WORKING PAPER NO.246

Novel Organization form as a Growth Driver

By

Sourav Mukherji
&
J Ramachandran

June 2006

Please address all your correspondence to:

S Mukherji
Assistant Professor of Organisation Behaviour
Indian Institute of Management Bangalore
Bannerghatta Road
Bangalore 560 076, India
Email : souravm@iimb.ernet.in
Phone : 080-26993145
Fax : 080-26584050

J.Ramachandran
BOC Professor of business Policy
Indian Institute of Management Bangalore
Bannerghatta Road
Bangalore 560 076, India
Email : jram@iimb.ernet.in
Phone : 080-26993080
Fax : 080-26584050

NOVEL ORGANIZATION FORM AS A GROWTH DRIVER

Sourav Mukherji, Indian Institute of Management Bangalore; souravm@iimb.ernet.in

J Ramachandran, Indian Institute of Management Bangalore; jram@iimb.ernet.in

Abstract

Scholars have conceptualized the competitive landscape confronting modern organizations as business ecosystems. An ecosystem is well suited to describe the complex interdependencies that exist among specialist organizations operating in a dynamic environment. Is a business ecosystem similar to a network form of organization? We do not think so. Based on our research on the relational network that exists between Indian software service organizations and their clients, we distill two features that distinguish a business ecosystem from traditional networks. These are (1) the ability of the relationship to simultaneously accommodate collaboration – a definitive feature of networks (Powell, 1990) with competition, a definitive feature of markets and (2) the involuntary nature of the interdependencies that prevail among the network players. We present our arguments through description and analysis of a novel organization form witnessed in the Indian software services industry – the Offshore Development Centre. In the process, we locate the competitive advantage of Indian software services organizations in their ability to create innovative organization structures and processes that enabled them to achieve rapid scaling even in the face of business volatility and technology churn.

Introduction : Network form of Organizations

The twin forces of globalization and technological advancement have led to the emergence of a new organizational form called the network organization, comprising clusters of business units coordinated by market mechanisms (Cravens et al, 1996). Firms till 1970s and 80s were designed to gain economies of scale and advantages of experience through central planning and control systems. Vertically integrated firms with layers of hierarchy were suited to serve growing needs for efficiently produced goods, both at the domestic and international markets. However, competitive pressures over the last few decades resulted in a paradigm shift in organization design. Organizations needed to be both effective and efficient – adapt with increasing speed to market pressures and competitors' innovations while simultaneously control or lower product and service costs. This resulted in firm level specialization - firms performing only those functions within the firm boundary for which the firm has or can develop expertise and skills and outsource other activities that can be performed quicker, more effectively or at lower costs by others. This philosophy led to the formation of network structures, whereby firms engaged in specialized activities interacted with one another, causing market forces to bear on each element of the product or service value chain (Snow et al, 1992). The result was coupling of market-based efficiencies with specialization driven effectiveness.

However, a network structure is more than multiple buyer-supply relationships. Podolny & Page (1998) define a network form of organization as "*any collection of actors (N > 2) that pursue repeated, enduring exchange relations with one another and, at the same time, lack a legitimate*

organizational authority to arbitrate and resolve disputes that may arise during the exchange." Members of network are not completely dependent on one another (Jarillo, 1988) because, unless there is independence along certain dimensions of the relationship, the relationship is one of actual or quasi-integration. As an organizational form, some scholars viewed the network as a hybrid of markets and hierarchies. Coase (1937) had conceptualized firms and markets as alternate means for organizing economic transactions. Williamson (1975, 1985) argued that transactions that involve uncertainty about their outcome, recur frequently and require substantial 'transaction specific investment' are more likely to take place within hierarchically organized firms. However, the distinction between firms and markets get blurred when firms are found to be engaging in forms of collaboration that resemble neither the arm's length market contract, nor vertical integration. Thus, scholars assumed the view that economic exchange can be arrayed in a continuum like fashion, with discrete market like transactions located at one end and highly centralized firms on the other. In between, there are various intermediatory or hybrid forms, all of which can be classified as networks. At the market end of the continuum, where prices capture all the relevant information necessary for exchange, one finds putting out systems, various forms of repeated trading, quasi-firms and subcontracting arrangements. Towards the hierarchy end, there are franchising, joint ventures, decentralized profit centers and matrix management.

