

MANAGEMENT & VIRTUAL DECENTRALISED NETWORKS: THE LINUX PROJECT

By George N. Dafermos

ABSTRACT

This paper examines the latest of paradigms – the ‘Virtual Network(ed) Organisation’ and whether geographically dispersed knowledge workers can virtually collaborate for a project under no central planning. Co-ordination, management and the role of knowledge arise as the central areas of focus. The Linux Project and its virtual decentralised development model are selected as an appropriate case of analysis and the critical success factors of this organisational design are identified.

The study proceeds to the formulation of a managerial framework that can be applied to all kinds of virtual decentralised work and concludes that value creation is maximized when there is intense interaction and uninhibited sharing of information between the organisation and the surrounding community. Therefore, the potential success or failure of this organisational paradigm depends on the degree of dedication and involvement by the surrounding community.

In addition, the paper discusses the strengths and implications of adopting the organisational model represented by the Linux Project in other industries.

“This paper was submitted as part requirement of the degree MA in Management of
Durham Business School, 2001”

Acknowledgements

I would like to thank my supervisor, Dr. Joanne Roberts for all the help I received over the research. Had not been for her guidance, this paper would not have materialised.

I also wish to thank all those who shared their experience and valuable insight with us by accepting to be interviewed. They are in alphabetical order (they are also mentioned in Appendeix IV: Interviewees):

Dan Barber, Chris Browne, Chris Dibona, Matt Haak, Philip Hands, Ikarios, Ko Kuwabara, Robert Laubacher, Michael McConnel, Glyn Moody, Ganesh Prasad, Richard Stallman

TABLE OF CONTENTS

Abstract
Acknowledgements

CHAPTER 1: INTRODUCTION

CHAPTER 2: FROM HIERARCHIES TO NETWORKS

PART 1: THE EVOLUTION OF THE ORGANISATION

- 2.1 Science seeks to solve problems
- 2.2 Enter the organization
- 2.3 Bureaucracy is the inevitable organisational design
- 2.4 The American Revolution
- 2.5 The Corporate Man
- 2.6 The beginning of the end
- 2.7 The fall of the old order
- 2.8 The Japanese threat
- 2.9 Quality is everything
- 2.10 Learning means evolving

PART 2: THE NETWORKED ORGANISATION

- 2.11 The Network structure reigns
- 2.12 Mergers, Acquisitions and Strategic Alliances
- 2.13 Economic Webs
- 2.14 Outsourcing & Software
- 2.15 Unbundling outsourcing
- 2.16 Virtualness & the Virtual Organisation
- 2.17 Project-centric perspective

CHAPTER 3: RESEARCHING AN EMERGING PARADIGM

- 3.1 Qualitative Research
- 3.2 Case Study Approach
- 3.3 Advantages of the method
- 3.4 Disadvantages
- 3.5 Primary Sources of Data
 - 3.5.a Observation
 - 3.5.b Interviews
- 3.6 Secondary Sources of Data
- 3.7 Framework of Analysis

CHAPTER 4: THE LINUX PROJECT

- 4.1 Open Source & Free Software
- 4.2 Innovation
- 4.3 Structure and Decentralised Development
- 4.4 Modularity
- 4.5 Motivation
- 4.6 Learning
- 4.7 Management of the Economic Web

CHAPTER 5: MICROSOFT VS. LINUX

- 5.1 Business Processes
- 5.2 Management, Structure and, Knowledge

CHAPTER 6: THE NEW PARADIGM

PART A: TRANSFORMATIONS OF MANAGEMENT

- 6.1 Management can be digital and networked
- 6.2 Management should ensure that the organizational and project design maximizes organizational learning and empowers big teams to collaborate digitally
- 6.3 Management Focus shifts from Organisational Dynamics to Economic Web Dynamics

PART B: IDENTIFYING THE NEW PARADIGM

- 6.4 The emergence of a new paradigm?
- 6.5 Motivation is the source of sustainability
- 6.6 The Virtual Roof
- 6.7 Knowledge is the competitive advantage
- 6.8 Rational Organisational Design

PART C: APPLICABILITY OF THE LINUX MODEL TO OTHER INDUSTRIES

- 6.9 Implications

CHAPTER 7: CONCLUSIONS

- 7.1 Epilogue
- 7.2 Commentary on the objectives of this research

APPENDIX

1. Microsoft – The Cathedral
2. Increasing Returns
3. Communication Networks
4. Interviewees
- Bibliography

LIST OF FIGURES

- 2.1 The Scalar Chain of Authority & Breakdown in Communication
- 2.2 New paradigm organization
- 2.3 Alliances in technologically, unstable, knowledge-intensive Markets
- 2.4 Alliances are driven by economic factors (environmental forces)
- 2.5 From the Value Chain to the Digital Value Network
- 4.1 Free Software
- 4.2 Open Source
- 4.3 Innovation skyrockets when users and producers overlap
- 4.4 Structure of Linux
- 4.5 The Linux development model maximizes learning
- 4.6 Positive Network Effects driving ongoing growth-adoption of the GNU/Linux operating system
- 6.1 Linux structure depicted as flows of information among value streams
- 6.2 Pareto's Law & The Linux Project
- 6.3 The Linux project & The Virtual Roof
- 6.4 Creation and Exploitation of Massive Knowledge
- I Synch-and-Stabilize Life cycle for Program Management, Development and Testing
- II Microsoft Scalar Chain of Control
- III Microsoft holds the most powerful position in a gigantic network of corporate partnerships
- IV Value increases with the number of users-members
- V The S - Curve

LIST OF TABLES

- 2.1 Mechanistic and Organismic Style of Management
- 2.2 Original Megatrends
- 2.3 Transition from Industrial to Information Age Organisations
- 2.4 From Closed Hierarchies to Open Networked Organisations
- 2.5 Modern and Virtual Organization compared on Weber's criteria
- 5.1 Microsoft Vs. The Linux Project
- I Overview of Synch-and-Stabilize Development Approach
- II Synch-and-stabilize Vs. Sequential Development

CHAPTER 1: INTRODUCTION

The last century has had a great impact on the organisational structure and management. During this period, organisations have gradually evolved from 'bureaucratic dinosaurs' to more flexible and entrepreneurial designs and consequently revised their management practices to cope with the constantly growing complexity of the business landscape and take advantage of a unique competitive advantage - knowledge.

In the same time, technological breakthroughs in connectivity have extended the reach of organisations and individuals alike to the extent that access to an unlimited wealth of resources without intervention of any central authority is feasible.

These technological achievements enabled organisations to become more centralised or decentralised according to their strategic orientation and further enhanced the efficiency of managing global business processes. However, centralisation is still the prevailing mode of managing despite the increased desirability of decentralised operations.

In the light of the volatility and competitiveness that the new world of business has brought with, new perceptions of the organisation and management have flourished. These perceptions are termed paradigms and this study examines the latest: the 'Virtual or Network(ed) Organisation'.

The Linux Project is an example of this emerging paradigm as it has defied the rules of geography and centralisation and has been growing organically under no central planning for the last ten years.

It is being co-developed by thousands of globally dispersed individuals that are empowered by electronic networks to jointly co-ordinate their efforts and, has recently gained the attention of the business world for its business model that represents a serious threat to leading software companies, especially Microsoft Corporation.

Rationale

To date, the existing organisational and management theory that examines the “virtual – network(ed) organisation” is not clear and does not provide more than a basic explanation about boosting technological developments related to emerging business opportunities to be seized by flexible organisations in a global, volatile marketplace.

Similarly, no in-depth analysis has been carried out regarding the management of “virtual organisations” and the key success factors that play a decisive role on the viability and potential success or failure of this fluid organisational structure.

Objectives

This primary research focuses on the management of decentralised network structures and whether virtual and decentralised collaboration is feasible, especially under no central planning.

It presents an attempt to analyse the Linux Project and identify the crucial success factors behind this novel organisational model with emphasis on its management and investigate whether the adoption of this model in other industries is likely to be successful.

Also, it seeks to provide a managerial framework that can be theoretically applied to industries other than the software industry. The prospective opportunities and limitations of the framework’s adoption are analysed.

- ❑ Chapter 2 documents the evolution of the organisation with emphasis on the role of knowledge and discusses the emerging paradigm - the 'virtual network(ed) organisation'.
- ❑ Chapter 3 shows how this research was carried out and explains the choice of the methods used. The strengths and limitations of the chosen approach are also analysed.
- ❑ Chapter 4 provides some background information on the software industry and analyses the Linux Project.
- ❑ Chapter 5 compares Microsoft Corporation with the Linux Project and highlights areas of significant difference.
- ❑ Chapter 6 provides a managerial framework that may be suited to all types of virtual decentralised work, analyses the new organisational – management paradigm as proposed by the Linux Project and, discusses the applicability of its model to other industries.
- ❑ Chapter 7 comments on the completion of this research and whether this study's objectives have been met.

CHAPTER 2: FROM HIERARCHIES TO NETWORKS

PART A

THE EVOLUTION OF THE ORGANISATION

2.1 Science seeks to solve problems

The concept of hierarchy is built on three assumptions: the environment is stable, the processes are predictable and the output is given (Hedlund 1993). Obviously, these assumptions no longer apply to today's business landscape.

Hierarchies were first developed to run military and religious organisations. However, hierarchies with many layers started to appear in the 20th century, in organisations as the sensible organisational design. In 1911, the book *The Principles of Scientific Management* was published. F.W Taylor proved that efficiency and productivity are maximized by applying scientific methods to work. When he started working, he realised that the most crucial asset of doing business - knowledge and particularly technical know-how about production - was well guarded in the heads of workers of the time. He was the first who developed a methodology to convert tacit knowledge into explicit knowledge. He intended to empower managers to understand the production process. Armed with a watch, he embarked on his 'time-and-motion' studies where by observing skilled-workers, he showed that every task when broken down to many steps would be easily disseminated as knowledge throughout the organisation. As learning did not require months of apprenticeship, power - knowledge about production - passed from workers to managers. Ironically, the man who grasped the significance of communicating knowledge throughout the organisation, had formulated a framework that regarded the organisation as a machine and the workers as cogs.

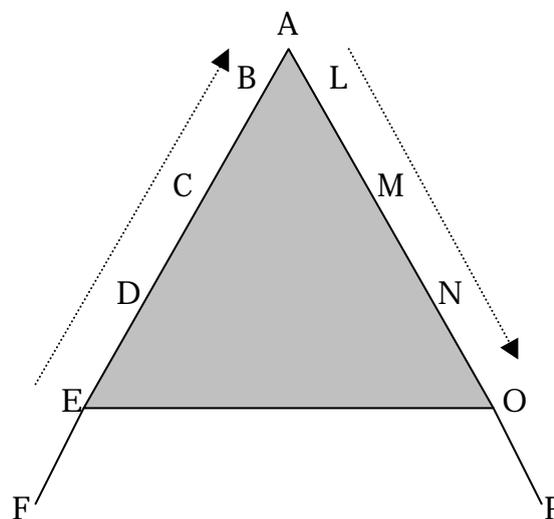
2.2 Enter the organisation

Shortly after Taylor, H. Fayol (1949) elaborated a managerial framework. He focused on the efficiency of the production process and reinforced Taylor's view that specialisation is essential along with constant supervision, and that no organisation can prosper without

a set of rules that 'control and command'. That was the part of his 'story' that got well accepted at the time.

The other side was anarchical for then, but utterly prophetic. He rejected the abuse of managerial power since authority is not to be conceived of apart from responsibility. Moreover, he was the first to identify the main weakness of hierarchy: breakdown in communication (Fig. 2.1) and pointed out that employees should not be seen as cogs in a machine. Despite his insight that hierarchy does not (always) work, he concluded that a "scalar chain" (hierarchical chain) of authority and command is inevitable as long as mass production and stability is the objective.

Figure 2.1 The Scalar Chain of Authority & Breakdown in communication



Section E needs to contact section O in a business whose scalar chain is the double ladder F-A-P. By following the line of authority, the ladder must be climbed from E to A and then descend from A to P, stopping at each rung, and then from O to A and from A to E. Evidently it is much easier and faster to go directly from E to O but bureaucracy does not allow that to happen very often.

Source: Adapted from H. Fayol, General and industrial Management, Ch.4, 1949

2.3 Bureaucracy is the inevitable organisational design

Taylor showed the way, Fayol provided a set of rules and Weber evangelised the adoption of bureaucracy as the rational organisational design. His writings were so influential that modern management theory is founded on Weber's account of bureaucracy. Firstly, he made the distinction between 'power-force' and authority: the

'power-force' implies that the management forces employees to act whereas 'authority' implies that managers give directions on reasonable grounds and based on well-known legitimate rules. Weber was convinced of the superiority of a bureaucratic authority (legal authority with a bureaucratic administrative staff as he termed it). He based his analysis on the evidence that long-living, successfully stable organisations like the army are being brought together by clear rules delivered by 'officers'.

To deal with the complexity of increasingly larger organisations, an 'administration system' should be enforced to control the flow of knowledge and the employees, and in this way, trigger unprecedented efficiency in (mass) production. In his words: [bureaucracy] is capable of the highest degree of efficiency... as inevitable as the precision machinery in the mass production of goods...it makes possible the calculability of results for the heads of the organisation... is the most important mechanism for the administration of everyday profane affairs.

Ironically, he replied to the critics of bureaucracy by arguing that any other structure is an illusion and only by reversion in every field – political, religious, economic – to more flexible structures would it be possible to escape the influence of 'bureaucracy'. History has now proven that his revelations are correct. Now, it was the time of 'industry' men to shape the organisation and managerial minds along Weber-Taylorist lines.

2.4 The American Revolution

The 1840s and the US railroads marked the beginning of a great wave of organisational change that has evolved into the modern corporation (Chandler, 1977). A. Chandler is the first "business historian". His study *Strategy and Structure* (1966) shed light on the American corporation, focusing on General Motors (run by A. Sloan in the 1930s) and du Pont. Chandler analysed the defects of the centralised, functionally departmentalised structure and argued that the bigger a company grows, the more inefficient a hierarchy gets because the management can no longer deal with the increasing complexity of co-ordinating people (Chandler, 1966, p382-383). He concluded that decentralisation will flourish, as it allows large companies to establish an organisational platform for better communication and co-ordination.

A few years later, Chandler laid emphasis on control and transaction cost economising and explained how decentralisation and hierarchy could fit together. He characterised this as the 'decentralised line-and-staff concept of the organisation' where the managers

were responsible for ordering men involved with the basic function of the enterprise, and functional managers (the staff executives) were responsible for setting standards (Chandler, 1977, p.106). Express delegation of authority was of paramount importance (Chandler, 1977 p102).

