CODIFIED-TACIT AND GENERAL-SPECIFIC KNOWLEDGE IN THE DIVISION OF LABOUR AMONG FIRMS. A STUDY OF THE SOFTWARE INDUSTRY

Rosa Grimaldi, Salvatore Torrisi

1. Objectives[1]

This study analyses the process of knowledge codification and the division of labour among software firms. The empirical analysis aims at providing insights into some dimensions of the production and distribution of knowledge.

First, software production is often referred to as a 'craft' or 'creative' activity similar to industrial design. However, the rising labour costs due to the competition among firms to gain access to a relatively small pool of skills and the internationalisation of the software market spur firms operating in this industry to adopt more efficient production techniques and to restructure their organisation. This in turn stimulates the codification of knowledge underlying software development activities. This paper investigates the extent to which processes of industrialisation of software activities lead to more codified knowledge, which type of knowledge is codified and which one remains tacit in this processes.

Secondly, the empirical part focuses on the nature of knowledge (tacit vs. codified) that firms share and exchange with other partners by means of collaborative agreements. In particular, it analyses the interdependence between knowledge codification and collaborative agreements. Collaborative agreements require a certain amount of codification of knowledge, which is necessary for exchanging and sharing knowledge among different organisations. Codified knowledge allows firms that enter a collaborative agreement to develop a common 'language' or to adapt the respective 'corporate languages' to each other so that communication between partners can occur. This suggests that in order to enter a collaborative relationship, firms must have invested in knowledge codification. On the other hand, the experience acquired through collaborative agreements induces a higher level of knowledge codification, or at least stimulates the development of organisational ‘interfaces’ which make some pieces of corporate knowledge transparent and as such transferable across firms. We wonder what is the significance of these interactions and feedback in the software industry.

The type of knowledge exchanged and shared obviously depends on the objectives of the collaboration. There are different types of division of labour between firms in the software industry. Many software producers are involved in various commercial agreements with distributors, resellers and retailers. Some of these agreements require a close interaction with the downstream firms (e.g., VAR, Value Added Retailers). Knowledge of specific users’ and markets needs, which tends to be more tacit and ‘localised’ (codified but poorly articulated in written documents), is exchanged on these occasions. But more often, the knowledge exchanged is very simple and codified as in the case of contractual agreements with IT product distributors. Large software producers often set up contractual relationships with smaller firms. Often, these relationships are centred on the outsourcing of activities, which are not considered critical by the contractor (e.g., programming, detailed design, and maintenance). In other circumstances, outsourcing may involve also critical activities, such as system design and system integration. These forms of division of labour in the production of software often require a great deal of technical knowledge exchange between the partners. This knowledge is in part codified and embodied into products (software programs). Finally, joint R&D projects require a more intense exchange of codified knowledge embodied in designs, standards, users’ requirement specifications, development tools, documentation and object code. However, al large amount of tacit knowledge is probably produced and exchanged in R&D agreements. Our case studies focus on technological and commercial agreements with the aim of analysing the specific economic incentives to invest in knowledge codification in different activities.

It is important to note that this paper does not aim to assess the importance of tacit vs. codified knowledge in the software industry. Instead, we are interested in understanding some factors that affect the changing boundaries between tacit and codified knowledge in the software industry.

The paper is organised as follows. Section 2 illustrates concepts emerged in the relevant literature. Section 3 analyses the organisation of knowledge and inter-firms networks in the software industry. Section 4 analyses the characteristics of knowledge codification and exchange in four software Italian producers. Section 5 concludes the paper.

2. Knowledge Production and Transmission across Firms: survey of the literature

Knowledge is a complex and variegated good, which can be tacit or codified, localised or abstract. These characteristics of knowledge bear important consequences for the economics and management of innovation. On one extreme end of the spectrum, knowledge is assimilated with information and, as such, gives rise to market failures which call for public support to innovation (Arrow, 1962). On the other end of the spectrum, knowledge is tacit, embodied in skills and can be in part transferred through personal, informal contacts and training (Winter, 1987). Even skills and capabilities based on a formal, scientific background (for example mathematics or physics) are, in part, tacit. As Nelson (1962) and other scholars have illustrated, the knowledge created during scientific activities is usually difficult to articulate and evaluate from outside. The implications for firm strategy and public policy are intuitive. Unlike information, tacit knowledge cannot be easily transmitted or imitated. For these reasons follower firms and countries have to accumulate ‘absorptive’ capabilities in order to take advantage of the externalities produced by innovators (Cohen and Levinthal, 1989).

Tacit knowledge or 'know how' is also referred to as ‘procedural’ knowledge to mean that it is possessed by some individuals knowing how to perform given tasks or having the skills or the ability to solve a given problem. A fundamental difference between codified and tacit knowledge is that the latter allows firms to solve specific problems, even when there is no general understanding of the reasons behind these problems or the optimal rational methods for their solution. Skills or know how are associated with the use of implicit routines or procedural rules that can be shared via learning, imitation and practical examples, rather than explanations and manuals.

Between information and purely tacit knowledge there are intermediate levels of knowledge codification. Knowledge is fully uncodifiable when there is no written ‘codebook’ to which the members of an ‘epistemic community’ can refer. This is often the case of process technologies, new scientific disciplines and business organisations. In these circumstances, collective routines, conventions, procedures or ‘procedural authorities’ and charismatic leaders may represent the ‘loci’ of knowledge. Knowledge is codified but unarticulated or latent when a group or an ‘epistemic community’ shares a common knowledge base, which is codified into a ‘jargon’. For example, the language known to the specialists of a scientific discipline or technology can be contained in a ‘codebook’ whose existence and 'authority' is implicitly recognised within the epistemic community but it is unknown to outsiders. Finally, knowledge is codified and articulated when a ‘codebook’ exists and is transparent to ‘non specialists’ in the form of textbooks, dictionaries, manuals and standards (Cowan et al., 1998).

Moreover, the degree of knowledge codification is in part endogenous in that there are economic incentives that affect the decision of organisations to invest in codification. The codification of knowledge entails high initial fixed costs that lead to increasing returns. The benefits of codification arise from: a) the reduction of costs of knowledge acquisition and greater reliability of information storage and retrieval; b) reduction of uncertainties and information asymmetries, which facilitates knowledge transaction; c) reduction of costs for building up standards, which represent a language by which production processes can be described; d) increasing opportunities of 'division of innovative labour' which lead to economies of scale and dynamic economies in the production of knowledge (Arora and Gambardella, 1994; Cowan and Foray, 1997).