Powell (1990) however disagrees with the hybrid notion and views networks as 'pure' form, because of its alternate governance mechanism embedded in reciprocity and collaboration, which is neither present in markets nor in hierarchies. Podolny and Page (1998) push this thought further and conceptualize both 'markets' and 'hierarchies' as special form of networks. However, scholars across the divide agree that are a wide variety of organizational relationships that can be classified under the network form. In their review of network forms, Snow et. al (1992) describe three kinds of network structures – internal, stable and dynamic, depending on the degree of asset ownership by the 'lead firm'. While the 'internal network' operates within the boundary of a single organization, a 'stable network' employs partial outsourcing by the 'lead firm' that owns majority of the assets. In 'dynamic networks', outsourcing is extensive where the lead firm identifies and assembles assets that are mostly owned by other organizations. In its limiting case, the lead firm becomes a broker among independent specialists.

Likewise, Cravens et al (1996) distinguish network forms based on whether the relationship between network members is transactional or collaborative. A transaction based network, operating in turbulent environment gives rise to 'hollow networks', which is akin to Snow et al's (1992) dynamic network or Achrol's (1991) 'marketing exchange company' that resembles a brokerage or clearinghouse. In stable environment, a transaction-based relationship gives rise to 'value-added networks' where the lead organization utilizes a network of suppliers, but maintains

substantial internal operations. Inter-organizational link characterized by collaboration, where the network members need to make asset specific investments, give rise to 'flexible networks'. Such networks are characterized by a complex production and distribution process and operate in volatile environment, necessitating greater coordination and collaboration between the lead organizations and the nodes. Quinn (1992) calls this as a 'spider web' network where there is close communication links between the nodes and the coordinating company, although each node operates on an independent basis by performing the tasks agreed with the hub company.

Moore's (1993) conceptualization of a 'business ecosystem', which was further extended by Iansiti and Levin (2004) emerged independently of debates around the network form of organization. They argued that an ecosystem analogy is well suited to describe the complex interdependencies that exist among specialist organizations operating in a dynamic business environment. They criticize the focus of existing management literature on tightly coupled systems and draw attention to 'huge, unfocussed, unbounded amorphous and constantly evolving networks like the Internet'. They site examples of the ecosystems evolving around Wal Mart, Microsoft and Intel, who create powerful platforms on which other members of the ecosystem can built their products and services. Platform creators or 'keystones' differ with the vertically integrated firm, the 'dominator', because they share value with other members of the ecosystem in order to enhance the health of the entire ecosystem. While this long-term orientation is similar to one observed in relational contracts described by network researchers, the business ecosystem does not lay explicit stress on collaboration, norms of reciprocity or building of trust, which are considered essential for network members. Akin to the conceptualization of 'community ecology ' (Astley, 1985), the focus of business ecosystem is the diversity of organizations and the complex interdependencies that exist among them.

While the notion of business ecosystem can be subsumed under a network form, we believe that the power of this conception lies in its ability to distinguish itself from extant literature on networks. Based on our research on the relational network that exists between Indian software service organizations and their clients, we distill two features that distinguish a business ecosystem from traditional networks. These are (1) the ability of the relationship to simultaneously accommodate collaboration – a definitive feature of networks (Powell, 1990) with competition, a definitive feature of markets and (2) the involuntary nature of the interdependencies that prevail among the network players. We present our arguments through description and analysis of a novel organization form witnessed in the Indian software services industry.

Indian Software Industry Evolution

Over the last decade and a half, increasing use of information and communication technology in contemporary business and society has led to high demand for software applications. However, the developed nations, where such demand is most pronounced, faced shortage in the availability of software developers necessary for the production of software applications. India, with its large pool of English speaking engineering graduates became an ideal destination for sourcing skilled software professionals. Wage arbitrage that existed between India and the developed nations made such sourcing even more attractive.

Several Indian software service organizations leveraged this opportunity and started acting as suppliers of software professionals to their clients in the developed nations. Demand for software skills was further increased during the later half of nineties on realization of the 'year 2000' (Y2K) problem in mission critical and transaction intensive software applications deployed all over the world. The dominant model for meeting this demand was that of resource augmentation, where the software was developed on client premises or 'on-shore'. Indian software professionals worked as contractors, delivering specific tasks while the software projects were conceived, designed and managed by the client organization. However, Indian software services firms realized that resource augmentation projects were vulnerable to competition from other developing nations with comparable wage levels. There was also possibility of greater cost savings if projects were done off-shore, in India, and such savings would translate into greater profitability for both the client and the supplier. Thus, Indian software firms made conscious efforts to move their onsite engagements to off-shore locations, and to reduce their exposure to non-sustainable opportunities like Y2K (Arora et al, 2001).