2.5 The Corporate Man

Henry Ford adored the idea that organisations were modelled on machines and workers were regarded as 'cogs' and invented the 'assembly line': a system of assembly-line production (known as Fordism), based on divided labour linked together mechanistically. These 'cogs' should be steered in a systematic way to boost efficiency and had no ability to innovate, think or improvise. If every cog had been assigned a specific, repetitive task, everything was supposed to go well. Everything was organised through a pyramid of control designed in a purely bureaucratic fashion. The key word was mass: mass production, mass markets.

2.6 The Beginning of the End

By the 1970s this model had begun to falter. A slowdown in productivity, international competition and upward pressures on wages squeezed profits (Clarke & Clegg 1998). These bureaucracies decided to expand, to embark on a process of 'internationalisation' by following Chandler's guidelines on decentralisation. The aim was reduction of costs. Economising on costs by using cheaper labour was not the solution to satisfy consumers and respond to overseas competition. In a saturated market of no expanding consumer demand, the corporate mantra "any colour they want as long as it is black" no longer worked. Consumers started complaining about low quality, workers were crying out for more rights and even sabotaged the production process, and competitors from abroad started invading the US and European market, particularly Japanese carmakers and consumer electronics. Management writers were claiming that the Japanese threat's success was attributed to a different management paradigm.

Different analysts argued that managers had paid no attention to organisational thinkers since the revolution that the assembly line brought with. In 1963, T. Burns recognised that the hierarchy of command is maintained by the assumption that the only man who

knows – or should know – all about the company is the man at the top, and this assumption is entirely mistaken (Burns, 1963 p18). He categorised the organisation according to two opposite management systems (table 2.1): the mechanistic and the organismic.

Table 2.1 Mechanistic and Organismic style of management

A mechanistic management system is appropriate to stable conditions. It is characterised by:

- Hierarchic structure of control, authority and communication
- A reinforcement of the hierarchic structure by the location of knowledge of actualities exclusively at the top of the hierarchy
- Vertical interaction between the members of the concern, ie. between superior and subordinate

The Organismic form is appropriate to changing conditions. It is characterised by:

- Network structure of control
- Omniscience no longer imputed to the head of the concern; knowledge may be located anywhere in the network; the location becoming the centre of authority
- Lateral rather than vertical direction of communication through the organisation
- A content of communication which consists of information and advice rather than instructions and decisions

Source: T. Burns, *Industry in a New Age*, *New society*, 31 Jan 1963, pp.18

The management of a successful electronics company struck him as ‘dangerous thinking’ because written communication was discouraged and any individual’s job should be as little as possible, so that it would ‘shape itself’ to his special abilities and initiative (Burns & Stalker, 1961). This management style is obviously the organismic.

What he realised is that turbulent times ask for different structures (Burns 1963 p18).

He concluded that a mechanistic system is appropriate to stable conditions whereas the organismic form is for changing conditions, which give rise to fresh problems and unforeseen requirements for action which cannot be broken down or distributed automatically arising from the functional roles defined within a hierarchic structure.

Similarly, Lawrence and Lorsch suggested that managers can no longer be concerned about the one best way to organise (Lawrence & Lorsch, 1967). All they suggested was that the more complex the environment becomes, the more flexible the structure should be to allow for rapid responses.

2.7 The Fall of the Old Order

These organisations made standardised products, for relatively stable rational markets. Their aim was consistency and control; creativity and initiative were frowned upon. But with the liberalisation of world trade, competition became fierce and consumers started demanding products tailored to their needs (Leadbeater 2000).

This signalled the fall of the command-and-control hierarchy. There were just two problems. First, mass-production oriented processes had been 'stove-piped' into non communicating business functions. Second, "workers told to 'check your brain at the door', were ill-equipped for the dynamic changes about to wreak havoc on the corporation" (Locke, 2001, p25). The bureaucratic firm was not qualified to generate knowledge and continuous learning to adapt to these turbulent times.

2.8 The Japanese Threat

Sakichi Toyota visited the Ford plant in the 1950s. He realised that much would have to be done differently in Japan (Cusumano 1985) because the Japanese market demanded many types of cars, thus a more flexible manufacturing system was needed.

Toyota understood that such flexibility and speed could only be delivered by establishing close relationships with suppliers on the basis of mutual benefit. Suppliers became involved in critical decisions and Toyota instead of vertically integrating with them, preferred to use 'just-in-time- (JIT) systems. JIT systems establish complex relations with component subcontractors so that supplies arrive when needed. The benefit is minimisation of inventory costs and acceleration of innovation. This is achieved because personnel and ideas are freely exchanged between the partners that make up the subcontracting network (Clarke & Clegg 1998). Toyota came up with a method of creating, sharing and disseminating knowledge.

Toyota gave rise to significant innovations in production (ie. Jidoka). However, the most distinct innovation was the management. Toyota was built on Taylor's principles and had a hierarchy with many layers. What they did so differently was that Toyota empowered its employees. They introduced radical methods like job-rotation and project-form of organising. New skills are built into the employees as the latter become more flexible and mobile. Workers were encouraged to develop more skills and work content was not

inexorably simplified as in the typical organisation under Fordism. They used self-managing teams where 'team members' allocated tasks internally without any intervention from the higher management and when the team had reached the improvement limits, the team members would move to other areas to pick up new skills. It was the first time that the workers could stop the machines and make crucial decisions. Of course, a sense of trust was developed among the network of partners (Ibid). After all, workers, management and suppliers were a 'family'.

The Japanese had had the same objectives their competitors had: continuous improvement in production. But they realised the strategic importance of the human element and they encouraged their employees to become kaizen (continuous improvement) conscious by developing as many skills as possible. Also, they seized the opportunities provided by networking between suppliers and the firm to become faster, more flexible and reduce costs. In addition, this model does not take for granted that customers will buy whatever they are offered. The whole production system seeks to ensure quality (Ibid). This management style is called "lean production".

2.9 Quality is everything

In the 1980s, quality was the hype. Managers thought that if the lean production model worked for the Japanese, then it would work everywhere. This signalled the era of the Quality movement. At the centre is a managerial philosophy that seeks to increase organisational flexibility enabling companies to adapt to changes in the marketplace and swiftly adjust business processes (Dawson & G. Palmer, 1995, p3-4). Quality became synonymous with change, employee empowerment and customer focus.

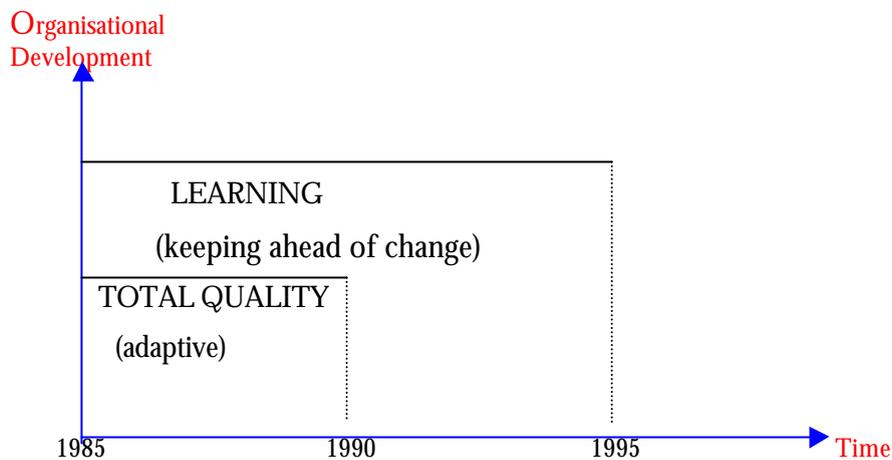
The new philosophy was termed TQM (total quality management). Dawson and Palmer identified the TQM as "a management philosophy of change which is based on the view that change is necessary to keep pace with dynamic external environments and continually improve existing operating systems. Those organisations embracing this new philosophy support an ideology of participation and collaboration through involving employees in decision-making" (Ibid pp29-30). From late 1970s onwards, all (Western) corporations jumped on the TQM bandwagon. They were evangelising change and; the only sad thing was that the emphasis was on incremental innovation, instead of radical

(Clarke & Clegg 1998). Depending on the organisation and how the TQM approach was implemented, it worked reasonably well until 1990.

2.10 Learning means Evolving

In 1992, P. Senge put into a context what was already known. The role of knowledge is so crucial that no organisation can afford not to extend its existing knowledge and create new. He dwelled upon social systems theory' concepts and turned them upstream. The organisation as a social system is an information model whose viability depends upon its capacity for self-design (Gherardi, 1997, p542). The difference between 'learning' and TQM was that the emphasis was now on organisational learning rather than individual learning. The point was that organisations should learn to do different things in different ways (Hayes, Wheelwright & Clark, 1988, p252). Characteristically, Hodgetts, Luthans and Lee conceptualise it as the transition from an adaptive organisation to one that keeps ahead of change (fig 2.2).

Figure 2.2 New Paradigm Organisation



Source: Hodgetts, Luthans and Lee, New Paradigm Organizations, (1994)

The 'learning paradigm' suggested that 'power' tends to shift towards smaller firms as they learn relatively faster (knowledge flows more freely within small than large firms due to absence of bureaucratic impediments) (Rothwell, 1992).

Furthermore, G. Morgan supported that learning is maximized in flexible, decentralised modes of operation. He insisted that a decentralised networked organisation is the "best

design” as long as the network is fostered and not managed and he resembled this design as a spider web. He stressed it is pure risk, but modern times demand risk-embracing to bring innovation (G. Morgan, 1994).

PART B

THE NETWORKED ORGANISATION

2.11 The network structure reigns

Organisations have been forming stable or elastic networks for a long time. Reasons for doing so vary greatly. Some attribute it to harsh competition because of deregulated global consumer markets and others claim that it is the only way to gain access to new markets, new technologies and know-how. Due to environmental forces and emerging opportunities that firms could exploit, being close to your environment (suppliers, customers and competitors) started being recognised as a unique competitive advantage. Management gurus have long ago evangelised the advantages related to networking and ‘condemned’ the disadvantages related to bureaucracy and command-and-control hierarchies.

J. Naisbitt prophesied the shift to a decentralised, networked, global organisation ([Table 2.2](#)).

Table 2.2 Original Megatrends	
From	To
Industrial society	Information society
National economy	World economy
Centralisation	Decentralisation
Hierarchies	Networks

Source: J. Naisbitt, Megatrends, 1982

Organisations controlled by hierarchies where the functional departments are separated will be replaced by organisations based on team-work with cross-teams that treat people

as assets (Hall 1993, p.281). Hames emphasised that this paradigm relies on open and adaptive systems that promote learning, co-operation and flexibility and takes the form of networks of individuals instead of individuals or structures alone (Table 2.3) (1994).

Table 2.3 Transition from Industrial to Information Age Organisations

Industrial Age		Information Age
Focus on measurable outcomes	→	Focus on strategic issues using participation and empowerment
Individual accountability	→	Team accountability
Clearly differentiated-segmented organisational roles, positions and responsibilities	→	Matrix arrangement – flexible positions and responsibilities
Hierarchical, linear information flows	→	Multiple interface, 'boundaryless' information networking
Initiatives for improvement emanate from a management elite	→	Initiatives for improvement emanate from all directions

Source: Hames, The Management Myth, 1994

Tapscott and Caston argued that control by hierarchies recedes in efficiency due to boosting technological advancements that favour open networked organisations (Table 2.4)

Table 2.4 From Closed hierarchies to Open Networked Organisations

	Closed hierarchy		open networked organisation
Structure	hierarchical		networked
Scope	internal/closed		external/open
Resource focus	capital		human, information
State	stable		dynamic, changing
Direction	management commands		self – management
Basis of action	control		empowerment to act
Basis for compensation	position in hierarchy		competency level

Source: Tapscott & Caston, Paradigm Shift, p.22 (1993)

The 'boundaryless' networked organisation envisages the ideal flexible production system that serves niche-based markets as the response to a society characterised by a decline in the ideas of mass society, mass market and mass production as people no longer want to be identified as part of the mass (Limerick & Cunnington 1993).

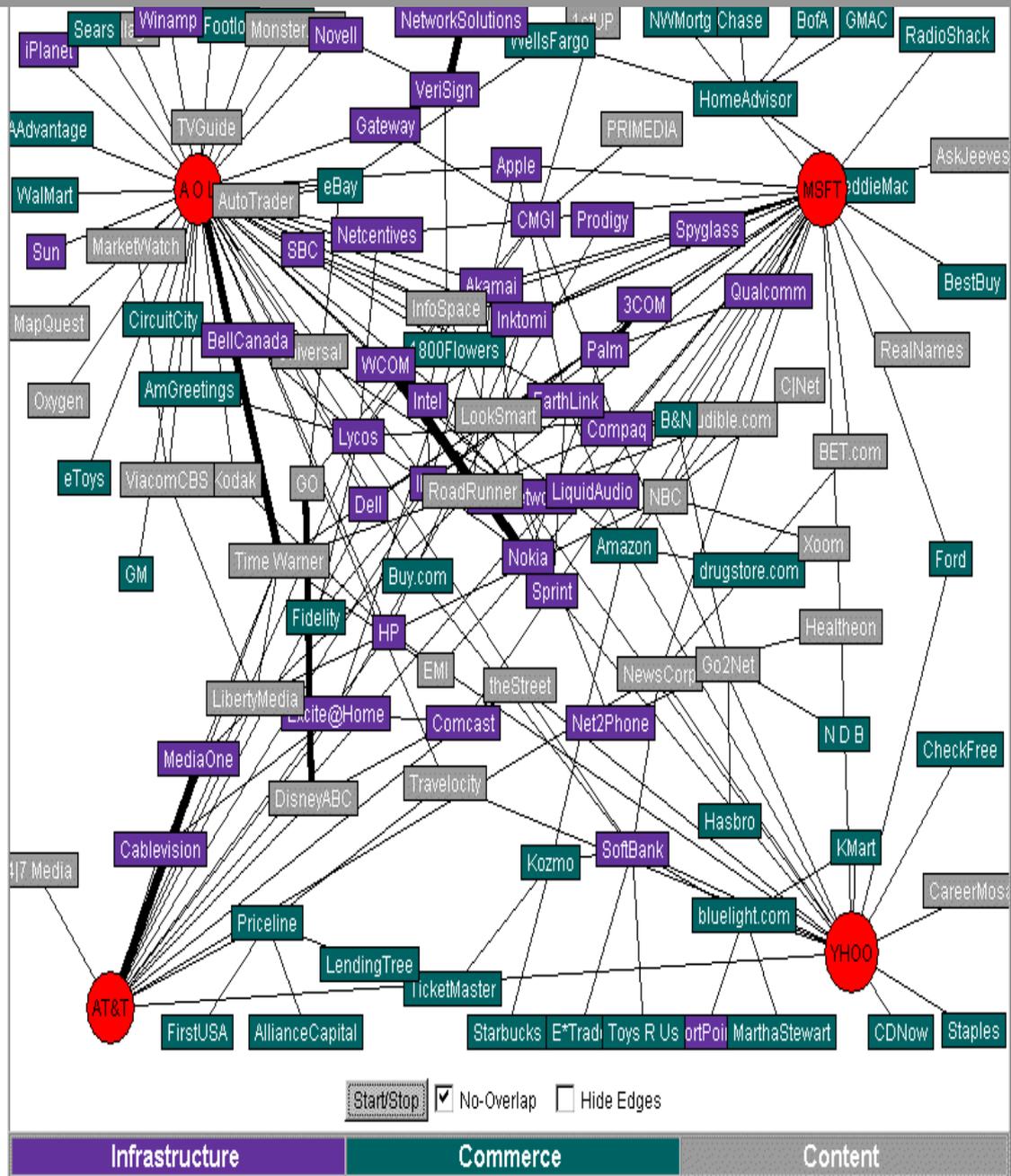
By the late 1990s, management thinkers had embraced the 'networked organisation paradigm' and had similarly dismissed the authoritative model. Typical is the comeback of Hames (1997) who again proclaimed "what hierarchy was to the 20th century, the distributed network will be to the 21st...the network is the only organisational type capable of unguided, unprejudiced growth...the network is the least structured organisation that can be said to have any structure at all' (p141).