Yet, the codification process is somehow hampered by the fact that some aspects of knowledge remain 'sticky' or extremely difficult to codify and transfer across firms. Some scholars argue that the distribution between tacit and codified knowledge does not change over time (Lundvall and Johnson, 1994; Dosi, 1996). According to these authors tacit knowledge is needed to use codified knowledge, thus, if there is an addition to the codified knowledge base, there must also be an addition to the tacit knowledge base. Also, the body of knowledge underlying complex production processes is often difficult to disentangle and transmit in separated batches because of the intricate interconnections among different parts and tasks of these processes

This body of studies shows that the boundaries between tacit and codified knowledge vary across industries and are affected by product complexity and the rate of technical change. Moreover, tacit and codified knowledge tend to co-exist both in highly unpredictable activities such as scientific research and in more routine activities such as traditional manufacturing industries, where design, styling and related activities largely rely on know how and competencies that are relatively difficult to imitate[2]. Furthermore, the codification of knowledge is conditioned by the complexity of the organisation (scale and scope). When firms grow and enter new markets their organisational structure is put under pressure and new communication channels are developed to make co-ordination possible (Chandler, 1990).

When products are complex and innovative activities require different types of scientific and technological knowledge, firms have to mix internal competencies, knowledge and experience with external sources of knowledge, by establishing linkages with other organisations such as universities, specialised suppliers, and users (Teece, 1986).

The characteristics of knowledge, its complexity and degree of codification, affect the organisation of firms and industries. This is acknowledged in the evolutionary theory of the firm and in the standard theory (transaction costs and agency theory) (Winter, 1993; Combs and

Ketchen, 1999; Jensen and Meckling, 1995). But the relationship between the nature of knowledge and the relationships between organisations is still relatively unexplored in the empirical literature. This is not a one-way relationship. On the one side, the difficulty to codify knowledge reduces the opportunities to set up linkages with other firms and therefore favours the accumulation of idiosyncratic internal capabilities, which in turn reinforce the tacit dimension of the firm's knowledge base. But, on the other side, collaboration among firms is made necessary by product complexity and rapid technical change. The growth of inter-firm networks spurs organisations to develop new languages, standards and 'interfaces'. [3]

3. Organisation of knowledge and inter-firm networks in the software industry

This section aims at illustrating the factors that affect the evolution of knowledge in the software industry. These factors can be summarised as follows:

1. - Appropriability of innovation;
2. - Organisation of software production;
3. - Skills and market standards;
4. - Inter-firms collaborative agreements.

An important dimension of the software industry evolution is represented by the appropriability regime. As pointed out in various studies, there are significant differences in the degree of appropriability across industries which reflect differences in the intrinsic characteristics of the knowledge base of each industry and the ability of institutions devoted to the protection of intellectual property rights (IPR) to develop instruments which meet the needs of specific industries. Moreover, appropriability changes over time with the evolution of the technology underlying an industry (Levin et al., 1987; Malerba and Orsenigo, 1996). The software industry is traditionally characterised by a low degree of appropriability which is largely due to the intrinsic difficulty to disentangle innovative and protectable expressions of original ideas, such as an 'user graphic interface', from unprotectable ideas, such as an algorithm. For this reason, copyright is used in software as an alternative to patents. However, even copyright represents a weak instrument of appropriability in the case of software. This weakness is also explained by the fact that this is a relatively young industry and therefore the extension of copyright to software products is relatively recent. In the US a Software Amendment to the Copyright Act was introduced in 1980 while in Europe the European Commission issued a directive concerning the application of copyright to software in 1991. Moreover, although copyright does not require the disclosure of the source code, reverse engineering from object codes (machine codes) is relatively easy.

The uncertainty surrounding legal protection of software explains why many software firms rely on secrecy and therefore, tacit know-how to protect themselves from imitation (Torrisi, 1998). The increasing importance of packaged software (software products) as compared with custom or bespoke software increases the appropriability of innovation in this industry. First, copyright and even patents are more effective in the case of software products than for custom software because it is simpler to identify the 'expression' of ideas. Second, new software products, especially those for PCs and the Internet, can enjoy an even stronger form of protection associated with increasing returns from learning by using and network externalities which in turn give rise to the emergence of dominant designs or market standards. Increasing returns give first comers a cumulative advantage towards followers. A survey conducted by one of the authors in 1990 shows that European software firms mainly rely on dynamic appropriability (lead time, continuous innovation and the possession of skilled personnel) as opposed to legal protection (patents and copyright). However, follow-up interviews conducted in 1997 indicate that legal protection, especially copyright, becomes more and more important. These results probably depends on the declining uncertainty surrounding legal protection of innovation in the software industry and the rising importance of software packages for European software firms (Torrisi, 1998).

This evolution of the software industry affects the boundaries between tacit and codified knowledge. In particular, with the increasing reliance on legal protection the owners of a technology (e.g., operating systems or browsers) have to disclose at least in part their knowledge in order to get legal protection.[4] Obviously, most codified knowledge is in fact embodied into software products, components and development tools, not in development processes, which remain largely into the 'tacit' domain. It also important to remember that the disclosure of 'source code' is not required by copyright. Moreover, software activities on the 'technological frontier' are difficult to codify. For organisations, which face dynamic environments characterised by a high selection of concurrent designs, the costs of codification are too high relatively to the expected benefits, which are very difficult to predict.

The organisation of software production provides interesting insights into the economic factors that affect the balance between tacit and codified knowledge in this industry.

Attempts at rigorously organise software development activities have been made since the 1960s (Bohem, 1981). Some large firms have pushed the process of industrialisation of software production up to a level that is close to the traditional manufacturing model. The literature has referred to this approach as the 'software factory', which is described as 'a deliberate attempt to transform software from an "unstructured service" to a "product" with a guaranteed level of cost and quality'. Among the main factor that have led to the software factory approach, especially in Japan, there are the possibility of re-using codes and other components across different projects and the shortage of skilled personnel (Cusumano, 1992, p. 467 and 475).