However, outsourcing to remote locations is a risky proposition for client organizations because unlike tasks that are done within the boundaries of an organization, outsourced tasks cannot be monitored or controlled from close quarters. Neither are the competences of employees who are engaged in doing the tasks well known. The client organization can hedge against these risks by drawing suitable service level contracts. For a physical product that can be tangibly measured, it is relatively easy to lay down the delivery parameters in terms of physical characteristics. This is inherently difficult for a knowledge intensive product like software that lacks measurable characteristics, which can be used as a proxy for its quality and functionality. This problem is compounded by the fact that a software product, during its development phase, is prone to considerable amount of requirement churn (Brooks, 1987). It is very difficult to specify at the outset all the functionalities that the software product will need to have to address the business problem or customer need. Thus, while offshore outsourcing was an attractive commercial proposition, it was fraught with risk that could not have been overcome with traditional buyer-

supplier contracts. This migration from an on-shore model to that of off-shore outsourcing was made possible, to a large extent, by instituting unique governance mechanisms for project management that mitigated the risks for both the client and the service provider. Central to such governance mechanism was Software Engineering Institute's Capability Maturity Model (SEICMM) that enabled exchange of rich information between the client and the service provider at every stage of software development.

Governance Mechanisms Enabling Behaviour Control

Feasibility of outsourcing, especially to remote locations, is contingent upon the extent of control that the client can employ over the service provider. When clients outsource, they become vulnerable to risks of poor or non-performance by the supplier. Such risks increase if the product is technologically complex or the production process is lengthy, as is the case of most software applications, since it becomes difficult for the client to quickly switch to alternate source of supply. While the damages incurred because of poor or non-performance by the supplier can be compensated by resorting to contractual clauses (on penalty), most clients would like to avoid such damages, especially if the client's internal processes are in someway, interdependent on the performance of the outsourced product or process. In outsourced projects, the client can impose two kinds of control systems on the suppliers - 'behaviour' control and 'output' control (Govindrajan and Fisher 1990). In order to determine the suitability of each form for a particular situation, one needs to consider two aspects of the production process, namely 'task programmability' and 'outcome measurability'. A task is programmable if there exists a clear definition of the behaviour needed to perform the same. When knowledge about the transformation process is high, it is possible to explicitly define the appropriate behaviour necessary to achieve the task objective and behaviour control becomes appropriate. Output measurability refers to an output's susceptibility to reliable and valid measurement. If outputs are unobserved or unreliable as means of measuring effort, output control cannot be employed and clients need to resort to behaviour control.

Output measurability is inherently difficult for a knowledge intensive product like software applications because unlike a physical product, a software application lacks measurable characteristics that can be used as a proxy for its quality and functionality. However, modern methods of software testing, simulation and function point analyses have made 'outcome measurability' easier for software applications, implying that it is possible for the clients to impose outcome control over the production process. Agency theory (Baiman 1982) informs that when the control of agents (suppliers, in this case) is based on outcomes, the risk of production is shifted to the agent. The agents would be averse to bearing such risks under conditions of uncertainty because as a consequence of uncertainty, there might be other factors apart from

behaviour of agents that would influence the outcome. Under such conditions, since superior efforts can lead to poor performance and vice versa, it is better for principles to employ behaviour control, provided they are able to get information about the agent's behaviour (Govindrajan & Fisher, 1990).

'Requirement specification' for software development, especially in the case of Indian software service firms who are distant from client markets, is an iterative process. Most software projects suffer from 'requirement churn', where design changes are introduced on client's request after commencement of the production process. This introduces considerable uncertainty into the process in terms of effort and schedule estimation. Thus, even though modern methods of software testing and analysis could have enabled outcome control in outsourced software development contracts, it would have put the service provider at a disadvantage because of the risks associated with requirement churn and uncertainty. These risks could only be mitigated by shifting the risk to the client provided there are adequate mechanisms for information exchange between the client and the supplier, which would enable behaviour control. This was made possible by evolving suitable governance mechanisms of project management and extensively deploying principles of software engineering.