2.12 Mergers, Acquisitions & Strategic Alliances

Ansoff was the first to propose the notion of 'synergy' and that $2+2=5$, implying that companies could attain a competitive advantage by joining forces (1965). Mergers, acquisitions and strategic alliances were the first wave of networking disguised under 'internationalisation' and 'expansion'. However, most did not deliver merely because: a) there was 'no strategic fit' (they did not operate in the same or complementary market and thus they could not add any value) between the new partners, b) they were built on bureaucratic structures that impeded the information flow or were shattered by corporate politics, c) appealed to managers only because it was more "fun and glamorous" to run bigger firms (Ridderstrale & Nordstrom, 2000) and d) it was the expensive way to get networked (Hacki & Lighton, 2001). Whatsoever, the drivers behind strategic alliances witness that in such a competitive marketplace, the only way to compete is through a network (Fig. 2.4).

Nowadays, it has become increasingly common to pursue organisational sustainability and economic self-interest by establishing some kind of alliance especially in information – intensive industries that are in the forefront of upheaval and therefore 'galvanised' by technological uncertainty (Fig. 2.3).

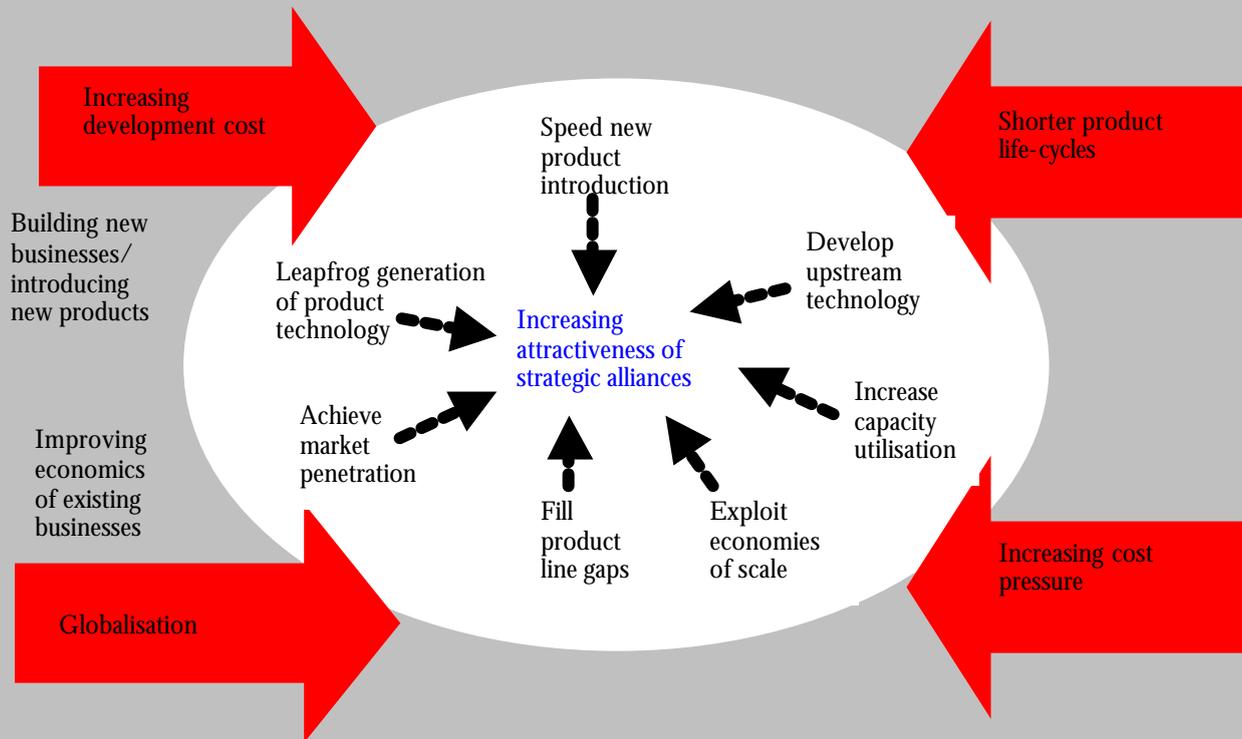
Figure 2.3 Alliances in technologically unstable, knowledge – intensive Markets



Source: www.orgnet.com

The figure is originally titled 'Internet Industry: Strategic Alliances, Joint Ventures and other Partnerships'

Figure 2.4 Alliances are driven by Economic Factors (Environmental forces)



Source: Krubasilk & Lautenschlager, Forming successful Strategic alliances, (1993), p56

2.13 Economic Webs¹

An economic web is a dynamic network of companies whose businesses (are built around a single common platform to) deliver independent elements of an overall value proposition that strengthens as more companies join (Hagel III). Webs are not alliances. There is no formal relationship among the web's participants as the latter are independent to act in any way they choose to maximize their profits. And this is what drives them into weblike behaviour. Typical example is the Microsoft-Intel ("Wintel") web, composed of companies that produce Windows – Intel –based software applications and related services for PC users. Unlike alliance networks, in which companies are invited to join by the dominant company, economic webs are open to all

¹ Even though the use of the term 'Economic Web' is not established and commonly used, Nalebuff & Brandenburger (1997) and Shy (2001) have reached similar conclusions regarding the importance of the 'economic web' in strategy formulation. Their 'conception' is based on Game Theory applications

and numbers equal power. The purpose of a network platform is to draw together participating companies by facilitating the exchange of knowledge among them.

The platform (a technical standard) of an economic web does not affect participating companies' relationship with the shaper (the company that owns the standard) and enables them to provide complementary products and services (Hacki & Lighton, 2001 p33). Two conditions must be present for a web to form: a technological platform and increasing returns²

The technological standard reduces risk as companies need to make heavy investments in R&D in the face of technological turbulence, while the increasing returns create a dependency among participants by attracting in more producers and customers (Hagel III, 1996).

2.14 Outsourcing & Software

J. B. Quinn explained why software increases in strategic importance and enables networked structures, and introduced the concept of strategic outsourcing: 'Concentrate the firm's resources on its core competences where it can achieve pre-eminence and provide unique value for customers' (Quinn & Hilmer 1994). The advantages are: a) By outsourcing, you concentrate your resources on what the firm best does, b) well formed core competences present perfect barriers to entry against potential competitors, c) a company can mobilize the ideas, the innovations and the specialist skills of the suppliers, which it would never be able to replicate itself, d) in rapidly changing markets, with shifting technologies, this collaborative strategy reduces risk, shares know-how, speeds learning and shortens development cycles.

In 1990, he visualised the firm as a package of service activities; and that services, not manufacturing activities provide the major source of value to customers. According to this view, bureaucracy has to be destructed as it was developed for the era that manufacturing was the primary platform of delivering value added. He suggested that "...there is no reason why organisations cannot be made 'infinitely flat' guided by a computer system" (Quinn & Paquette 1990a, pp67-78 and 1990b, pp.79-87). He investigated the transition to a "spider web" organisation which is a non-hierarchical

² see APPENDIX II – Increasing Returns

network and stressed that innovative organisational forms depend on software (Quinn, Anderson & Finkelstein, 1996, pp71-80).

The apotheosis came in 1996 when he proved that software is so pervasive that is the primary element in all aspects of innovation from basic research to product introduction and, software is the facilitator of organisational learning that innovation requires as well as an excellent platform of collaboration (Quinn, Baruch & Zien, 1996).

2.15 Unbundling outsourcing

Outsourcing focuses the key resources of an organisation on its core value-adding processes. “This is not vertical integration within an enterprise but vertical and horizontal integration across organisations, including alliance partners, sales and distribution agencies, key suppliers, support organisations, and other divisions within their own company” (Tapscott & Caston, 1993, p.9).

“Focus on what you excel at and outsource the rest”. Nike has struck gold by not applying its slogan (Davies, & Meyer 1998). Timberland no longer makes shoes and in the case of Dell we are talking about a factoryless company. M. Dell realised that “IBM took \$700-worth of parts, sold them to a dealer for \$2000 who sold them for \$3000. It was still \$700-worth of parts” (Scanorama 1995). But what he actually realised is that information can replace inventory. As D. Hunter, Chief of Dell’s Supply Chain Management says: “Inventory is a substitute for information: you buy them because you are not sure of the reliability of your supplier or the demand from your customer” (The Economist, 2000a: p36).

2.16 Virtualness & the Virtual Organisation

The term virtual means “not physically existing as such but made by software to do so” (*concise Oxford dictionary, s.v. Virtual*) and management literature identifies the virtual organisation as the extreme form of outsourcing or the exact opposite of Weber’s bureaucracy ([table 2.5](#))

Table 2.5 Modern and Virtual organisation compared on Weber’s Criteria

Modern Organisation	Virtual Organisation
Functionality in design structure	Defunctionalised project-based design held together by network capabilities
Hierarchy governing formal communication flows and managerial imperative the major form and basis of formal communication	Instantaneous remote computer communication for primary interaction; increase in face-to-face informal interaction; decrease in imperative actions and increased governance through accountability in terms of parameters rather than instructions or rules
The files	Flexible electronic immediacy through IT
Impersonal roles	Networking of people from different organisations such that their sense of formal organisation roles blurs
Specialised technical training for specific careers	Global, cross-organisational computer-mediated projects

Source: Nohria & Berkely, the Virtual Organisation (1994)

But the literature is not clear. Davidow’s and Malone’s classic – the Virtual Corporation – identifies the virtual organisation to be an organisational design enabled by technology to supersede the information controls inscribed in bureaucracy and allow the employees to communicate electronically from any location.

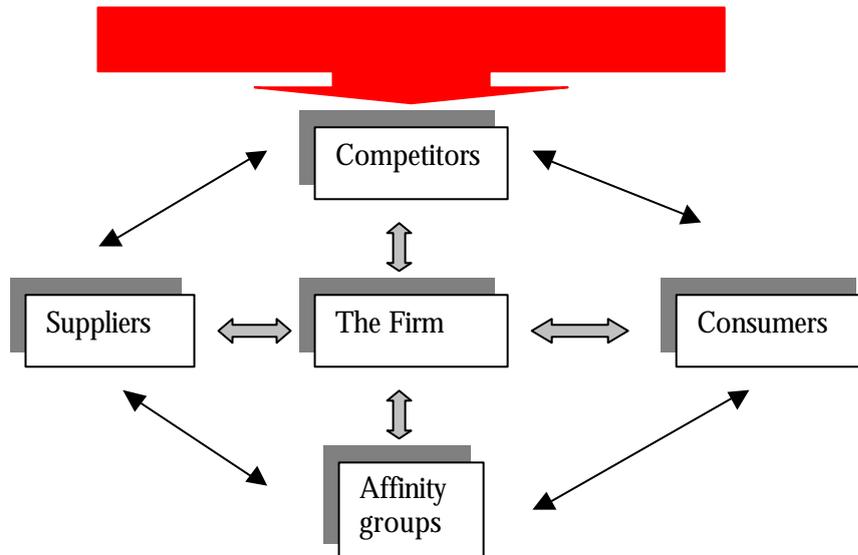
Charles Handy comments on the “virtual organisation” by citing a man using a laptop, a fax and a phone to transform his car to a mobile office and argues that the organisation of the future may outsource all processes and its employees will be communicating like that man (Handy 1989). Elsewhere, he cites Open University as an example due to the non-existent physical assets of the University (Handy 1995, p40-50).

Whereas, Tapscott points out that the value chain becomes a value network, as new (digital) relationships become possible ([figure 2.5](#)).

Figure 2.5: From the Value Chain to the Digital Value Network



The Value Chain (information flow physical: value added)



The digital value network (information flow digital: value generated)

Source: Adapted from Tapscott, *The Digital Economy*, p86-87, (1996)

Not surprisingly, what you mean by referring to a “virtual organisation” is subjective. Some imply an organisation that is electronically linked to its environment; others refer to a firm without offices that exists only in the dimension of computer networks.

2.17 Project – centric perspective

Moreover, few explored the ‘networking paradigm’ from a project-centric perspective. The Hollywood is one of the most famous ones where a ‘crew’ or network of people is formed to make a movie (a specific project) and as soon as the film is completed, the ‘temporary company’ is terminated (Malone & Laubacher, 1998).

Another example is the Silicon Valley. Average job tenure there is two years. Individual firms come and go – temporary alliances of people pursuing specific, narrowly defined projects. In some ways, Silicon Valley performs as a large, decentralised organisation. The Valley, not its constituent firms, owns the labour pool. The Valley, through its venture capital community, starts projects, terminates them, and allocates capital among them (Evans & Wurster, 1999, p211).

It is apparent that an organisation or a network can be formed on a temporary basis in order to undertake a project and once this is done, the whole organisation or network will disband. Eventually, all supply chains (or organisations) might become ad hoc structures, assembled to fit the needs of a particular project and disassembled when the project ends (Malone & Laubacher, 1998).

For the purpose of this paper, the “virtual organisation” we refer to, ***is a collection of people that interact only via electronic networks during the knowledge-intensive phases of the product produced (ie. design).***

CHAPTER 3: RESEARCHING AN EMERGING PARADIGM

3.1 Qualitative Research

It is traditional in the social and organisational research field to apply qualitative research techniques to the study of large social units (ie. organisations) and quantitative techniques to the study of individuals (Etzioni 1969). Researchers have tried to bridge this gap by applying quantitative techniques to the study of organisations (Lazarsfeld & Menzel 1969, Coleman 1969, Zelditch 1969) but even today most researchers that focus on organisational trends/changing paradigms choose to undertake a qualitative approach, particularly the 'case study' analysed below.

In retrospect, qualitative research facilitates quantitative research by means of acting as a precursor to the formulation of problems and the development of instruments for quantitative research: qualitative research may act as a source of hypotheses to be tested by quantitative research (Sieber 1973) and the author of this paper is optimistic that this research will trigger further investigations to test the concluding hypotheses.