Different software factories have the following characteristics in common: an integrated set of development tools, standardised procedures and management practices, and a matrix organisation. In some cases there is a division of labour between corporate engineers, which perform critical design and project management, and subsidiaries or subcontractors, which specialise in programming. Some software factories, such as Toshiba, also have developed an incentive structure (awards, controls and so on) to spur re–usability of components. These experiences are limited to large software firms. These have greater resources (and incentives) to invest in software engineering and to impose a stronger managerial discipline to their software production compared to small firms. The majority of small software firms are often described as relying on a job-shop, craft-like activity with little re-use of tools, methods and codes across projects. They employ highly skilled professionals whose productivity is often very unpredictable. Also, productivity is reported to vary extensively from one project to another. One reason for poor productivity is the intensive use of old programs like Cobol which 'does not lend itself to efficient programming practices; indeed it encourage inefficiency' (Garber, 1993, p. 110). A study conducted by one the authors on a sample of European software firms shows that in the early 1990s software firms focused mainly on development tools for the automation of specific phases of the development cycle (CASE tools), and only a few firms tried standard methodologies and development environments that integrate different tools (Torrisi, 1998). Moreover, specialised software firms showed a smaller propensity to adopt software engineering tools as compared with larger, integrated hardware–software producers. This picture changed somewhat at the end of the 1990s. Follow up interviews conducted in 1997 show that the use of integrated development environments and object-oriented tools is more diffused than it used to be in the past among large software firms. Increased international competition and the new opportunities offered by new development techniques (e.g., greater user friendliness and lower training costs) have spurred large specialised software (and services) producers to adopt a more efficient and flexible system of production. It is important to notice, again, that this picture shows the experience of large software and services firms. Probably, many small firms still rely on traditional job-shop, craft-like production systems.

This evidence suggests that a larger use of 'scientific' or formal systems of production is being used by software firms. This in turn produces new incentives to codification of knowledge, which we shall explore in the next section.

Skills and market standards

The relatively young age of computing as an applied science and the rapid technical change make the formal training of skills a critical issue. Since software is a skill-intensive industry, skill shortage represents a major bottleneck for software activities. The lack of skilled personnel is particularly important with respect to system engineers (analysts) and, programmers, especially those with knowledge of fourth-generation, object–oriented programming languages (like C++ and Visual Basic) and Internet-related development tools (such as Java, HTML and XML). A survey of software producers in the largest European countries conducted through the 1990s shows that the most serious critical skills shortage concerns project management, system engineering and application and services activities (Malerba and Torrisi, 1996).

More recently, the ‘Year 2000’ issue, the impact of the European Monetary System and the growth of Internet and e-commerce are putting a lot of pressure on skill supply. The European Commission has estimated about 500,000 vacancies in 1998 and about 1,200,000 by 2002 (EC, 1998).

Moreover, experienced software engineers possess a great deal of knowledge, which is specific to a particular development language or application environment adopted by a firm or a group of firms. For instance, programmers and analysts that have accumulated a significant experience in the development of Enterprise Resource Planning (ERP) programs for SAP, the world leader of ERP integrated platforms, will find it difficult to get a job in firms that develop programs for a different platform (e.g., Baan).

The competition among software firms for the acquisition of skills increases the turnover and therefore reduces the appropriability of training investments. This in turns reduces the incentives to invest in 'generic' training, which is a typical source of codified knowledge for software firms (e.g., the knowledge of a standard development language like C++). By contrast, firms tend to invest in firm-specific skills, accumulated through experience and firm-specific training, because these investments produce knowledge, which is of limited use outside the firm. This type of training may be used as a way to reduce skill turnover and preserve the knowledge stock of the firm.

The recent evolution of the industry however seems to reduce the opportunities for firm-specific training. Until very recently in the software market there were many incompatible platforms and development environments. The community of software developers was fragmented into different ‘epistemic’ groups, which used their own languages. This is still the case of developers of software programs for mainframes and minicomputers. In the case of personal computers and workstations a market standard has emerged (MS-DOS first and Windows for PCs later on). The software engineering skills then tend to become more codified and standardised and this sets in motion a process of convergence of the software development community towards common practices and tools. For instance, most CAD developers, such as Parametric Technology Corporation, which specialise in CAD packages for various Unix workstations have recently focused on products for the Windows platform, a market niche dominated by firms like AutoCAD. Similarly, the take off of Internet application software will probably increase the convergence of development practices (e.g. Java and XML development tools), although many different standards co-exist in the Internet world.

In this context software firms try to retain skilled personnel through specific incentive schemes such as profit sharing and stock options. Alternatively, they can share the knowledge of key processes and technologies within the firm in order to reduce the negative effects of turnover.

Inter-firms networks represent an important characteristic of the organisation of this industry. To have an idea of this trend we focus on over 600 external links (mergers and acquisitions, minority stakes, joint ventures and other collaborative agreements) established by a sample of 38 large European and US software firms during the period between 1984 and 1992 (Torrisi, 1999).

Over this period, the 18 European firms included in the sample set up on average about 1.6 external linkages per year against about 2.2 of the 20 US firms.

We can distinguish between *research-oriented linkages* and *market-oriented linkages*. The former may involve rival firms (for example, consortia for the definition of common standards or joint R&D agreements) or firms specialised in different stages of a technological ‘filière’. The latter are usually set up by firms that operate in different stages of a technological ‘filière’ (for example operating systems suppliers and turn-key systems developers) or in different regional markets.

Over 70 per cent of external operations signed by the European firms were *market-oriented*, against about 24 per cent of *research-oriented* operations. The sample firms have signed market-oriented linkages to gain access to specialised commercial assets or service expertise, and new markets. The cross marketing deal between Sema Group and Finsiel signed in 1992 is a case in point. Examples of research-oriented linkages are the joint development of a videotext software package for IBM mainframes by Cap Gemini and IBM in 1984 and the acquisition of 80 per cent stakes of Technologies machine Art robot manufacturer by Cap Gemini in 1987.

The share of *research-oriented* operations is higher for the US firms as compared with the European firms (34 per cent). This difference is probably due to the large number of US firms specialising in packaged software which show a comparatively high involvement in R&D activities (Torrisi, 1999).

These data indicate that through external linkages software firms aim to gain access to both general scientific or technological knowledge (concerning new technologies, platforms and standards) and to more context-specific knowledge (linked to particular markets, users and applications). What are the implications of these linkages for the process of knowledge codification at the firm level? We expect that the development of these linkages spur firms to increase their incentives to codify their knowledge base through the development of organisational routines and training. This hypothesis will be analysed in the following section.

To sum up, we have discussed some factors that affect the evolution of the boundaries between tacit and codified knowledge in this industry. In what follows we shall analyse few case studies to understand the dynamics of these factors along with other factors (e.g., organisational complexity).

4. Knowledge codification and division of labour: evidence from case studies

4. 1. Data and methodology

Our analysis is centred on four software firms based in Italy - Think3 (3D CAD, three-dimensional computer aided development software), Geographics (GIS, Geographical Information Systems software), Formula and ESA Software (ERP, enterprise software applications and services).