The development of software started as craft form of production, dependent on the brilliance and creativity of individual programmers. However, rigorous application of software engineering principles transformed the nature of software production, converting the 'art' of writing software programmes into an engineering discipline, characterized by a high degree of process standardization (Cusumano, 1992). The capability maturity model (CMM) developed by Software Engineering Institute (SEI) of Carnegie Mellon University, an evolutionary framework for improvement of software development, evolved as an international standard for software service organizations. By carefully managing requirements, using formal inspections on design and codes and systematically practicing risk management, SEI-CMM programmes improved the ability of organizations to meet goals for cost, schedule, functionality and product quality. Just as Taylorian principles of scientific management defined the 'one-best-way' for carrying out activities on the shop floor, software engineering principles laid down the 'one-best-way' of developing large-scale software applications and made software development more 'programmable' than it was in the erstwhile craft mode of production.

SEI CMM framework, through rigorous standardization of various process parameters, made it feasible to remotely monitor downstream activities of software development. Within the framework, it was possible to monitor every phase of software project in terms of productivity and quality as well as track its progress against accepted benchmarks. Such process control

mechanisms specified in SEI CMM were supplemented with weekly and monthly reviews where managers in charge of off-shore projects would present detailed quality and schedule information using standardized tools like Project Tracking and Oversight Charts (PTO) and Quality charts (Ramachandran & Dikshit, 2002). Formal management processes like 'Change Control Boards' were instituted to handle specification changes that needed to be incorporated after commencement of projects. The majority of contracts between the client and the service provider were in the nature of 'time and material', where the service providers agreed to charge the client based on the utilization rate rather than quoting a 'fixed price' for the entire project, a manifestation of outcome control. It was considered fair that the client bears the risk of requirement churn, since it is the client who controls the upstream processes that influence requirements.

Process standardization, thus, enabled offshore outsourcing. The next challenge for the Indian software exports firms was to scale the business. This implied either getting more projects from the same client or getting new projects from other clients. For the clients, it was advantageous to continue with the same supplier. On one hand, it reduced transaction costs of search, writing and enforcing contracts. But more importantly, giving projects to the same supplier ensured knowledge continuity across related software projects, which directly contributed to productivity benefits. However, giving a series of projects to the same supplier increased risks of knowledge spillover for the client, i.e., leakage of confidential information to competitors through the supplier. Such risks would also put a limit to the number of other clients that the service provider could work with – the second growth option for the software exporter. The software service firms overcame these problems through a structural innovation in the form of the offshore development center (ODC).

The Offshore Development Centre

An ODC is a dedicated facility for a specific client located within the organizational boundary of the service provider. It is governed by a contractual agreement between the service provider and the client whereby the client commits to engaging the services of the supplier over a specified time horizon that substantially exceeds the duration of a single project. In return, the service provider ensures that a team of software developers and other resources are earmarked for the client and the intellectual property generated out of such projects is completely protected, even from the rest of the organization. Thus, an ODC is like an island that is sealed off from the rest of the organization in terms of information and knowledge flows. This enables the service provider to get into multiple such contracts with several client organizations, many of who might be competitors of one another.

ODCs interlock the clients and service providers into a many-to-many relational network. Most large Indian software service providers like Wipro, Infosys and TCS have multiple ODC's within their organizations. Likewise, multinational organizations like Cisco, TI and Nortel have their ODC's in several of the Indian software services organizations, sometimes over and above having their fully owned subsidiaries. Estimates suggest that close to 80% of revenues earned by large software service providers are from 'repeat customers' and it is likely that 75-80% of work done for such repeat customers would be from ODC's.

Our discussions with managers of software services firms indicate that ODC's are a culmination of an evolving relationship between the client and the service provider. While the initial engagement with the client often started as a resource augmentation project that was carried 'on-site', the service providers used such engagements to build client confidence in their capabilities. On site engagement also provided them with the opportunity to tap into the client organization network, develop a better understanding of client requirements and convince the client to shift some of the on-site engagements to off-site, i.e., on to the premises of the software services provider in India. Since off-site rates are typically one-third that of on-site rates (Arora et al 2001), this was attractive for the clients, provided they were convinced about the capability of the service provider to deliver from remote location. Thus, on-site projects served like 'prototypes' for customers, such that they can build confidence about outsourcing projects to Indian services providers.