3.2 Case Study Approach

Researchers still rely on this basic approach (indepth case study) when they encounter a 'non thoroughly examined group phenomenon' and it can be deployed to examine a wide variety of groups (or organisations) (Festinger, Riecken & Schachter 1956, Radloff & Helmreich 1968, Hare & Naveh 1986, Stones 1982, White 1977, Bennett 1980).

One of the reasons behind the lack of extensive research and literature on 'virtual organisations' is merely because the latter represent an emerging phenomenon – organisational structure.

Therefore, the 'case study' is the only approach that can shed light and provide a profound view at the absence of a spectrum of proper examples to draw upon as it allows for a significant amount of data to be gathered and thorough examination.

The selection of a suitable 'case study' was obvious - the last ten years have provided us with a perfect example of virtual collaboration: the constantly evolving Linux (or

GNU/Linux) operating system project that is being co-developed and maintained by thousands (the exact number is unknown) of geographically dispersed people.

It certainly fulfils all the prerequisites for our research since all the phases of the development process take place only in the Internet without any physical interaction among the developers.

To enhance the validity of the case study approach, we decided to compare the chosen 'virtual organisation' with an organisation that is a) operating in the same industry, b) highly competent and competitive, c) has access and deploys a large amount of resources and d) relies on centralised decision-making, management and development.

Again, the selection was apparent – the Microsoft Corporation. There could not have been a more appropriate organisation since:

- a) Microsoft (MS) dominates the software industry for the last twenty years and thus:
 - Has access to/deploys a massive amount of resources (expert skills and talent, financial capital and market power-share)
 - Is known for being aggressively competitive
- b) both MS and Linux compete directly for the operating system market.

MS is the archetypical centralised model in the industry (it is further justified by this paper – see analysis of MS in the **Appendix III: Microsoft – The Cathedral**).

However, due to lack of space, the comparison between Linux and Microsoft that appears in **Chapter 5** focuses solely on the critical differences between the two models and does not offer a complete overview of Microsoft.

For this purpose, we have included Appendix I which aims at facilitating the reader's inquiry by providing a complete analysis of Microsoft. Appendix I also serves to support the assumptions that appear in the comparison in Chapter 5.

3.3 Advantages of the method

Case studies allow in-depth understanding of the group or groups under study, and they yield descriptions of group events – processes often unsurpassed by any other research procedure. Also, and at a more pragmatic level, case studies can be relatively easy to carry out and they make for fascinating reading.

But the real forte of the case study approach is its power to provide grist for the theoretician's mill, enabling the investigator to formulate hypotheses that set the stage for other research methods (Forsyth 1990).

3.4 Disadvantages

□ generalisation

Case studies, however, yield only limited information about groups in general (no generalisation can be made). Researchers who use this method must constantly remind themselves that the group studied may be unique and therefore non-representative of other groups.

□ bias

Also, because researchers can't always use objective measures of group processes when conducting case studies, their interpretations can be influenced by their own assumptions and biases.

In all, case studies limit the researcher's ability to draw conclusions, to quantify results, and to make objective interpretations. However, some topics such as groupthink, group decision-making and group work are almost impossible to study by any other method (Janis 1963, 1972, 1983, 1985, 1989, Janis & Mann 1977).

3.5 PRIMARY SOURCES OF DATA

3.5.a Observation

Because the group has not yet disbanded, we decided to observe it as it carries out its functions (through the various Linux mailing lists, web sites, bulletin boards).

The main strength of observation is its capacity to reveal covert and hidden activities. Studies of groups in large organisations (Dalton 1959), output regulation in industrial work groups (Roy 1960) and pilferage at work (Ditton 1977) all demonstrate the capacity to look behind the scenes and bring to the centre of the stage aspects of these milieux which would otherwise be inaccessible or possibly not even uncovered in the first place (Bryman 1988).

This study though is peculiar in terms of the observation technique used. It does not fall into any of the categories of observation (see Scott 1969 and Adler & Adler 1998 for the observation categories).

The key difference is that I observed the Linux community not physically, but virtually*. It is not participant observation or action research since I did not get involved in the actual development process and I did not engage in any conversation that took place in the various mailing lists and (virtual) discussion forums.

Furthermore, major issues such authorisation to enter and explore the particular organisation and whether I should reveal my 'research identity' (overt observation/open researcher) or keep it a secret (covert observation) (Schwartz & Schwartz 1955, Schwartz & Jacobs 1979, Whyte 1943, Landsberger 1958, Mayo 1945, Roethlisberger & Dickson 1939, Bramel & Friend 1981, Franke 1979, Franke & Kaul 1978, Barnard 1938, McGregor 1960) or how to present myself (Becker 1956, Spradley 1979, Fontana 1977, Thompson 1985, Malinowski 1922, Wax 1960, Johnson 1976) and gain trust (Frey 1993, Cicourel 1974, Rasmussen 1989) are irrelevant in this study because access to the

* For a long time, social researchers have been concerned with the impact of Computer Mediated Communication Technologies (ie. newsgroups, email, IRC, etc) on the conduct of primary social research (ie. observation). For a more elaborate discussion, see Belson (1994); Reid (1991), Rheingold (1993) and, Hiltz & Turoff (1978 & 1991)

organisation (through the mailing lists, etc.) is open to the public without asking for permission. There is no question then of invasion of privacy or of using a deceptive observation method (Cook 1981, Douglas 1976, Reynolds 1979).

This unique attribute adds to the overall objectivity and validity of our findings as the individual or group behaviour could not have changed due to our 'virtual presence'. Hence, the possibility for bias is eliminated.

3.5.b Interviews

Personal indepth semi-structured interviews have been used, by phone and e-mail (the decision was up to the interviewees).

Semi-structured interviews can produce both qualitative and quantitative data (Crozier 1964) but the reason we chose this form of interviewing was in order to make the interview pleasing to the persons interviewed (Webb & Webb 1932, p139), and because semi-structured interviews provide rich detailed data of greater value than straight question and answer sessions especially when the research aim is to explore a phenomenon (Zweig 1948). Moreover, they do not put the interviewer in an unnatural relationship with those who are researched (Oakley 1981, Wakeford 1981).

In overall, field researchers use a semi-structured (or unstructured) approach that is based on developing conversations with informants (Burgess p121) and this strategy follows a long tradition in social research where interviews have been perceived as 'conversations with a purpose' (Ibid, p102). They are usually used to complement observation and other techniques and they achieve the flexibility needed when dealing with a complex phenomenon (Corbin 1971, Oakley 1981, Finch 1984, Burgess 1984).

The interviewees were selected on the basis of their involvement in the Linux project and the Open Source – Free Software movement. We contacted individuals that are recognised by the hacker community, the press and previous researches on the topic as key figures. In addition, we contacted several 'commercial' GNU/Linux distributors and leading software/telecommunications companies (Intekk communications and Lucent Technologies) so as to have a broader picture of the overall management of the 'economic web' (see [Chapter 1: Economic Webs](#)).

This approach is dictated by the fact that this is a cutting-edge research and the interviewees have to be at the forefront of technological and organisational transition/evolution and have influence over their 'surrounding community'.

3.6 SECONDARY SOURCES OF DATA

A large amount of secondary data have been used. In the second chapter, we have drawn upon the organisational literature that identifies the most significant management paradigms whilst tracing back the evolution of the organisation and management. The 'linking thread' is knowledge (access to, sharing, creation, codification, exploitation and dissemination) and its impact that explains the transition to the virtual – network(ed) organisational structure.

In the second part of the literature review, we have analysed important aspects of the software industry that help to understand the behaviour of firms and individuals in the software industry.

As far as the Linux project is concerned, we have used members' biographical writings, previous interviews with key members and descriptions of the group written by other researchers and key figures.

It should be mentioned that Microsoft is covered mostly by secondary data for the following reason: Cusumano's book is acknowledged to be the best research on MS so far, it is academic-oriented, entirely based on primary research and offers a complete cultural and technical overview of the company. Also, to have an alternative view, we selected an insider's book – Drummond's "Revolutionaries of the Empire".

3.7 Framework of Analysis

The framework (criteria) of analysis draws upon key features of major organisational paradigms (ie. TGM paradigm focuses on continuous improvement, 'Learning paradigm' focuses on organisational learning) and how these are managed.

This framework aims at achieving a critical analysis of both Microsoft and the Linux project not based only on one criterion (ie. ability to innovate or flexibility) but through a complete investigation of both entities as it stems from the development process. This decision is based on the fact that "the business literature is rife with stories of performance indicators that failed to capture important aspects of a complex setting. These misattributions may occur because of causal connections that no one understands" (Axelrod & Cohen 1999: p139).

Only through a 'total approach' like this, it is possible to fully explore the management strategies and their efficiency and come to realise what management processes can support - empower the development of a virtual organisation and facilitate virtual, decentralised collaboration.

CHAPTER 4: THE LINUX PROJECT

4.1 Free Software & Open Source³

To understand the workings of the software industry and the Linux project, we need to briefly analyse the ideology of hackers, the advent of free software and some events of historic significance.

The term 'hacker' is an euphemism used among Computer Science circles to describe a talented computer programmer.

In 1971, Richard Matthew Stallman (known as RMS), started working at MIT's AI Lab. At the time, the AI (artificial intelligence) Lab was regarded as the heaven for most computer programmers as it gave rise to many innovations, particularly the ARPAnet in 1969 (the ancestor of the Internet). Around MIT, an entire culture was built that defined programmers as scientists and members of a networked tribe because ARPAnet enabled researchers everywhere to exchange information, collaborate and hence, accelerate technological innovation (Raymond, 1998a) After all, one of the motivations for launching the ARPAnet project was to connect communities of computer programmers in order to share their programs and knowledge (David & Fano, 1965, p36-39 and Abbate, 1999 and Naughton, 2000).

Programming for 'hackers' was the ultimate pleasure and nothing but programming mattered. Their home had become the AI Lab. They defined themselves through programming and the hacker culture was their religion (S.Levy, 1984). They were scientists and as such, their creations-discoveries (software) should be available to everyone to test, justify, replicate and work on to boost further scientific innovation. Software is code (or source code) and the code they developed was available to everyone. They could not conceive that software would have been trademarked some day.

RMS was fascinated by this culture and spent nearly ten years at the AI Lab until in 1981 a company called Symbolics hired all AI Lab programmers apart from two: one of them was RMS. The era that software was trademarked had begun and increasingly more hackers entered the payroll to develop proprietary software whose source code was well guarded as a trade secret. He was so frustrated that his community had been destroyed by Symbolics and proprietary software that he decided to embark on a crusade that has

³ The history is of necessity highly abbreviated and we do not offer a complete explanation of the origins of Open Source. For more detailed treatments, see Moody (2001) Browne (1999), Dibona (1999), Levy (1984), Raymond (1999a)

not been matched before or since: re-build the hacker community by developing an entire free operating system. For him, free means that the user has the freedom to run the program, modify it (thus the user needs to be provided with the source code), redistribute it (with or without a fee is irrelevant as long as the source code is been provided with the copy) or redistribute the modified version of it. The term 'free software' has nothing to do with price. It is about freedom (Stallman 1999, p45).

In 1984, he started his GNU (it stands for GNU's Not Unix) project which meant to become a free alternative of the Unix operating system and in 1985 he founded the Free Software Foundation (FSF). To ensure that GNU software would never be turned into proprietary software, he created the GNU General Public License ⁴ (GNU GPL).

The GNU GPL outlines the distribution terms for source code that has been "copylefted" with guidelines from the FSF. Copylefting involves copyrighting a program and then adding specific distribution terms that give everyone the right to use, modify and redistribute the code. The distribution terms of the GPL are 'viral' in the sense that derivative works based on GPL'd source code must also be covered by the GPL. As a result, other programs that use any amount of GPL'd source code must also publicly release their source code. Therefore, GPL'd code remains free and cannot be co-opted by proprietary development (Lighthouse '99 1999, p9). In fact, the Linux OS is licensed under the GNU GPL and uses most of the GNU programs. That is why it is often referred as GNU/Linux.

During his crusade, he wrote powerful, piece-of-art software that could be both used by programmers as programming tools and also provide 'pieces' that when combined would eventually make up his dream: the GNU operating system. His software appeals to many programmers and so, the pool of GPL'd software grows constantly bigger. He remains a strong advocate of all aspects of freedom, and he sees free software as the area he can contribute the most (Moody, 2001, p29).

However, at a historic meeting in 1997, a group of leaders in the Free Software Foundation came together to find a way to promote the ideas surrounding free software to large enterprises as they concluded that large companies with large budgets were the key drivers of the software industry. 'Free' was felt to have too many negative connotations to the corporate audience (Figure 4.1) and so, they came up with a new term to describe the software they were promoting: Open Source (Figure 4.2) (Dibona,

⁴ accessible at www.gnu.org/licenses/gpl.html

1999, p4; Lighthouse '99, 1999 p24). Stallman was excluded from this meeting because “corporate friendly” is no compliment in his book (Computer Weekly 2001, p36).