These firms have been selected according to the following criteria. First, they represent different types of product complexity. CAD and GIS software are relatively simple, stand-alone products, based on specific market standards (e.g., Windows and AutoCAD). These products are used for technical applications (such as technical design and territory monitoring). On the other hand, ERP applications are complex products whose development implies the design and integration of a large number of components addressing different dimensions of the user organisation and relying on different technological standards. For example, an ERP package is typically made of a core programme which focuses on the management of different processes internal to user organisations (e.g., accounting and auditing) and a series of complementary modules which addresses different functions (e.g., CRM, customer relationship management, and supply chain management). By contrast, stand-alone products, such as a CAD package, address specific processes or functions.

Second, these firms differ with each other in terms of organisational complexity (size and number of locations). Our selection criteria then allow to analyse our main explanatory variables (changes in software production and inter-firm agreements) controlling for the effect of firm size and product complexity.

These firms are representative of the population of medium-large software firms in Italy (and many other continental European countries) on many grounds. First, three of them have started between the 1970s and the early 1980s as service providers (ad hoc software development, software on subcontracting and testing, software packages reselling) and migrated towards packaged software thereafter (Geographics has been founded in 1993 and belongs to a group founded in 1975). Second, they draw a significant share of their revenues from the national market (Think3 being the only one with a significant degree of internationalisation with over 50% of total revenues coming from abroad). Third, they have started to look for new risk capital (venture capitalists and the stock exchange market) to nurture their growth (Geographics, as mentioned before, is part of an Italian software group). Fourth, they are reshaping the organisation of their development activities in order to increase the efficiency of their operations and to improve the quality and responsiveness of their products and services in a context of increasing competition. Finally, they have set up quite extensive networks of collaborations in Italy and abroad (see the Appendix for a description of the sample firms).

The interviews were conducted with top technical (R&D) managers. In two cases (Think3 and Geographics) the person interviewed was one of the company founders.

With four case histories we cannot measure the importance of codified knowledge in software development neither we can properly test ‘theories’ or research hypotheses. But we can illustrate some characteristics of the codification process when a certain degree of knowledge codification is observed. We can also describe the incentives and obstacles to knowledge codification in firms that operate in a competitive environment, which requires a significant amount of knowledge exchange between firms. Finally, the differences in term of product and organisational complexity produce interesting implications for the economics and management of knowledge codification. The analysis then provides suggestions that can help future empirical research.

The main issues addressed in our analysis are the following.

- What are the most critical capabilities that are employed in software development?
- What are the main sources and the ‘loci’ of tacit and codified knowledge?
- What are the incentives to codify tacit knowledge in these firms?
- To what extent the modularisation of problem solving activities and the use of new development techniques affect the process of codification?
- What is the role played by collaborative alliances in the process of knowledge codification?

4.2. The organisation of knowledge within the firm

These firms rely on different types of key capabilities or skills for their development activities. This reflects their different specialisation.

Think3 considers as highly important the competencies which are employed into two different phases of development activities, namely analysis of user's requirements and overall system design (which defines the architecture of the product). Before the selection of PC/Windows platform another critical skill for Think3 was represented by the capability to elaborate methodologies (sets of rules, tools and standards) which allows a new product to be compatible with different platforms. Not surprisingly, also Geographics considers technical skills (especially the development methodology) as the most critical skill. However, the knowledge of specific applications and the methodology to understand users' needs are also very important in this firm. This reflects the different product configuration between these two firms - while Think3 focuses on stand-alone products in a standard environment (Windows), Geographics tend to develop also 'solutions' that integrate different technologies.

Firms that focuses on complex business solutions (Formula and ESA) consider as most critical for their activity the skills acquired through project management, system engineering and adaptation of the product 'backbone' to the needs of specific industries (banks, manufacturing, public administrations etc.). These skills in Formula are owned by the 'industry managers' while in ESA are embodied in system engineers and project managers. Post-sales consultancy is also very critical skills for Formula.

It is worth noting that the most critical skills represent also the most important repository of tacit knowledge for these firms. In the case of Think3, analysis of users' requirements and system design are not formalised and there are not tools employed for managing and controlling. This firm does not use tools available on the market because they greatly reduce flexibility and do not improve significantly productivity. Most knowledge needed to carry out these activities is then 'brainware', poorly codified and not formalised. Moreover, these activities are conducted in teams and therefore the relevant knowledge is shared among different people. This requires the development of conventions and routines accepted by the members of the organisation or the existence of a 'procedural authority'. Organisational routines and procedural authority are clearly identifiable in system design activities conducted at Think3. System design is normally carried out by distinct teams of 1 to 6 people to which specific tasks are assigned. Each team is led by a 'task leader' while the project overall is under the responsibility of a 'product manager' which reports to the corporate responsible for 'Engineering activities'. Moreover, during the process project leaders discuss their results with senior experts which report to 'product managers'. These 'experts' represent a repository of tacit knowledge for the firm overall. Note that the responsibility of these 'experts' is quite clearly defined and this gives them the authority to solve conflicts and disagreements. The reason why the knowledge of these experts is not codified is that codification probably is not efficient. Codification would require a continuous, rapid revision of documentation that takes time and human resources. The costs are not worth the benefits, according to the top management of this firm, because codification of this knowledge would probably introduce too much rigidity into design activities. Geographics shows a similar pattern of knowledge accumulation and diffusion within the firm. In this particular case, the knowledge of senior personnel (project leaders), who normally spend over 150 days at the customer sites, is transferred by observation and face-to-face discussion.

In the case of Formula and ESA tacit knowledge is mostly concentrated in the experience accumulated in adapting products to the requirements of specific categories of customers, and in post-sales services. In Formula the locus of this experience is dispersed across the firms' subsidiaries and affiliates. The problem is that this fragmentation of knowledge reduces the possibilities of learning from errors and hampers the circulation of information about best practices. For this reason Formula has created an 'horizontal' structure, the 'post-sales technical direction', where different types of experts (software engineers, commercials, consultants and project managers) are pooled together. This structure is aimed at providing an infrastructure for the codification and circulation of 'tacit' knowledge across different divisions and functions. This organisational innovation allows Formula to maintain a 'collective memory' about best practice and errors. Also ESA has tried to circulate the knowledge of its key personnel (senior project managers) within the firm in different ways. First, senior project managers transfer their knowledge to their direct collaborators on a daily basis. Second, the six most senior project managers work in different development centres located in different places. Finally, the know how of these key people is, to some extent, codified by means of formal training. For instance, a key expert in web analysis transfers his knowledge to junior software engineers through training courses. In general, a large share of training time in this firm is devoted to marketing techniques and product management.