Most off-site engagements started with a specific project contract - delivering a software solution within a finite time at a given cost. Once the project was completed, the relationship would come to an end and the service provider would again have to make efforts to get new assignment from the client, often through a process of competitive bidding. However, there are significant benefits if the client is able to work with the same supplier. This is because software is a knowledge intensive product. The functionality of a software solution cannot be completely measured in terms of tangible parameters like lines of code. Moreover, the logic underlying a piece of software code is complex and tacit in nature, often remaining embedded with the members of the development team. Thus, there are significant productivity advantages if the same team of developers is assigned to develop or maintain a piece of software with which they have had prior familiarity. This is because there exists knowledge interdependencies between several generations of software products (e.g., drivers for a certain family of laser printers) between related technology (e.g., different flavours of UNIX operating such as IBM-RISC or Sun-Solaris) or related industry domain (e.g., financial services). This implies that clients would benefit if they outsource projects concerning similar software, to the same team of developers, or to the same service provider who has migrated up the learning curve of the particular software solution.

Dedicated offshore development centers were the perfect structural vehicle for achieving such knowledge continuity. The governance contract ensured the availability of a dedicated team of software developers. A master service agreement was drawn between the client and the service provider that laid down the rules of engagement in terms of the quantity and commercial rates of resources that were to be deployed at the center. ODC managers whom we interviewed indicated that often the clients paid for software developers located in the ODC at agreed upon billing rates even if those developers were not deployed on any software projects, just as organizations would pay salaries to their employees in anticipation of future work, even if there was no work at present. Over and above infrastructure costs, the clients often bore the training and development costs of the ODC members and assigned them email identities that had the client organization as their domain names. It was also usual for the ODC to feature in the organigram of the client organization – all of which contributed to the ODC being viewed as an extension of the client organization, rather than being part of the service providing organization.

In one of the software service organization, the internal structure of the ODC was modified to mirror the structure of the client organization in order to bring about greater communication and better transparency in reporting relationships. It is interesting to note that most of these operational and structural characteristics were not mandated by the master service agreement that was drawn between the client and the service provider. Rather, they evolved as a consequence of the rich relationship and tacit understanding that developed between the two parties. There are several instances where the service provider shifted manpower resources from one ODC to another and received implicit approval from the client, even though the formal contract explicitly prohibited such transfers. Industry is also replete with instances where the client gave advance notice to the service provider before ramping down the size of the ODC, so that the service provider could make necessary adjustments for accommodating the excess resources in other projects or ODC's, even though the client had no such contractual obligations.

The ODC as a Structural Innovation

Scholars have often debated on what can be suitable organizational principles – structures, processes and policies, for delivering high quality service. Most of the insights till date have been about having a compelling organizational vision, responsive processes and employee oriented policies. Extant literature has not identified any novel organizational structure that would suit service organizations, implicitly indicating that existing structural forms like the U form, the M form, the bureaucratic or the adhocratic can be suitably applied to service organizations depending upon contingency factors of strategy, size and technology. While this might still hold true, the ODC seems to be a novel organizational structure that has evolved among the Indian

software service organizations. Its suitability for delivering knowledge intensive services combined with its amenability to rapid scaling possibly makes it as crucial a structural innovation as was the multidivisional form for reviving competitiveness of the American manufacturing industry in the later half of last century.

In its simplistic form, outsourcing of the routine components of software development activity is a market-based transaction. They do not incur significant transaction specific investments. As was mentioned before, the inherent difficulties associated with writing a market contract for an intangible product were largely overcome by embracing the process control mechanisms of SEI-CMM certification and a continuous and intensive review process. However, arms length transactions were inadequate to deal with knowledge continuity that provides significant productivity benefits. Moreover, like most knowledge intensive products that are transacted across markets, software service outsourcing incurs the risk of knowledge spillover (Nooteboom, 2002). Two parties who are transacting a knowledge intensive asset makes investments in bridging cognitive distance, by building appropriate absorptive capacity (Cohen and Levinthal, 1990) and capacity to make oneself understood by the partner. Such investments can create the risks of spillover of critical knowledge to the competitor. In case of software services, the clients run the risk of sensitive knowledge spillovers to competitors via the software service provider. While confidentiality clauses in the contract can be a mechanism to mitigate such risks, such clauses are difficult to enforce because of the tacit nature of software knowledge that remains embedded with the team of software developers. Thus beyond a certain scale, the perceived risks of knowledge spillover and the need to maintain knowledge continuity would have made a hierarchical solution more suitable than transactions made across the markets. If clients wanted to get the benefit of lower labour costs prevalent in India, it would have suited them to create Indian subsidiaries - the option of offshore-insourcing as opposed to the prevalent model of offshore outsourcing.