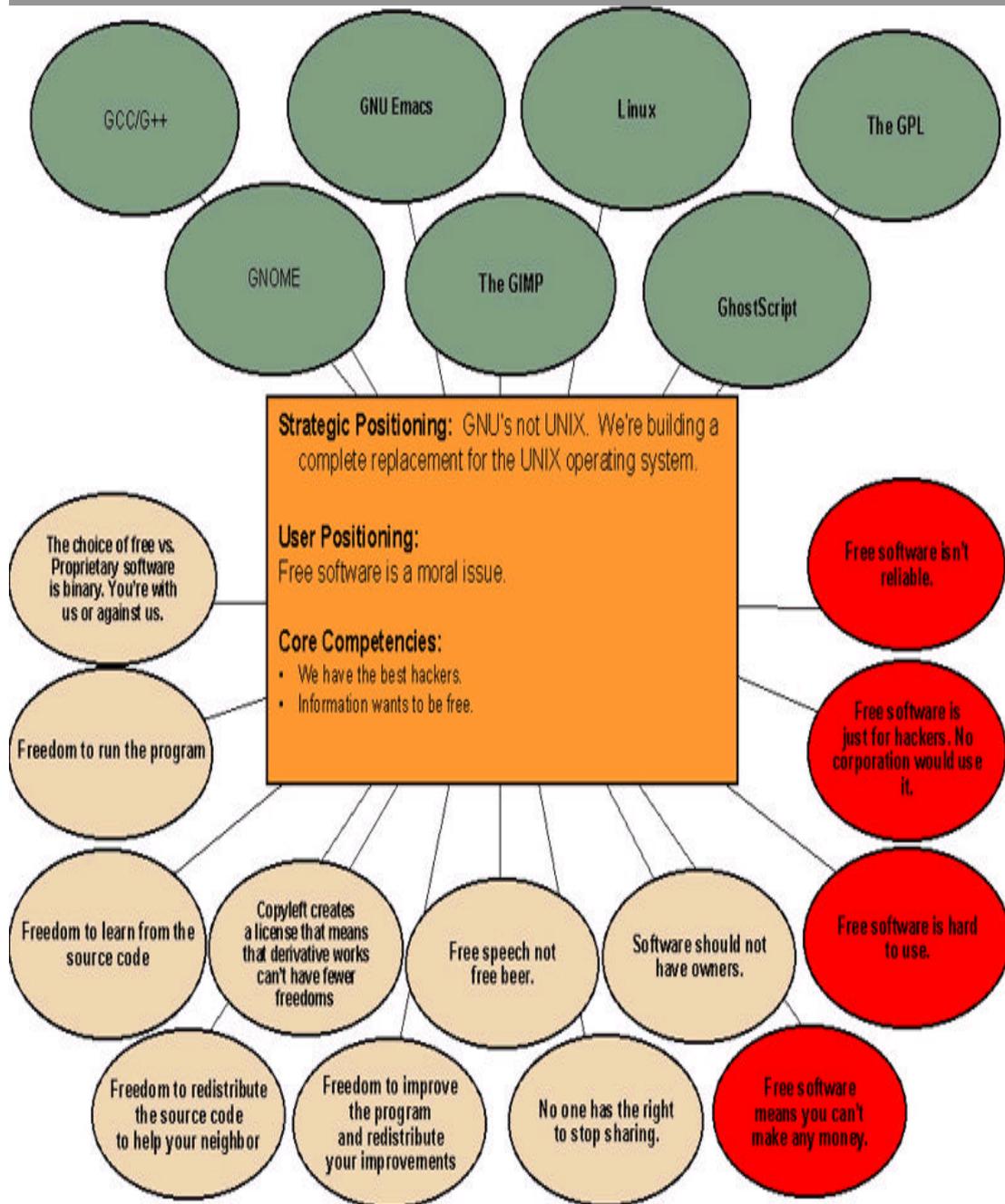
They concluded that an Open Source Definition and license were necessary, as well as a large marketing – PR campaign. The “Open Source Definition and license⁵ adhere to the spirit of GNU (GNU GPL), but they allow greater promiscuity when mixing proprietary and open source software” (Dibona, 1999, p4).

Apparently this strategy has worked wonders since then: key players of the industry (IBM, Netscape, Compaq, HP, Oracle, Dell, Intel, RealNetworks, Sony, Novell and others) have shown great interest in the Open Source Movement, its development and business model and open source products. These days, an equivalent Economic Web around Open Source exists, in which for example, many companies offer support services and complementary products for Open Source products⁶. In addition, there is plenty of favourable press coverage and even Microsoft regards Open Source as a significant competitive threat (Valloppilli, 1998).

⁵ accessible at www.opensource.org/docs/definition.html

⁶ For a discussion of the business models based on Open Source, see Lighthouse '99 (1999), Raymond (1999b) and Dibona (1999)

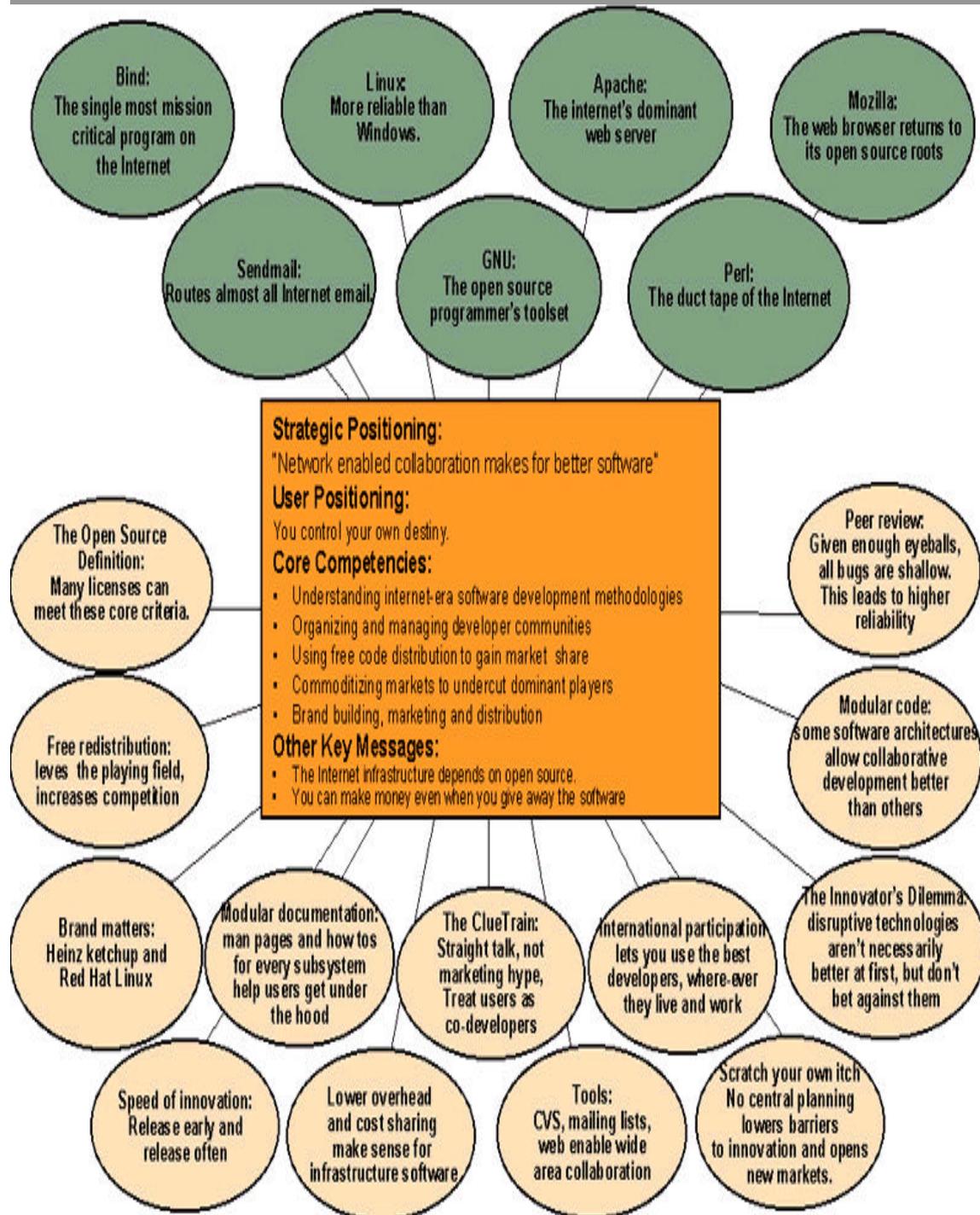
Figure 4.1: Free Software



-The ovals at the top represent the outward face of the movement – the projects or activities that the movement considers canonical in defining itself.
 -The ovals at the bottom represent guiding principles and key messages.
 The dark ovals represent undesirable messages that others might be creating and applying to free software - negative connotations to the corporate audience

Source: T. O'Reilly, Remaking the P2P Meme, Peer-to-Peer, p42, 2001

Figure 4.2 Open Source



The new 'map' puts the Strategic Positioning in a much clearer context: Open Source is about making better software through sharing the source code and using the Internet for collaboration. And the User Positioning speaks of user empowerment instead of moral issues.

The list of core competences is more focused and the negative messages are replaced with directly competing messages that counter them.

Source: T. O'Reilly, Remaking the P2P Meme, Peer-to-Peer, p44, 2001

THE LINUX PROJECT

In 1991, Linus Torvalds made a free Unix-like kernel (a core part of the operating system) available on the Internet and invited all hackers interested to participate. Within the next two months, the first version 1.0 of Linux was released and from then on, tens of thousands of developers, dispersed globally and communicating via the Internet, have been contributing such amounts of code, that as early as in 1993, Linux had grown to be as stable and as reliable as a very powerful operating system. Linux kernel is 'copylefted' software, patented under the GNU GPL, and thus, nobody actually owns it. But more significantly, Linux is sheltered by the Open Source (hacker) community. From its very birth, Linux as a project has mobilised an incredible number of developers offering enhancements, modifications/improvements and bug fixes without any financial incentive. And despite the fact that an operating system is supposed to be developed only by a closely-knit team to avoid rising complexity and communication costs of coordination (Brook's Law), Linux is being developed in a massive decentralised mode under no central planning, an amazing feat given that it has not evolved into chaos. In the beginning it was regarded as a miracle. An operating system is a huge amount of programming work to be done, especially when it is done from scratch as Linux.

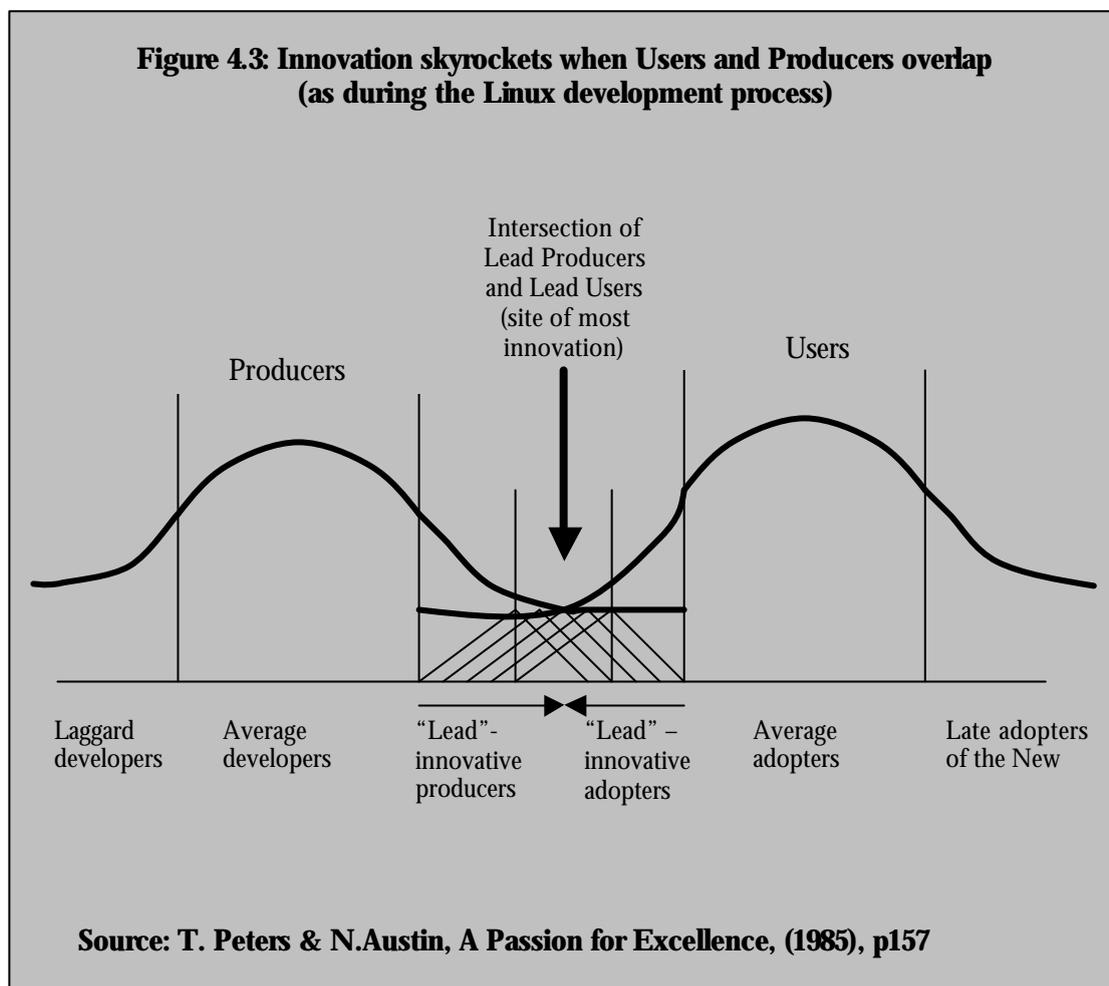
4.2 Innovation

-release early and often: Linus put into practice an innovative and paradox model of developing software. Frequent releases and updates (several times in a week) are typical throughout the entire development period of Linux. In this way, Linus kept the community constantly stimulated by the rapid growth of the project and provided an extraordinary effective mechanism of psychologically rewarding his co-developers for their contributions that were implemented in the last version. On top of this, in every released version, there is a file attached which lists all those who have contributed (code). '**Credit attribution**' if neglected, is a cardinal sin that will breed bitterness within the community and discourage developers from further contributing to the project.

According to conventional software-building wisdom, early versions are by definition buggy and you do not want to wear out the patience of your users. But as far as the Linux development stage is concerned, developers are the users themselves and this is

where most innovation is created (Figure 4.3). “The greatest innovation of Linux is that treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging” (Raymond, 1998b).

Similarly important was Linus’s decision to create a highly **portable** system. Whenever new hardware is introduced, Linux development focuses on adapting Linux to it. “Linux therefore quickly appreciates technological elements and turns them into resources for the Linux community” (Tuomi, 2001 p12-13). Its capability to adapt to environmental changes and continuously learn is unprecedented.