In 1995 ESA has introduced a major restructuring of its production activities. The launch of a new product addressing the needs of large firms stimulated a transition from a craft organisation of labour to a more formal and rational division of labour. A new division was introduced to manage the new product and new personnel was hired to introduce new methods of work. This process has resulted in the award of ISO 9000 certification and the adoption of a standardised development environment. The development process relies on automated tools such as Computer Associates' Erwin and Relational Rose's Clear CASE. The firm has first adopted groupware (proprietary) tools for establishing an internal communication environment and has then turned to intranet-internet (open) technologies. These technologies and the standardisation of production have helped to establish an internal system of interfaces and a detailed repository of documentation. This implies high fixed costs but allows an efficient control of operating costs and timing. Moreover, the standardisation of procedures and codification of knowledge allows ESA to decentralise its development activities in three different locations (Bari, Rimini and Florence) each focussing on different development stages or technologies. For instance, the development centre located in Bari has taken the lead in the development of a new module (for e-commerce management) that has been integrated into ESA's ERP products. One of the main reasons that pushed towards knowledge codification in this firm is represented by the high turnover of the personnel. To face in part this problem ESA has accepted the costs of some duplication of efforts. Some critical tasks are carried out by two or more people that work in parallel in different locations and share the same set of information and codified knowledge. Obviously, this implies also a significant effort in terms of system integration. This was the case of the e-commerce module, which was jointly developed by two different R&D centres of the firm. More recently, in order to reduce duplication of efforts and to exploit economies of scale and scope, R&D management activities have been centralised and some tasks have been simplified, although the location of R&D activities remains distributed across different locations. A new staff activity has been introduced to monitor and collect information about different projects. This organisational change has allowed to monitoring each project during its lifecycle, by pointing out best practices and problems that are shared with other projects. This has improved the process of organisational learning and has increased knowledge codification.

The activities which follow the analysis and system design, such as programming or coding, debugging, and testing can be codified more easily. As a matter of fact these activities are very codified and automated in all our case studies. For instance, in Think3 these activities are supported by various CASE tools acquired on the market, such as Rational Rose's Clear CASE, which is employed to manage the source code and the product's releases. However, even standard tools have to be customised and require significant maintenance activity. Often the tools available on the market cannot be adapted to specific development needs and this led Think3 to develop in-house tools used in quality control (these tools are used to manage the tests databases). Other tools developed in-house are used for programming and control of programming activities.

Codification and standardisation of production activities are much less important in firms with simpler products and a smaller organisation. Not surprisingly neither Think3 nor Geographics have adopted standard procedures for quality control (e.g., ISO 9001). These are perceived as a source of rigidity in a context of rapid technical change. In 1998 Thinks3 developed in-house a quality verification methodology with the financial support of the EU. This methodology, called PROVE (quality improvement through VEerification PROcess'), covers testing and inspections. It aims at improving the quality of products by introducing a measurable verification process to be carried out in parallel with the software development cycle. The goal is to make the verification process compatible with other priorities like market and product differentiation. Thinks3 claims that it has demonstrated the feasibility of a structured and quantitative approach to verification in a commercial software producer. This methodology has been experimented thereafter and the following results have been reported by users: a reduction of errors in the development state, a greater reliability of subsequent releases, a larger productivity of verification activities which depends on the use of a reusable set of control procedures, the availability of quantitative data on the correctness of the product which can be analysed continuously. Methods, tools and measures defined by PROVE have been made available to software engineers through the internal Web site. Thinks3 reports significant, albeit not quantifiable, productivity increases from the use of PROVE.

All firms make use of a variety of development languages on request of the clients, including CASE tools and object oriented languages such as Rational Rose and Platinum's toolsets. But, unlike Think3 and Geographics, Formula and ESA have pursued a strategy of quality control based on the adoption of international standards. Their products and programmers have been granted certifications from large software publishers such as Microsoft. Moreover, as mentioned before both Formula and ESA have been given ISO 9001 certification. This difference probably reflects the different size of these firms and the different characteristics of their respective products.

All firms carry out training activities since, as mentioned before, it is difficult to find the right skills on the market. Training in these firms is primarily centred on improving the productivity of new personnel. For instance, in the case of Formula a 'tutor' is assigned to each 'outsider' for one year and in the case of Think3 the senior 'experts' play a crucial role in training activities. These activities are also aimed at distributing the knowledge of specific tasks within the firm. Although this represents a way to protect the firm against the risk of 'competence dispersion' associated with the turnover of key personnel, the main objective of training for these firms is represented by the productivity gain associated to the use of new personnel, codification of procedures and the use of new development technologies. An alternative way to retain (and motivate) skilled personnel is represented by specific incentive mechanisms such as the profit sharing experienced by Geographics.

4.3. The role of collaborative agreements in fostering knowledge codification

As mentioned before, the firms analysed in this paper are all involved in several collaborative linkages with producers of complementary products and services. Most linkages have a purely commercial objective and, as such, they do not require a great deal of technology exchange. Most exchanges concern products (e.g., a module for specific applications) or services (help on-line and training) on the basis of clearly defined contractual arrangements. However, there are examples of commercial linkages, which implies the exchange of technical, partly tacit knowledge. A case is represented by the linkages of Think3 with VARs, particularly in Germany and Taiwan. These linkages are important to 'localise' the products and to meet the requirements of foreign markets. This implies an adaptation of user interfaces, the user messages handling system, and the manuals. A critical dimension of localisation activities is represented by documentation, which is co-ordinated by Think3 with the support of CASE tools. Documentation of localisation is conducted in close collaboration with local VARs and requires frequent contacts through email, telephone, videoconferences and face-to-face meetings.

We have selected two examples of collaborative agreements, which imply a significant exchange of technical knowledge between the partners. In what follows we illustrate the organisation of these linkages, the type of knowledge exchanged, and the instruments to support these exchanges. Finally, we discuss the implications of these linkages for the process of knowledge codification for our two firms.

Think3-Toyota Caelum

In 1997 Think3 and Toyota Caelum Inc. (TCI) started a technical collaboration which is still active. Thinks3 was requested by TCI to disclose its software development methodology and know-how. Subsequently, Think3 has provided TCI with a library of Application Programming Interfaces (APIs). These are programs that can be automatically accessed and used by other applications. Think3 is currently assisting TCI in implementing a CAM (computer aided manufacturing) system which uses Thonk3's APIs and CAD methodology.