However, internalizing these operations had their drawbacks. Dynamic industries like information technology and communication are characterized by widespread demand fluctuations even in the short and medium term (Bourgeois & Eisenhardt, 1988). To effectively deal with such demand fluctuations, a hierarchical solution had to be flexible enough to scale up or down rapidly - an inherently difficult proposition in a manpower intensive business. Indian software service firms, because of their diversified portfolio of clients, were far better suited to mitigate risks of demand volatility than their clients (Mukherji & Ramachandran, 2004). This, along with their superior process capabilities in software development and lower costs structures arising out of better knowledge of the input market, outsourcing looked to be more attractive proposition than the

commitment intensive option of establishing a subsidiary. Thus, neither arms length transaction, nor a hierarchical solution seemed to be suitable, giving rise to the hybrid form of the ODC.

An ODC is able to incorporate both the characteristics of the markets and the hierarchies. The master service agreement serves as a foundation that determines the commercial agreement between the buyer and supplier. It incorporates confidentiality clauses that guard against possibilities of knowledge spillover. It also provides guarantee of business repeatability over extended time horizons, so that the supplier is in a better position to handle demand volatility. However, because the ODC is created in a context of an existing favaorable relationship between the client and the service provider, the master service agreement serves more to set the boundary conditions, than being the determinant of actual conduct between the two parties. Akin to a hierarchy, it accommodates process controls and behaviour control by the client over supplier operations and resources. It is quite common for the client manager to play an active role in the selection and development of ODC employees, even though such employees have no direct employment contract with the client. Coupled with the fact that such employees bear the identity of client organization and the ODC features in the organigram of the client organization, the ODC and its employees tend to identify a lot more with the client organization than with their actual employer – the service provider. Thus, the ODC becomes a subsidiary of the client organization that is located within the premises of the service provider – except the fact that the client does not need to make substantial investments and their commitment is time-bound by the length of the master service agreement. This unique governance mechanism has laid the foundation of some of the long-term relationship that is witnessed in this industry, such as those between clients like Nortel and GE with their various Indian software service providers, many of which have been continuing for more than a decade. The durability of such relationships is especially remarkable in an industry where historically alliances and joint ventures have been unstable.

Conclusion: Beyond Network Forms

The ODC is clearly a structural hybrid that accommodates important features of both the market and the hierarchy. It incorporates relational feature within a buyer supplier contract and sequential transactions takes place within the context of a general pattern of interaction. Its 'reliance over the long haul' (Powell, 1990) reciprocity and efficient exchange of information qualifies it to be another form of a network organization. However, it is essentially a dyadic relationship and even if viewed from the client end, interactions are one-to-many rather than many-to-many. Extant literature does not clearly specify whether a collection of dyadic relationships can qualify as a network form, even though each of the relationships exhibits features of a network.

The many-to-many relationships in the software service industry are however implicit, and a consequence of organizational action rather than organizational deign. The ODC enables a single service provider to maintain multiple dyadic relationships. Central to the service provider's value proposition is its ability to diversify risks, better than its clients would have succeeded individually. Akin to an investor in mutual funds, the service provider is able to mitigate the systemic risk associated with individual organizations and their industries by catering to multiple clients within the same industry, as well as to clients across multiple industries (Mukherji and Ramachandran, 2004). This brings about an implicit interdependence among the clients of the service provider, whose ability to extract value from the relationship becomes contingent upon the relationship that other organizations, some of who are its competitors, have with the service provider. One can view this as a many-to-many relationship, where a diverse range of organization become unconsciously dependent on one another, even though their explicit relationship is restricted to the service provider.