4.3 Structure & Decentralised Development

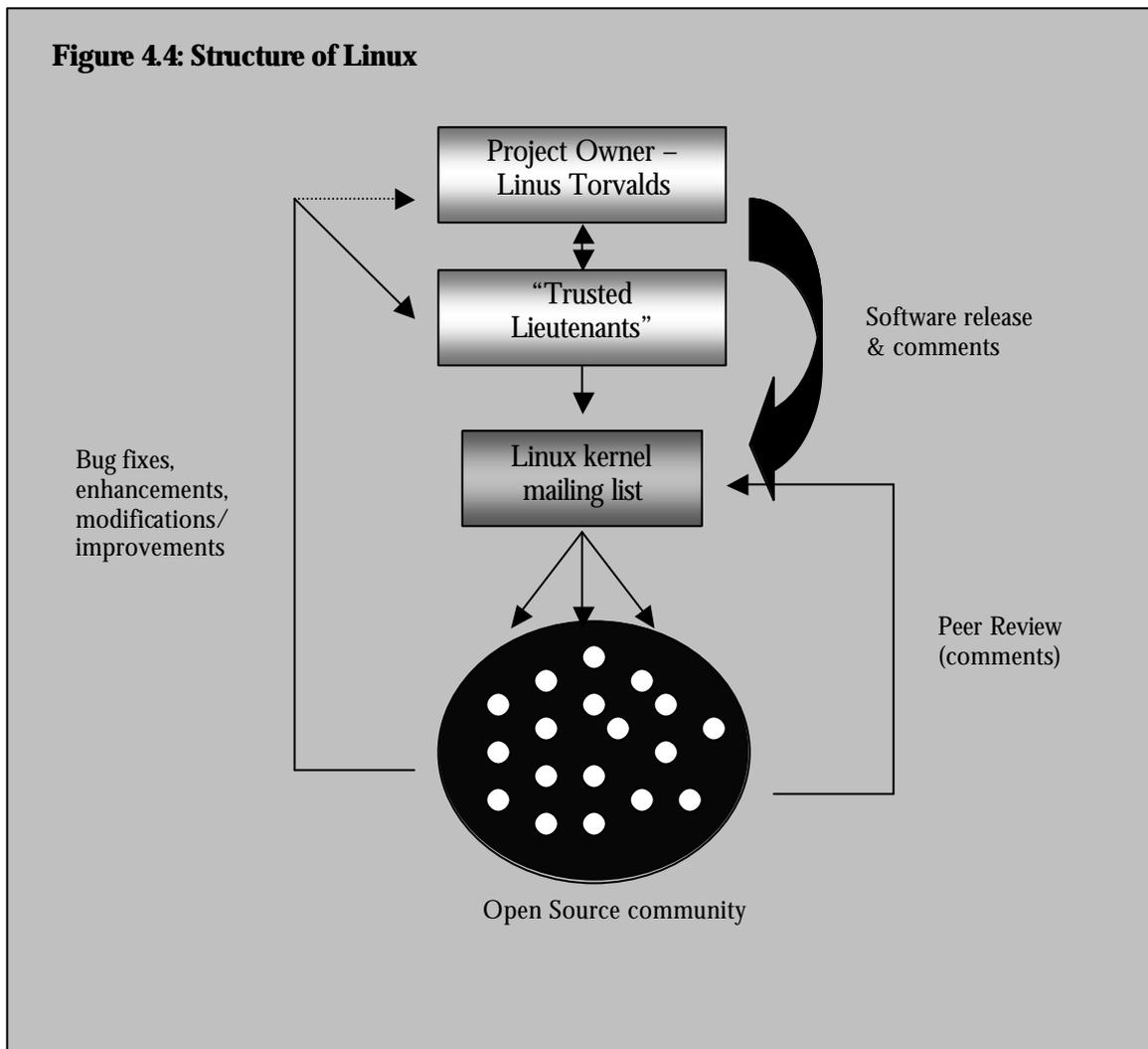
Linus cultivated his base of co-developers and leveraged the Internet for collaboration. The open source self-appointed anthropologist, E. Raymond explains that the leader-coordinator for a 'bazaar style of effort' does not have to be an exceptional design talent but to leverage the design talent of others (Raymond, 1998b).

However, "the Linux movement did not and still does not have a formal hierarchy whereby important tasks can be handled out...a kind of self-selection takes place instead: anyone who cares enough about a particular program is welcomed to try" (Moody, 2001 p62). But if his work is not good enough, another hacker will immediately fill the gap. In this way, this 'self-selection' ensures that the work done is of superb quality. Moreover this "decentralisation leads to more efficient allocation of resources (programmers' time and work) because each developer is free to work on any particular program of his choice as his skills, experience and interest best dictate" (Kuwabara, 2000). In contrast, "under centralised mode of software development, people are assigned to tasks out of economic considerations and might end up spending time on a feature that the marketing department has decided is vital to their ad campaign, but that no actual users care about" (interview with Philip Charles).

The authority of Linus Torvalds is restricted to having the final word when it is coming to implementing any changes (code that has been sent to him). On the other hand, he cannot be totalitarian since everything is done under perfect transparency. Most communication takes place at the linux mailing list (which serves as a central discussion forum for the community and is open to the public) and Linus has to justify all his decisions based on solid technical arguments. The management's accountability is essential and only by earning the community's respect, leadership can be maintained (interview with C.Dibona).

There is only one layer between the community of Linux developers and Linus: the "**trusted lieutenants**". They are a dozen hackers that have done considerably extended work on a particular part of the kernel to gain Linus's trust. The 'trusted lieutenants' are responsible to maintain a part of the Linux Kernel and lots of developers sent their patches (their code) directly to them, instead of Linus. Of course, apart from Linus that has encouraged this to happen, this informal mechanism represents a natural selection by

the community since the 'trusted lieutenants' are recognised [by the community] as being not owners but simple experts in particular areas (Moody 2001 p81, p84) and thus, their 'authority' can always be openly challenged. This does not mean that Linus has more influence than they have. Recently, "Alan Cox (one of the 'trusted' ones) disagreed with Linus over some obscure technical issue and it looks like the community really does get to judge by backing Alan and making Linus to acknowledge that he made a bad choice" (interview with G. Moody).



4.4 Modularity

What made this parallel, decentralised development feasible is the highly modular design of the Linux kernel. "It meant that a Unix-like operating system could be built piecemeal,

and others could help by working independently on some of the various components” (Moody, 2001, p14, p82). Modularity means that any changes done, can be implemented without risk of having a negative impact on any other part of the kernel. Modularity makes Linux an extremely flexible system and propels massive development parallelism (Torvalds, 1999a, pp38-39 & 1999b, p101-11) and decreases the total need for coordination (Moon & Sproull, 2001, p6).

4.5 Motivation

Open Source programmers write software to solve a specific problem they are facing, ‘scratching their personal itch’ as E. Raymond points out (Raymond, 1998b) and once they have done so, they can choose either to sit on the patch or give it away for free and hope to encourage reciprocal giving from others too (Raymond, 1999b). But it is the ‘hacker ethic’ and the community dynamics that provide strong incentives to contribute. First of all, the process of developing such software is quite pleasurable to them. “They do not code for money, they code for love” (interview with G.Moody) and they tend to work with others that share their interest and contribute code. They enjoy a multiplier effect from such cooperation (G.Prasad, 2001) and “when they get the opportunity to build things that are useful to millions, they respond to it” (interview with R.Stallman).

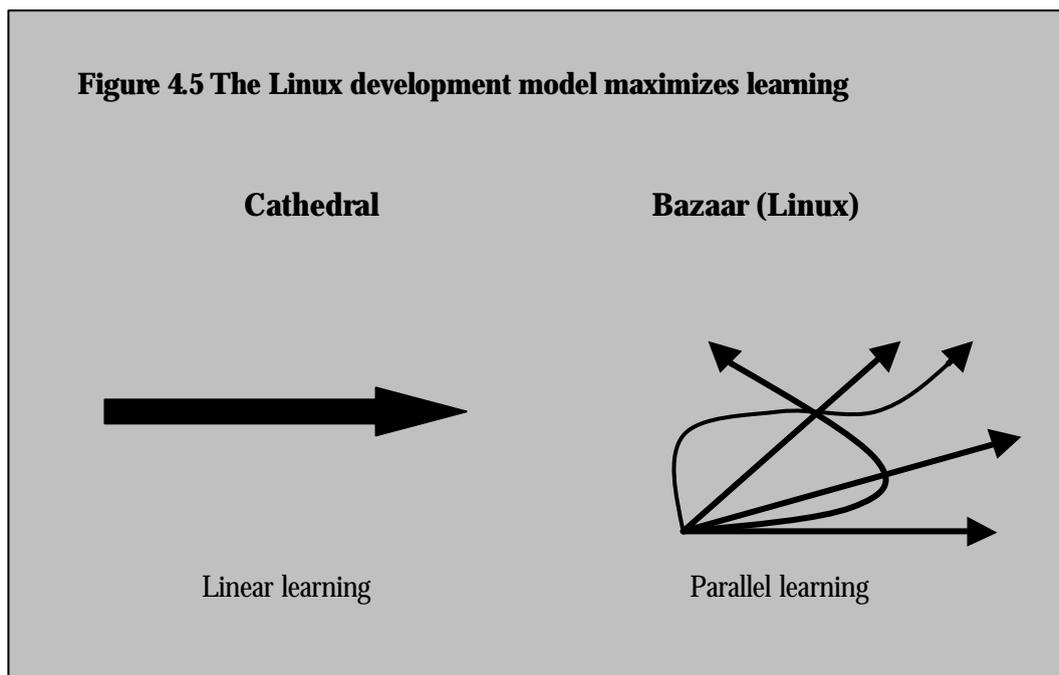
In addition, the Linux community has dealt with the lack of centralised organisation through an implicit reputation system that characterises the Open Source community (Kuwabara, 2000). O.S. programmers have always been unconsciously keen on gaining the community’s respect (‘seek peer repute’) and the best way to do so is by contributing to an already existing project that is interesting enough or has created enough momentum within the community (Raymond, 1998c).

Interestingly, there are also people motivated to contribute because of indirect monetary returns. These programmers fall into four main categories (Lee & Cole 2000: p21):

- a) they work for a commercial Linux distributor (ie. Red Hat) or a company that provides complementary services-products,
- b) their work involves occasional Linux programming (ie. their employer’s information systems – computer infrastructure relies on Linux),

- c) their involvement with Linux makes them more attractive (employable) to potential employers and,
- d) they develop applications that run on Linux in hope of encouraging Venture Capitalists to fund their start-ups or receiving ownership rights (ie. shares) in start-ups.

However, these financial motives are indirect since no one is paid by the Linux project or by its leader – Linus Torvalds.



4.6 Learning

Massive parallel development is evident in the case of Linux. “In a conventional software development process, because of economic and bureaucratic constraints, parallel efforts are minimized by specifying the course of development beforehand, mandated by the top. In Linux, such constraints are absent since Linux is sustained by the efforts of volunteers. In the ‘Cathedral’, parallelism translates to redundancy and waste, whereas in the ‘Bazaar’, parallelism allows a much greater exploration of a problem-scape for the global summit and the OSS process benefits from the ability to pick the best potential implementation out of the many produced”. Thereafter, (apart from quality and innovation), social learning is maximized in a ‘bazaar style’ of development (figure 4.5) (Kuwabara, 2000 p28, p29; Nadeau, part 1, p8).

Linux has a **parallel release** structure: one version (1.2.x and 2.0 series) is stable and 'satisfies' the users that need a stable and secure product, and another version (2.1.x series) is experimental and 'cutting-edge' and targets advanced developers (Browne, 1998 p3; Moon & Sproull, 2001, p6).

The significance of the parallel release structure should not be underestimated. All decision-making situations face a trade-off: **exploration versus exploitation**. This principle states clearly that the testing of new opportunities comes at the expense to realising benefits of those already available and vice versa (Holland 1975; March 1991; Axelrod & Cohen 1999). The obvious organisational implication is that whenever resources are allocated to the search for future innovation, resources dedicated to the exploitation of existing capabilities are sacrificed and vice versa. This trade-off is fundamental to all systems and vital to their ability to evolve and survive; a favourable balance is therefore critical (Kauffman 1993).

Surprisingly, this trade-off is not applicable to Linux: the parallel release structure ensures that both present and future (potential) capabilities are harnessed and this renders Linux extremely adaptive to its environment and also ensures the highest level of innovation and organisational learning.

This massive parallelism (and parallel learning) has resulted in a more secure, reliable, defect-free operating system. One of the most critical stages of software development is characterising the bugs (detecting any defects through testing) and then debugging the software. The massive parallelism has rendered Linux 'untouchable' as all bugs are quickly identified. E. Raymond terms this phenomenon: 'Given enough eyeballs, all bugs are shallow' and he points out that security is an aspect of reliability. And such reliability can only be achieved through massive and parallel peer review. The paradox is that security is directly dependent upon the transparency of the source code and security can only be compromised by non-open code (Schneier 2000).

4.7 Management of the Economic Web

In the last few years, many organisations have been built around the GNU/Linux operating system and other open source software. They mostly offer support services (ie. maintenance, training), complementary products and packaged distributions. Most of them that are in a favourable financial position, spend money on R&D related to open source projects and contribute in any way they can in order to remain dominant players in the open source software market and for goodwill towards the open source community that provides them with a quality product to start with (Young 1999, p105). Nowadays, increasingly more 'big players' are joining the web: IBM, Dell, Oracle, Intel, HP, SAP and others have been tantalised by Linux and its open source development model, that have started investing heavily in the 'Linux platform'.

The management of this web depends heavily on the fact that every member of the web does not place any restrictions or rules on the other participants. Linux as a project community focuses on the software development process and does not interfere in any way with the businesses built around it. The key factor may be the open source code that enables a common computing and communications infrastructure to be established more easily rather than with a 'closed source code' based technological platform. E. Raymond attributes that to the issues of trust and symmetry – “potential parties to a shared infrastructure can rationally trust it more if they can see how it works all the way down, and will prefer an infrastructure in which all parties have symmetrical rights to one in which a single party is in a privileged position to extract rents or exert control” (Raymond 1999b, p15; interview with M. McConnel).

In all, the massive network effects that pervade the software industry trigger further growth and adoption of the GNU/Linux OS (Figure 4.6) which in turn might lead to lock-in of the market under this platform and establish Linux as the dominant operating system (interview with C.Dibona).

CHAPTER 5: MICROSOFT Vs. LINUX

Table 5.1: Microsoft Vs. Linux Project

	MICROSOFT (PHYSICAL)	LINUX PROJECT (VIRTUAL)
Business Processes (development)	Linear	Parallel
Cost of development	High	Low
Cost of coordination	High (Brook's law holds)	Low (Brook's law doesn't)
Mode of organisation	Centralised	Decentralised
Management	Hierarchical	Collaborative- community Emergent (leadership)
Hierarchical layers	Several	Four but not in a bureaucratic fashion
Modularity	Low	High
Knowledge functions (access, sharing, diffusion, creation, exploitation)	Low	Massive
Organisational learning	Linear	Parallel
System	Closed	Open
Users – producers	Separated	Overlap
Number of participants	Limited	Unlimited
Product transparency	Absent (copyright)	Massive (copyleft)
Decision-making		
transparency	Low	Massive
Product innovation	Low	High
Organisational innovation	High	Massive
Management of the economic web	Offensive – successful	Collaborative – successful
Cost of platform	High	Low
Flexibility of platform	Low	High
Use of standards	Low	High

5.1 Business Processes

The critical business process in both Microsoft and Linux is the development process. Microsoft is characterised by centralised development since most activities take place in one site: Redmont, Seattle. Brook's Law* holds strongly in every physical software

* Co-ordination costs and complexity grow with the square of developers but work done rises linearly

development model and MS is not an exception. Therefore, the cost of co-ordination is high.

On the other hand, Linux is synonymous to decentralisation since the project is developed by thousands of dispersed people who collaborate under no central planning. It defies Brook's Law because of its parallel release structure, extreme modularity, 'trusted lieutenants' structure and as a consequence, co-ordination costs are almost negligible.

Microsoft has elaborated on a development model (synch-and-stabilize) that may be identified as parallel when compared to the conventional 'sequential' model (**see Figure I: Synch-and-Stabilize Life Cycle & Table II: Synch-and-Stabilize Vs. Sequential Development**) but is still far from comparable to the incredibly parallel development model of Linux. This is partly attributed to the Linux parallel release structure and mainly because of Linux being an open system that encourages anyone to participate. This is not the case with MS that naturally 'encourages' (employs) only a tiny fraction of the programmers willing to participate (to be employed) in the face of financial constraints and increasing co-ordination costs.