Clearly this agreement implies a unidirectional flow of knowledge from Think3 to TCI. The knowledge is transferred in the form of training and documentation. TCI technicians have initially visited Think3's R&D laboratories to observe and absorb know how (which by definition is difficult to document) from their Italian counterparts. The documentation transmitted concerns the methodology and APIs.

The technicians of the two partners meet monthly and communicate more frequently by e-mail and by telephone. About 10 people working at Think3 are involved in this project, including few senior experts who provide their 'tacit' knowledge to assist TCI in the development of its CAM system and take part in the training activities. The large number of people involved in the project is motivated by the extensive ‘division of knowledge’ within this firm. The responsible for this project, who has previously been involved in the PROVE (Process improvement), co-ordinates the documentation, the training and quality controls.

Some benefits of this agreement for Think3 are quite expected. It opened up a window of market opportunities in Japan (over 1,500 licensing contracts have been signed since 1997) and strengthen the international reputation of Think3.

This experience confirms the difficulty to codify and transmit knowledge about the analysis of users’ requirements and the design of system architecture mentioned earlier. In addition, the two partners found their approaches to software development significantly different in terms of procedures, analysis and methodologies. This has affected the training activities and the flows of know-how. However, these difficulties have not blocked the project and the mutual trust between the partners.

A major result of this experience from Think3’s viewpoint is represented by the significant pressure that it has put on the internal organisation in the direction of a greater codification, formalisation and organisation of internal development and documentation. As mentioned before, the process of codification in Think3 had been set in motion before this agreement started. However, this project has impressed an acceleration to the process of formalisation of the product development process on the whole, including requirement’s analysis, feasibility study, specification, coding, testing, releases and documentation. This case then illustrates the interdependence between internal codification of knowledge and external sources of codification.

FLUENT is a joint R&D project supported by the EU (Esprit, IV Framework Programme) which started in 1998 and is due to end by 2001. The project involves nine partners from Italy, Portugal, Hungary, and Greece. Formula took the initiative, contacted the partners and co-ordinates the projects. The partners have previously worked together in other EU projects.

The objective of FLUENT is to develop methods and tools for managing complex logistic flows, occurring in a distributed manufacturing network with multiple plants and co-operating firms. The project aims at providing a common IT infrastructure supporting a unified and generalised model of logistic flows, which draws on Formula’s ERP basic architecture. The system is based on Internet-Intranet technology. The idea is to elaborate a ‘network model’ to represent the position and the relationships of suppliers, manufacturers and customers of specific industries.

The project is based on a clearly defined division of tasks among the partners. The main research tasks are the following: network modelling, planning, definition of development tools and measurement tools. Each partner focuses on a specific pilot-user and is responsible for the development of a component of the whole system. For instance, the Hungarian partner focuses on the ‘Electronic Data Interchange’ (EDI) subsystem, the Greek partner on the ‘Performance Measurement Systems’ and the Portuguese one on the ‘Decision Support System’ (DSS). The specific contribution of Formula consists in providing its competencies and methodologies in the field of ERP.

As far as Formula is concerned, the project involves a team of 20 people, including a co-ordinator (with purely managerial tasks), a project manager (with technical tasks), system analysts, programmers, researchers and marketing competencies.

The partners signed a Consortium agreement, which includes IPR arrangements. The clear technical division of labour among the partners favours the distribution of IPR. On the other hand, the IPR agreement helps to make clear what their expectations are about the commercial benefits of the project and the specific technical contribution of each partner.

Each task is under the responsibility of a partner, which collects the inputs from the partners and circulates documents, drafts and other materials.

The knowledge accumulated during the phases of user’s requirement analysis, system design and detailed design is fully shared among the partners (and pilot-users) through circulation of the relevant documentation. The IPR of this documentation is also shared. The information exchanged is in the form of executable (object) codes, documentation, images, and records of activities and decisions taken during the meetings.

The design of single components or modules, including source codes and detailed product specifications, is not shared among the partners. During the development of these components there are technical issues (such as development and testing tools) discussed and shared among the partners. This is because, according to Formula, technical knowledge is codified and recorded in documents e.g., executable or object code). As a consequence the partners have never disagreed on technical issues so far. By contrast, the knowledge of specific applications (such as the development of a DSS for a given category of users) is primarily based on experience and as such it is more difficult to codify and transmit across firms. And this requires more discussion and personal contacts among the partners.

Overall the knowledge and information produced in this project are circulated through different channels. Plenary face-to-face meetings are organised every three months to discuss the evolution of each task and to make critical decisions. Email contacts occur on a daily basis among the partners. These contacts involve different levels of the organisations – the project leaders, system designers, researchers and the marketing personnel involved in this project. Videoconferences are organised to face specific problems such as the integration between different modules or packages. In this case, the co-ordinators and the technical leaders of the involved parties communicate via videoconference. Another channel for the circulation of knowledge and information is represented by a database which collects material, documents, scientific and technical papers produced within or outside the project, including technical or market evaluations of specific technologies and tools provided by sector analysts and other external experts.

In order to favour the development of a shared language among the partners, Formula has organised a training workshop for its partners at the beginning of the project. The experience accumulated during this workshop has been codified and circulated among the partners.

The procedures required by the EU and the number of partners have imposed a significant codification of procedures and knowledge produced during the project. Through this experience Formula has improved its ability to articulate the knowledge produced in different stages of the development process. The main benefit is represented by improved project management capabilities.

This project has also brought about new technical knowledge, especially in the field of EDI and communication protocols, requirements' analysis and integration in other ERP environments (such as SAP's and Baan's).

5. Conclusions

This paper analyses the codification of knowledge in the software industry and focuses on different economic incentives to codification, including the need to improve the productivity and quality of software production and the networks of inter-firm alliances. The experiences of four Italian software firms are examined. The paper compares their capabilities, the main sources of tacit knowledge, their specific incentives to invest in codified knowledge, the use of formal development methodologies and quality control systems adopted. Finally, the paper analyses two distinct technological collaborations that two of these firms have recently established.

Our sample includes firms that differ on two main grounds – the type of product and the complexity (size) of organisation. Two firms (Think3 and Geographics) specialise in 'simple' or stand alone products (CAD and GIS packages) and have relatively small and simple organisations. The other two firms (Formula and ESA Software) specialise in complex products (ERP solutions) and have larger, multi-plant complex organisations.