The above conceptualization closely matches the description of a natural ecosystem, where a diversity of species become interdependent on one another, even though they do not intend to collaborate consciously. Rather, population of species competes among themselves and with other populations, just as multiple clients of a service provider compete among themselves, within and across industries. Since extant literature on network forms identifies trust, collaboration and cooperation as distinguishing characteristics of networks (Powel, 1990; Podolny & Page, 1998), notion of competition is virtually absent from them.

Thus, we are witnessing a unique relationship in the Indian software industry, which is neither a market, nor hierarchy, nor is it a network form. Its resemblance with natural ecosystems tempts us to label it as a 'business ecosystem', though it differs from the existing conceptualization of business ecosystems in one significant way – the absence of a grand designer. In Iansiti & Levin's (2004) conceptualization, a lead firm or 'keystone' consciously assumes the role of a designer by creating a platform that can be leveraged by other members of an ecosystem. The Indian software service provider, in its present form, can hardly qualify as a platform creator. Thus, even though it acts as a hub for the evolution of the ecosystem around itself, it is not the designer of the same. In that sense, the ecosystem around the service provider is closer to the natural ecosystems, which have also evolved without any conscious design (Dawkins, 1991) and is a consequence of competition among a variety of species, rather than being a product of any explicit collaboration.

References:

Achrol R (1991), " Evolution of Marketing Organization: New Forms for Turbulent Environment", *Journal of Marketing*, October, pp 77-93

Arora A , V S Arunachalam, J Asundi & R Fernandes (2001) " The Indian Software Services Industry", *Research Policy* , 30: 1267-1287

Astley W G (1985), " The Two Ecologies: Population and Community Perspectives on Organizational Evolution", *Administrative Science Quarterly*, 30, pp 224-241

Baiman S (1982) " Agency Research in Managerial Accounting: A Survey". *Journal of Accounting Literature*, 1, pp: 124-153

Brooks F P (1987) "No Silver Bullet: Essence and Accidents of Software Engineering". *Computers* 20(4): 10-19

Bourgeois L J III & K M Eisenhardt (1988) "Strategic Decision Processes in High Velocity Environments: Four Cases in the Microcomputer Industry", *Management Science*, 34:816-835

Coase R (1937), "The Nature of Firm", *Economica*, 4, pp 386-405

Cohen W L & D A Levinthal (1990). "Absorptive Capacity: A New Perspective on Learning and Innovation", *Administrative Science Quarterly*, 35:128-152

Cravens D W , N F Piercy & S H Shipp (1996), "New Organizational Forms for Competing in Highly Dynamic Environments: the Network Paradigm", *British Journal of Management,* 7, pp 203-218

Cusumano M (1992) "Shifting Economies: From Craft Production to Flexible Systems and Software Factories". *Research Policy*. 21(5): 453-480

Dawkins R (1991); *The Blind Watchmaker* , Penguin

Govindarajan V & J Fisher (1990), "Strategy, Control Systems and Resource Sharing: Effects on Business Unit Performance", *Academy of Management Journal*, 33(2), pp 259-285

Iansiti M & R Levin (2004); "Strategy as Ecology", *Harvard Business Review*, March, pp: 68-78

Jarillo J C (1988), "On Strategic Networks", *Strategic Management Journal*, 9, Jan-Feb, pp 31-41

Moore J F (1993); "Predators and Prey: A New Ecology of Competition", *Harvard Business Review*, May-June, pp: 75-86

Mukherji S & J Ramachandran (2004) " Complementary and Continuous Innovations: Case of the Indian Software Industry", *Journal of Academy of Business And Economics*, 4 pp:211-215

Nooteboom, B (2002), *Trust: Forms, Foundations, Functions, Failures and Figures*. Edward Elgar

Podolny J M & K L Page (1998). " Network Forms of Organization", *Annual Review of Sociology*, 24, pp 57-76

Powell W W (1990); "Neither Markets Nor Hierarchy: Network Forms of Organizations", *Research in Organizational Behavior*, Vol 12 pp 295-336

Quinn J B (1992). *Intelligent Enterprise*, Free Press, New York

Ramachandran J & P Dikshit (2002) *Motorola India* . IIM Bangalore Case

Snow C C, R E Miles & H J Coleman (1992). "Managing 21st Century Network Organizations", *Organizational Dynamics*, 20(3) , pp 5 –20

Williamson O E (1975); *Markets and Hierarchies* New York:Free

Williamson O E (1985); *The Economic Institutions of Capitalism*. New York:Free