This massive parallelism that characterises the Linux Project allows for an ideal allocation of resources towards both exploitation of existing advantages and exploration of potential future opportunities and results in skyrocketing organisational learning and innovation (both product and organisational) that in turn, renders Linux extremely adaptive to any environmental changes.

5.2 Management, Structure and, Knowledge

Microsoft's management is hierarchical as all the strategic decisions are made by Gates and 'the President's Office' and employees are discouraged to bypass their superiors. Microsoft's structure is built on this premise (see **Figure II: Microsoft Scalar Chain of Command**) and political manoeuvring within the company is not rare. And despite the fact that frequent job rotation and project-form of organising are the norm, information flow is restricted mostly due to politics and therefore the only value extracted out of the knowledge functions is the twenty years experience that MS has of the industry. Of

course this does not mean that MS maintains 'positive' relationships with its surrounding environment since its arrogant management of the economic web has seriously damaged the information exchange between them. The effect on organisational learning and innovation is certainly negative.

In total contrast, Linux relies on four structural layers that are comprised by the leader (L.Torvalds), the 'trusted lieutenants', the 'pool of developers' and the open source community. Even though the community layer does not appear to be part of the Linux structure at first glance, it is in fact incorporated in it and has significant influence over the project. The pattern of the layers cannot be said to be vertical but rather horizontal if there has to be a hierarchical classification.

This structure is not paradox if we consider the technical objectives of Linux as a project: quality and flexibility. For such quality and flexibility to be achieved, a system ensuring the highest quality and flexibility had to be in place. This mechanism takes shape in the form of emergent leadership and emergent task ownership which implies that all decisions and hierarchical positions can be openly challenged so as the best decision is reached and the most capable individual is chosen. Naturally, the prerequisites for such a style of managing are product and decision-making transparency and Linux has embraced both from its inception.

Consequently, this transparent 'collaborative – community centred' management is by far the greatest innovation of Linux and the key enabler for limitless value extraction out of the 'knowledge functions' since all organisational members including the surrounding community are encouraged to freely share information.

At last, the nature of Linux renders it attractive to the economic web because the platform (on which the economic web is based): is flexible (easily customisable), transparent (non-proprietary), the platform's cost is low and build on common standards. These features of Linux as a technological platform reduce uncertainty in the light of technological upheaval and enable investment in Linux (by the economic web) to take off. Evidently, the non-proprietary platform presents an opportunity for 'collaborative management of the economic web' and to date, the Linux Project reaps full benefit out of this opportunity.

CHAPTER 6: THE NEW PARADIGM

PART A

TRANSFORMATIONS OF MANAGEMENT

6.1 Management can be digital and networked

Because of enabling technologies, management can be successfully networked as long as the human intelligence that makes up the organisation is networked too. “The intellectual power generated through networking minds for collective vision will far surpass the intellectual prowess of the smartest boss. Equally important, strategies developed collectively have higher probability of being implemented” (Tapscott, 1996). Collective thinking leads to collective action.

If the people that participate in a project are being supported by an appropriate technological infrastructure, management can also become entirely non-physical. With the advent of tele - immersion and related technologies, time and space will no longer be important. However, organisations do not have to wait for wide scale rollout of advanced complex technologies. Despite the technical prowess of Linux developers, use of simple and reliable technologies (ie. e-mail, file transfer, newsgroups) was common in the Linux project. This clearly shows that the management does not have to be a “PhD in computer science”, simply to deploy the already existing tools in the most efficient way to facilitate communication between large groups. The most progressive technologies will not deliver if they are not simple (to handle) and ‘common protocols-standards’ (to ensure interoperability and compatibility) (Interview with K.Kuwabara).

Of course, the organisational design has to change. In a hierarchical organisation, the employees are vertically or horizontally linked to each other and information rarely bypasses more than one layer, and therefore there is no fertile ground for the management to be networked.

Let’s consider the Linux structure again ([Figure 6.1](#)). It is evident that the flow of information is networked. Whenever a developer makes an inquiry, every other member has access to the inquiry and can answer back. Or whenever someone engages in a

conversation, the entire conversation is so transparent that anyone can intervene and participate. The Linux mailing list serves as the disseminator of information and links all the members together.

Obviously, for this structure to be efficient, **management and decision-making should be based on merit** (and not on status) **and be transparent** throughout the entire organisation.

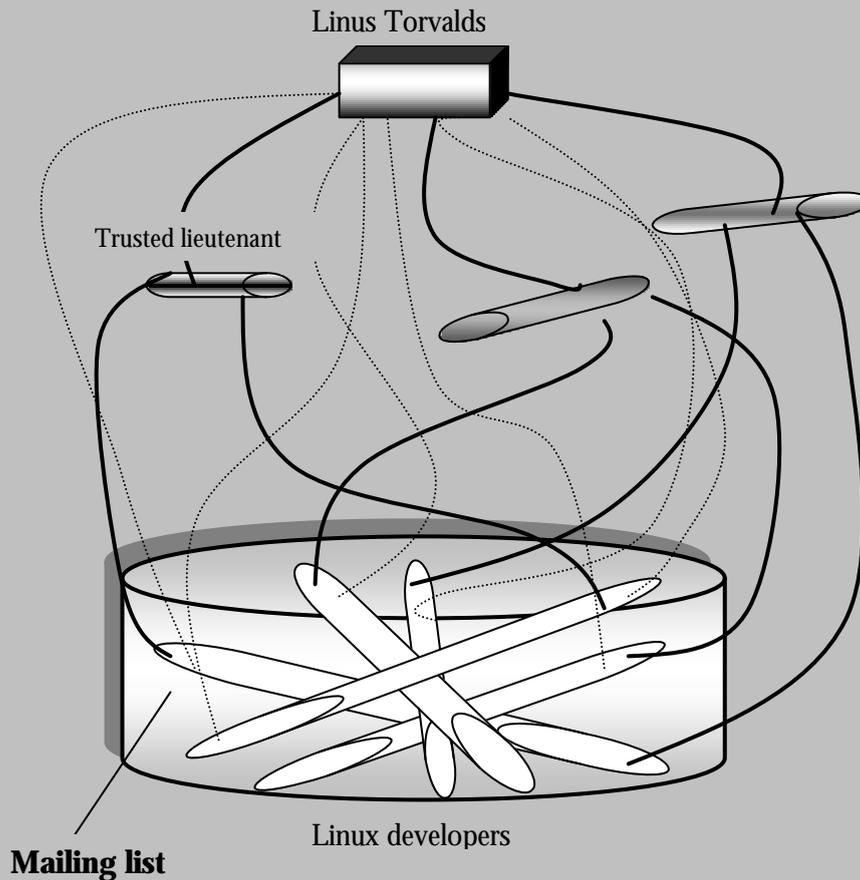
In the case of Linux, the selection of developers and the changes to be implemented are purely based on solid technical grounds (merit). This indirectly achieves control over product development (and eliminates the need for scholastic control over the employees) and ensures the highest level of quality. In addition, every decision made (by Torvalds) has to be clearly and logically justified and in case there is no catholic consent, the community is able to challenge it.

It is increasingly the role of management to break free from any restraints related to bureaucratic regimes, and empower the customers (both internal and external) of the organisation to participate in all crucial decisions. Issuing orders and direct exercise of control must be replaced by communication of vision and direction. Management becomes leadership. And this explains why a project such as Linux which operates and grows organically under no central planning needs a leader like Linus Torvalds: to initiate change, communicate vision, and “create a organisational mindset” where network communication is fostered.

To continue, the product should be ‘transparent’ too. Products under development should be accessible to all organisational members to use, test, fix problems and build upon features. Product specs that are guarded under secrecy, do not add any value to this purpose. The organisational equivalent of ‘copyleft’ (GNU GPL) will enable distributed collaboration. All organisational systems should reward and encourage sharing of information⁸.

⁸ see D.Constant, S.Kiesler, L.Sproull, 1996, The kindness of strangers: on the usefulness of weak ties for technical advice, Organization science, vol7, no2, p119-135, for discussions of incentives for information sharing in organisations

Figure 6.1 Linux structure depicted as flows of information between value streams



Source: Adapted from J. Martin, Cybercorp, p93, (1995)

This structure shows that the 'trusted lieutenants' is a support value stream/function and not an additional layer since the developers (can) go directly to L.Torvalds, and the developers are the primary value stream. This is a pure network structure as all the participants have access to all the nodes of the system. Accordingly, the flow of information is 'networked' since all stakeholders have access to every occurring communication through the mailing list.

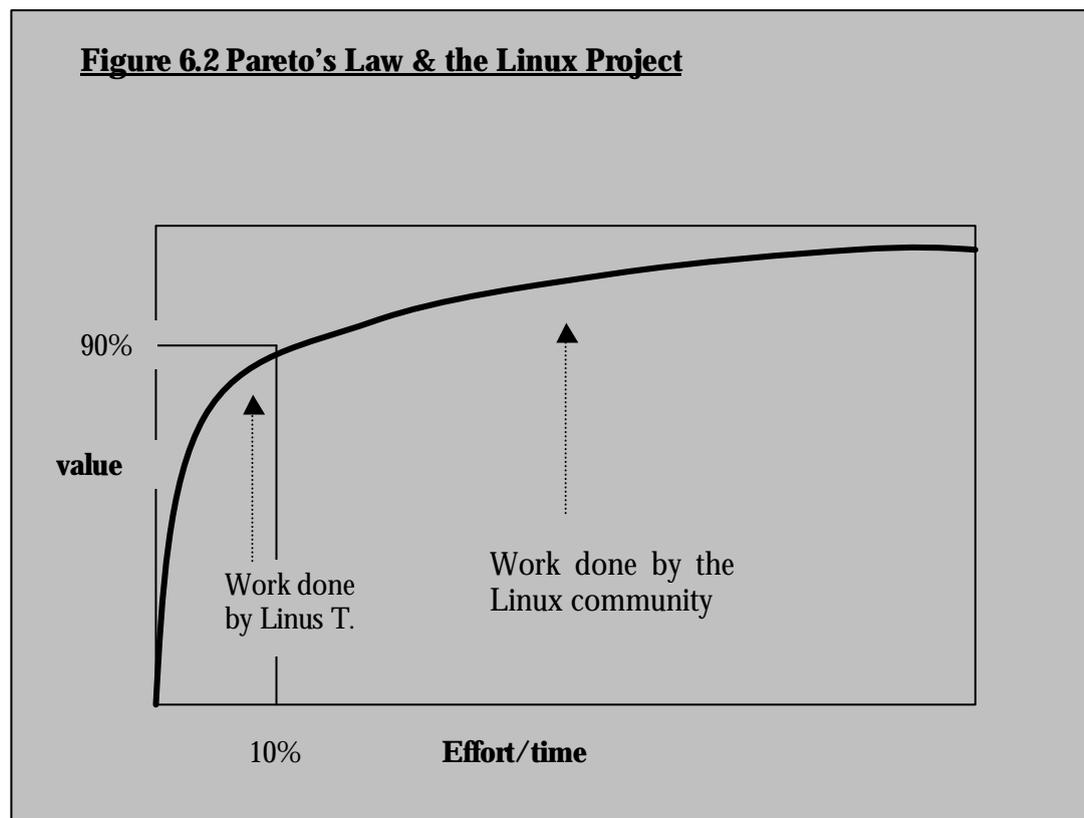
6.2 Management should ensure that the organisational and project design maximizes organisational learning and empowers big teams to collaborate digitally.

The organisational design must enable the human intelligence that resides within the organisation and its environment to get networked. With creative use of the technology, knowledge will be disseminated throughout the network towards all the participants (organisational learning), and not just exchanged among few nodes (individual learning). As the Linux project proves, massive parallel learning is not just a matter of technology but of design and management. The project design must be as **modular and simple** as possible to facilitate digital collaboration (between an incredibly big project team). It is more important that the design is simple and modular than error-free.

Task decomposition is established as an intelligent organizational principle (March & Simon 1958; Baldwin & Clark 1997) and modularity is an extreme form of task decomposition. Modular task decomposition reduces coordination costs to the minimum. Also, it allows multiple groups to work on the same module independently. In the end, the best work is adopted (Moon & Sproull 2000). Caesar's 'Divide and Conquer' strategy has not lost its effectiveness.

Hence, it is equally important for organisations to embrace parallel development. By initiating a **parallel release structure** (in projects), both stability and rapid development are ensured. Usually, organisations face a trade-off between exploration and exploitation (see page 45) and this why linear structures have been preferred. But, “parallel release structures could generate more innovation and continuous improvement” (Moon & Sproull 2000).

Let's consider what could be accomplished in this way. In many business situations and in software development, Pareto's Law applies: 90% of the value/functionality takes 10% of the effort/time. “One excellent thing about Linux is that the originator spent the 10% to produce a working product (highly modular and simple), released it, and has the community do the 90% to finish it off – a massive saving” (interview with Philip Hands) (Figure 6.2).



6.3 Management focus shifts from Organisational Dynamics to Economic Web (Network) dynamics

Managers should focus on the mechanisms (network effects) that govern the market their organisations are in. The first step is to identify the economic web(s) since success or failure is often decided not just by the company but by the success or failure of the web it belongs to. Active management of the economic web can be a decisive factor upon an organisation's future (Hagel 1996).

The conventional strategy requires organisations first to define their own strategies and then to proceed to alliances. But in industries that are built around technological platforms, the management confronts two basic choices: which economic web(s) to participate in (or form) and what role (shaper - the owner of the technological platform or adopter - built around the technological platform) to play within them. Corporate strategy follows web strategy (Ibid).

PART B

IDENTIFYING THE NEW PARADIGM

6.4 The emergence of a new paradigm?

So far, this paper has highlighted factors and structures that support decentralised development. The critical questions though still remain: **Under which circumstances is the Virtual Organisation a rational organisational model, capable of unlimited growth and competitiveness and how can it be created and sustained?**

This part will attempt to answer these questions.

We should mention that virtual organisations do not yet exist and maybe will never. The Linux project on which the virtual organisation is modelled on in this study shows why: the first problem is our perception of the organisation. The conventional perception is that the organisation aims at high profits by selling as large quantities of products or providing as many services as possible and to achieve this goal, it deploys a variety of tactics such as costs reduction, political manoeuvring, marketing campaigns, expansion of product portfolio, economies of scale and scope and so on. This paradigm relies on companies to deliver profits and as long the company is profitable, it is sustainable.

The emerging paradigm is about creative people enacting creative projects that in turn mobilize creative people. The emerging organisation paradigm is not a company but a PROJECT.