We found significant differences between these two categories of firms. There are differences between our firms about the perception of which type of knowledge is tacit and which one is not. The first category of firms tend to consider the analysis of user's requirements and system design as the main source of tacit, uncodified knowledge. Instead, for the second category of firm the main *source of tacit knowledge is represented by the experience accumulated in adapting its platform to specific users' needs, project management skills and post-sales services.

Another difference between these firms is represented by the different incentives to codify tacit knowledge. For instance, Think3 tends to perceive codification of knowledge of its senior experts in system design and the compliance with international quality standards (e.g., ISO 9001) as a source of organisational rigidity that can hamper innovation. For this reason, it has developed in-house a quality assurance methodology. Instead, Formula and ESA have set up internal 'horizontal' structures where different skills (technical, commercial, project management and consulting) are pooled together with the aim of improving the circulation of knowledge and the codification of best practices across different divisions and locations. Moreover, they have been given ISO certification. These differences point out the importance of product and organisational complexity in affecting the incentives to invest in knowledge codification.

The analysis of two cases of technological collaborations shows that these collaborations imply a significant deal of knowledge exchange among the partners. The type of knowledge flowing between the partners is both tacit (know how and methods transmitted by training and direct visits to the partners' sites) and codified (written documents, object codes, draws, images, and tools).

A clear division of labour and complementarity among the partners, and previous investments in the codification of software development activities seems to be an important condition for the success of these technical alliances. Moreover, the collaborations have impressed acceleration to the process of internal codification. Finally, different stages of the production activities have been affected by these collaborative agreements, from the analysis of users' requirements to documentation and project management.

Appendix - The case studies

Think3 (formerly CAD Lab) is a medium sized (250 employees) Italian-US producer of three-dimensional CAD packages. It was founded in 1979 as a spin-off of the Engineering Department of the University of Bologna with the aim to offer software development, data elaboration and testing services to customers operating in the construction industry. After two years the founders decided to focus on packaged software, two-dimensional CAD, and entered a partnership with Hewlett Packard (HP). During the 1980s Think3 expanded its installed base of customers in the domestic market thanks to new commercial alliances with OEM and VAR firms. More importantly, during the 1980s Think3 moved from a single platform approach (HP) towards a multi-platform approach (Apollo Computers, Digital, IBM PC and HP). In Italy Think3 was a first mover in developing CAD packages for a multiplatform environment. During the 1990s Think3 has invested in products for the PC/Windows platform, which now supports about 80% of its products (the remaining 20% being accounted for by Unix systems). In the same period Think3 pursued a strategy of vertical integration of downstream activities by acquiring two VARs in Italy. It also started to penetrate the international markets – France, Germany, Spain, Belgium, South Korea, Taiwan and Japan. This strategy has been pursued through alliances with local VARs or direct linkages with customers. In 1997 Think3 tried to enter the US market. To this purpose, Think3 set up an agreement with US Venture Partners, a venture capitalist based in Palo Alto, California, which poured fresh money into Think3 and in return it took a minority stake. In 1998 Intel entered into the venture taking a minority stake of Think3. In order to give credibility to its strategy of penetration of the US market Think3 moved its HQ to California. The marketing and sales strategy in the US is centred on e-commerce with the aim of overcoming the difficulties encountered with local VARs.

Think3 is currently a small multinational corporation that draws over 50% of its revenues from abroad. The bulk of its R&D activities is located in Italy (Bologna and Pesaro). Smaller R&D centres in Santa Clara, CA, Boston, and Aix-en-Provence have been set up to exploit regional technological spillovers. The entry into the US market has helped this company to establish new linkages with foreign partners. In particular, Think3 has recently set up a technological agreement with Toyota-Caleum that has opened up an important window of opportunity in the Japanese market.

Geographics

This firm is part of a larger group of software firms (Gruppo Zuffellato, based in Ferrara). The group was founded in 1975 and employs over 100 software engineers and programmers. Moreover, it relies on a network of external consultants for the development of customised solutions and customer services. The group is a reseller of standard hardware and software products, and specialises in the development of GIS (geographics information systems). Geographics is a small firm (ten employees) which focuses on GIS. It has been founded in 1993 and it develops 'vertical applications' for specific customers (mostly local public administrations). Its applications draw on legacy GIS technologies such as ArcView, MapObjects, ArcCAD and AutoCAD. It has established close commercial links with ESRI (US), the provider of the market standard technology (viewers, base engines). About 25% of Geographics' revenues come from the production of niche packages - e.g., GeoPRG and GeoAQA for the development and management of territorial networks (e.g., water) and geographical databases. Geographics offers also installation, training, management and technical support services to customers.

The partnership with ESRI has become closer and closer over time. Today, Geographics is an alpha and beta tester of ESRI technology. Therefore, the technological content of this partnership is limited.

We interviewed one of the founders of Geographics.

With 700 employees Formula is the largest Italian producer of Enterprise Resource Planning (ERP) packages. It also draws over 50% of its revenues from services (resale of third party packages, training and consulting). Unlike Think3, Formula offers multi-platform products.

This company was founded in 1972 as a distributor of foreign application software (such as Cincom's databases management system). In the 1980s Formula started to produce software products by its own and expanded its activities in Italy first and abroad thereafter. Today Formula has its HQ in Torino, Italy, and fully owned subsidiaries in Italy (Bologna and Bari), France and Spain, and affiliates in Venezuela and Mexico. In 1997 Formula took out a 29% shareholding in a new US venture, Innovative Process Management, Inc. (process management tools) and more recently established an office in Boston. In 1998 took 100% of SAS Srl, an Italian firm specialised in application software for logistics.

In 1997 Formula obtained an admission for trading on the EASDAQ market in Brussels and since then has tripled its capital.

Over time Formula has established a wide network of contractual linkages with different types of partners. These include 70-80 subcontractors (small software developers and consultants), about 15 VARs, few system integrators such as Ernst&Young and KPMG, over 7 producers of complementary software packages which are integrated into Formula main product – e.g., maintenance management systems, manufacturing scheduling, budgeting, and planning systems, and clients portfolio management systems. It has also set up commercial agreements with major software publishers such as Microsoft, Business Objects and Oracle and computers manufacturers such as IBM and Sun.

Its presence in foreign markets is still limited (with about 3 per cent of total revenues arising from abroad). However, it has recently created an international division to expand its foreign activities in the field of resource management applications.

