cambridge (MA): Harvard Business School Press.

Star, S. L., Griesemer, J. R., 1989, *Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology*, 1907-39, Social Studies of Science 19:387-420.

Steinmueller, W. E., 1998, *Does Information and Communication Technology Facilitate 'Codification' of Knowledge?*, TIPIK/TSER, SPRU: University of Sussex.

Suchman, L., 1987, *Plans and Situated Action*: The Problem of Human-Machine Communication, Cambridge (MA): Cambridge University Press.

Thomke, S.H., 1997, *The role of flexibility in the development of new products: An empirical study*, Research Policy 26:105-119.

Thomke, S.H., 1998, *Managing Experimentation in the Design of New Products*, Management Science 44(6): 743-762.

Thomke, S. H., Von Hippel, E., Franke R., 1998, *Modes of experimentation: an innovation process - and competitive - variable*, Research Policy 27:315-332.

Tyre, M.J., Von Hippel, E., 1997, *The Situated Nature of Adapting Learning in Organizations*, Organization Science 8(1): 71-81.

Vincenti, W. G., 1990, *What Engineers Know and How They Know It - Analytical Studies from Aeronautical History*, Baltimore: The John Hopkins University Press.

Von Hippel E., 1994, *"Sticky Information" and the locus of Problem-Solving*, Management Science 40(4): 429-439.

Wheelwright, S. C., Clark, K. B., 1992, *Revolutionising Product Development*, New York: The Free Press.

Winter, S.G., Zollo, M., 1999, *From Organisational Routines to Dynamic Capabilities*, TSER/TIPIK.

Winter, S.G., 2000, *Comments to the Dynamic Capabilities Session at the CCC Conference*, Chicago (IL), 7-9 April 2000.