ESA software

This firm was founded in 1982. It specialises in ERP (enterprise resource planning) software for SMEs. ESA focuses on two main product families, ESATTO and EXPLOIT (ERP solutions for SMEs and large firms respectively). It employs over 320 people in six subsidiaries and two R&D centres located in different Italian regions. It sells its products to about 350,000 firms in Italy and has about 500 partners specialised in the distribution, installation and customers' technical support. This network of partners is segmented into different layers. The first layer is represented by 11 partners where ESA has a minority stake and a group of 'core partners' (master senior partners) who are values added resellers and system integrators. The second layer is represented by firms (junior partners) that offer a limited array of services to customers (e.g., installation and maintenance). The lower end of the network is accounted for by partners that offer no services (pure dealers) and have only indirect links with ESA (through the upper level partners). The 'core' partners (about 10%) participate in ESA marketing strategy (e.g. product policy) while the rest of the partners is only involved in short term decisions. Moreover, since 1989 ESA has started a commercial collaboration with Microsoft (development of business solutions for the Windows environment).

In Italy ESA is the market leader in ERP solutions for SMEs. ESA will be quoted in the Italian 'new' stock market by the mid of 2001.

References

Arora, A. and Gambardella, A. (1994), ‘The Changing Technology of Technological Change: General and Abstract Knowledge and the Division of Innovative Labour’, *Research Policy*, 23, 523–32.

Arrow, J.K. (1962), ‘The Economic Implications of Learning by Doing’, *Review of Economic Studies*, 29, pp. 155–73.

Balconi, M. (2001), 'Tacitness, Codification of Technological Knowledge and the Organisation of Industry', *Dipartimento di Economia Politica e Metodi Quantitativi*, Università di Pavia, mimeo.

Bohem, B.W. (1981), *Software Engineering Economics*, Prentice-Hall, Inc., Edgewood Cliffs, New Jersey.

Chandler, A., Jr. (1990), *Scale and Scope. The Dynamics of Industrial Capitalism*, Harvard Business Press, Cambridge MA.

Cohen, W. and Levinthal, D. (1989), ‘Innovation and Learning: The Two Faces of R&D’, *The Economic Journal*, 99, pp. 569–96.

Combs, J. G. and Ketchen, D.G.Jr. (1999), 'Explaining Inter-firm Cupertino and Performance: toward a reconciliation of predictions from the resource based view and organisational economics', *Strategic Management Journal,* 20, pp. 867-888.

Cowan, R. and Foray, D. (1997), 'The Economics of Codification and the Diffusion of Knowledge', *Industrial and Corporate Change*, 6, 5, pp. 595-621.

Cowan, R., David, P. A. and Foray, D. (1998), 'The Explicit Economics of Knowledge Codification and Tacitness', paper prepared for the TIPIK workshop, Paris, December, 4-5, 1998. *Industrial and Corporate Change, special Issue.*

Cusumano, M. (1992), ‘Shifting Economies: From Craft Production to Flexible Systems and Software Factories’, *Research Policy*, 21, pp. 453–80.

Dosi, G. (1996), 'The Contribution of Economic Theory in the understanding of a Knowledge based Economy', in D. Foray and B.A. Lundvall (eds.), *Employment and Growth in the Knowledge based Economy”*, OECD, Paris.

EC (1998), *Job Opportunities in the Information Society*, European Commission, Brussells.

Garber, J.H. (1993), ‘Working faster’, *Forbes*, April 12, p. 110.

Irvine, J., Martin, B. and Isard, P. (1990), *Investing in the Future: An International Comparison of Government Funding of Academic and Related Research*, Edward Elgar, Adelshot.

Jensen, M. C. and Meckling, W. (1995), 'Specific and general knowledge, and organisational structure', *Journal of Applied Corporate Finance,* 8, 2, pp. 4-18.

Levin, R.C., Klevorick, A.K., Nelson, R.R. and Winter, S.G. (1987), ‘Appropriating the Returns from Industrial Research and Development’, *Brookings Papers on Economic Activity*, No. 13.

Lindholm, E. (1994), ‘Snap-On Code’, *Datamation*, 1, February, 1994, pp. 63–5

Lundvall, B.A. and Johnson, B. (1994), 'The learning economy', *Journal of Industry Studies*, 1, pp. 23-42.

Malerba, F. and Orsenigo, L. (1996), ‘Schumpeterian Patterns of Innovation are Technology–Specific’, *Research Policy*, 25 (3), 451–78.

Malerba, F. and Torrisi, S. (1996), ‘The Dynamics of Market Structure and Innovation in the Western European Software Industry’, in Mowery, D. (ed.), *The International Computer Software Industry*, Oxford University Press, pp. 165–96.

Nelson, R.R. (1962), ‘The Link between Science and Invention: the Case of the Transistor’, in

Nelson, R. (ed.), *The Rate and Direction of Inventive Activity*, Princeton University Press, New Jersey.

Norton H. (1991), Informatics in Europe, NCC Blackwell, Manchester.

OECD (1989), The Internationalisation of Software and Computer Services, ICCP series n. 17, Paris.

OECD (1997), *Information Technology Outlook*, Organisation for Economic Co-operation and Development, Paris.

Teece, D. J. (1986), “Profiting from Technological Innovation: implications for integration, collaboration, licensing and public policy”, *Research Policy,* 15, pp. 286-305.

Torrisi, S. (1998), *Industrial Organisation and Innovation. An International Study of the software industry*, E. Elgar, Cheltenham, UK.

Torrisi, S. (1999), ‘Firm Specialisation and Growth. A Study of the European Software Industry’, in Gambardella, A. and Malerba, F. (eds.), *The Organisation of Economic Innovation in Europe*,

Cambridge University Press, Cambridge.

Winter, S.G. (1987), ‘Knowledge and Competence as Strategic Assets’, in Teece, D.J. (ed.), *The Competitive Challenge: Strategies for Industrial Innovation and Renewal*, Ballinger, Cambridge, Mass., pp. 159–84.

Winter, S.G. (1993), 'On Coase, Competence and Corporation', in Williamson, O.E. and Winter, S.G. (eds.) *The Nature of the Firms,* Oxford University Press, New York, pp. 179-195.

Notes

[1] We thank the participants to the EC-TSER Project ‘Technology And Infrastructures Policy in The Knowledge-Based Economy - The Impact of The Tendency Towards Codification of Knowledge (TIPIK) for comments to earlier drafts of this paper.

[2] See Balconi (2001) for an analysis of the balance between tacit and codified knowledge in different industries.

[3] It is worth noting that inter-firm networks reduce the coordination problems arising from the externalities produced by investments in knowledge codification.

[4] Another incentive to knolwedge codification is represented by the opportunity (or the need) to make a firm's products compatible with complementary products produced by others (e.g., application solutions or applications development tools).