No boring project will mobilize the necessary “talent pool” – the creative human resources (interview with G.Prasad, G.Moody, P.Hands) and one of the crucial success factors of the virtual organisation model is the massive decentralised collaboration that demands abundant brainpower – a ‘legion of workers’.

For a project to be creative (so as to **mobilize critical mass of resources**), four conditions must be met:

- a) The users and the implementers overlap or communicate directly without interference of a filtering layer-mechanism – implementers and users have common objectives
- b) The organisational ‘copyleft’ is put into place to reward and encourage transparency and sharing of information
- c) Deeply rooted trust within the surrounding community – network
- d) Fluid networked organisational structure that gives rise to positive hyper competition and meritocratic self-selection of task owners (leadership is not imposed from the top but emerges from the bottom): **motivation to attract creative people**

6.5 Motivation is the source of sustainability

Traditional organisational structures are based on deadlines, payroll roles, fixed positions, etc. and this is unlikely to attract creative people (interview with G. Prasad). Creative people are excited by fluid structures that provoke competition among peers. The incentive is to differentiate yourself by rising above the standard level (through achieving excellency) and gain the recognition of your peers that regard you as a leader. This is the ultimate form of this motive and recognition by your boss in a bureaucratic environment is the least.

This motive is parallel with achieving a communal goal (your peers are thriving to achieve the same goal) and even though this motive is partly fuelled by the egoistic, competitive nature of creative people, it also ensures the highest level of quality (only the best work will be selected in a democratic process where all organisational members have ‘equal vote’ and excellency is the higher goal) and leads to continuous and rapid organisational evolution (since leadership is constantly challenged by equally creative people).

This positive hyper competition sustains the project and not profits. Also, it enables the benevolent dictator’ and ‘trusted lieutenants’ structures to emerge.

The next question is how, when and where these preconditions will come together to create THE CREATIVE PROJECT – the virtual organisation. The answer is obvious if we consider the Linux project from a different angle ([Figure 6.3](#)).

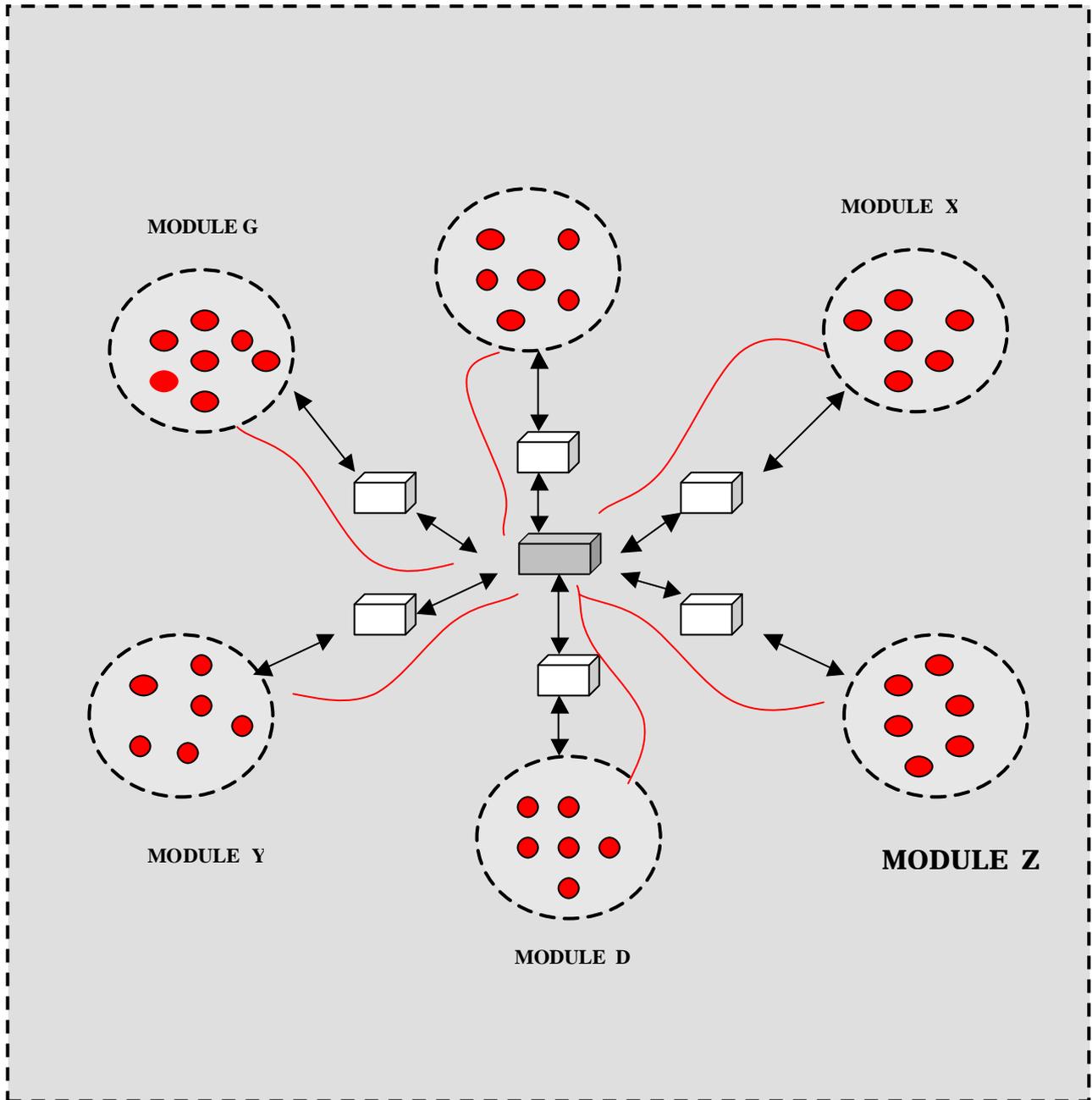
6.6 The Virtual Roof

The Linux project is characterised by flow of information within the entire organisation, implying among the thousands of developers, Linus Torvalds and the 'trusted lieutenants'. All of them have access, can engage into all occurring conversations and can communicate directly with every other member – implementer through discussion forums, mailing lists, newsgroups. But the flow of information is not restricted only within implementers but it extends to the global community reaching virtually everyone interested, including commercial companies (ie. Sygnus Solutions, SuSe, Red Hat, VA Linux) that provide support services and packaged distributions, computer scientists that may have not been involved directly (as implementers), companies that consider adopting open source software for their own internal use, users that need help from experts and anyone interested or curious enough to observe or even participate in any given conversation. **Access to the various open source-related web sites, discussion forums, etc. is open to the public and all interested parties.** Figure 6.3 shows that communication and information flow is so pervasive that spreads equally towards all directions and is diffused throughout the virtual roof.

Throughout the research, it became evident that the element encapsulating the view of the interviewees about the Linux Project and its abstract virtual boundaries is what the author identifies as the 'Virtual Roof'.

The virtual roof is the common denominator of the users, the implementers and the surrounding community and acts as intermediary (as a trusted third party) to ensure **trust** in a largely anonymous virtual marketplace by providing an electronic platform whereby network communication is nurtured.

Figure 6.3: The Linux Project & The Virtual Roof



The lines and the arrows represent information flow and specific project functions (transfer of patches - source code) whereas information flow pervades the virtual Roof and is diffused towards all directions

-  Information Flow & Transfer of Code
-  Information Flow & Diffusion (The Virtual Roof)

However, the virtual roof has no power to delegate authority or abuse its power in favour of either the implementers, the users or the surrounding community by establishing asymmetries of information (interview with C.Dibona).

The main reason is its transparent nature (open to the public and all “transactions” are visible) and that is based on the mutual realisation (among the various members) of the long-term benefits that are impeded by a potential abuse of power and asymmetries of information. This is the **organisational ‘copyleft’**. Deep down, it is the system of ‘lean production’ modified to create trust in the public, hence it extends its reach to include also the users and the global community.

All the virtual roof can do, is enforce common standards – rules of practice (that facilitate collaboration and decentralised development) that are approved by the community.

In a virtual roof, people can gather, talk, share ideas, spread information, announce projects and get community support (interview with D.Barber). These online communities once formed, make new relationships possible: **direct user – implementer interaction**. The ‘implementers talking to the users mechanism’ is possible in any project depending on one thing: the level of dedication by the surrounding community TO THE PROJECT (Ibid), and this is why virtual roofs exist: to establish trust, enable critical relationships that would be otherwise impossible on any significant scale and orchestrate decentralised efforts towards a centralised goal.

Virtual roofs dedicated to global (user-oriented) communities are essential to the life of projects whose success depends on the dedication of the surrounding global community and utilising geographically dispersed resources under no central planning. In such places, leadership is emergent and has a dual objective: give direction and momentum to the project.

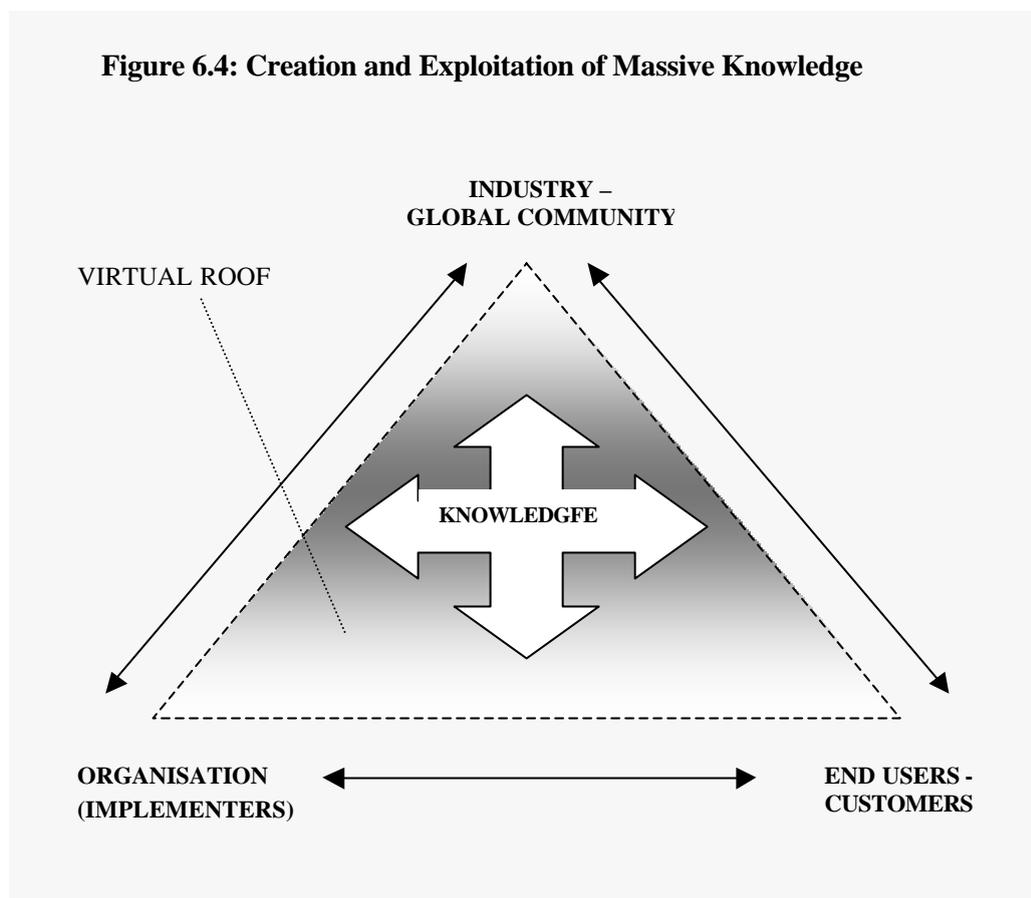
The last question deals with the unique competitive advantage of this organisational model and under which circumstances it is a rational organisational design.

6.7 Knowledge is the competitive advantage

To date, organisational thinkers had been mainly concerned with how the 'virtual organisation' will carry out the same identical functions –activities with the 'physical organisation'. Literally, how can the virtual organisation duplicate exactly what the physical organisation does?

This is a wrong question to start with. Perhaps, a more appropriate question would be what could the virtual organisation do that the physical organisation cannot (replicate)?

The answer lies on the fact that by bringing the organisation (implementers), the surrounding global community – industry and the end users – customers together, it can generate massive knowledge and exploit it in the most effective way (Figure 6.4).



It should be noted that the functions of knowledge (sharing, creation, diffusion and storage) are achieved through the high mobility and interaction of the human resources within the virtual roof and not within the organisation (core project group implementers) (Figure 6.4). The virtual roof is the device for generation of knowledge and the organisation is the device for the most efficient exploitation of the knowledge created. Thus, the separation of the organisation and the virtual roof will destroy this competitive advantage.

6.8 Rational Organisational Design

This model is likely to be the most rational organisational design when there is need or willingness:

1. To avoid potential fragmentation of the market (vendors offering incompatible versions of the same product that is likely to slow down the pace of innovation in the industry or harm the desired convenience or quality that is sought by the users)
2. To dethrone a product or vendor that has locked – in the market due to increasing returns mechanisms and is effectively having negative effects on the evolution of the industry and its long-term profitability
3. To manage the ‘economic web’ in a way that all participants will be well equipped to seize benefits or profits and keep the market highly competitive
4. When the users’ contribution (ie. knowledge, testing of product) is invaluable if not necessary during the development phase or even before, so direct interaction between implementers and users is important
5. Bring together a global community - industry on equal rights to discourage otherwise prevalence of political manoeuvring and corporate backstabbing
6. When the issue (project under development) is so complex and has over reaching consequences for the market that requires co-operation and/or generation of knowledge which cannot be created by just a single organisation

PART C
APPLICABILITY OF THE ‘LINUX MODEL’ TO OTHER
INDUSTRIES

6.9 Implications

“Standardisation within industries is about making (virtual) decentralised development possible, provided the ‘interfaces’ between the different modules are well specified” (G.Moody).

But there are differences and various obstacles to overcome.

“Many parts of Linux function completely separately, which makes the task much simpler. Equally, it is much easier to ‘hack’ a piece of software code than to ‘hack’ a wheel, so making even small modifications is always possible. This renders distributed software development a much greater flexibility than is probably possible for other industries” (Ibid).

Another difference lays on the nature of software. “Software can be copied endlessly and perfectly at negligible cost. With the advent of nanotechnology such problems might evaporate but the energy problem still remains, whereas the only energy software requires is a programmer’s time and effort” (Ibid).

Another issue is that in the case of Linux, the users were the producers themselves. Instead of having the information from the users filtered through five or six layers before it gets to the implementers, the implementers talked to the users and could find exactly what it is they need. “The open source communities may be unique in the closeness of the developers and the users and this is probably an argument for adopting slightly different approaches towards openness, but in some ways it is more a matter of attitude or respect rather than anything else” (Ibid).

Similarly, “the ‘trusted lieutenants’ structure is highly unlikely to be viable or successful in a traditional corporate setting at which the one who pleases the boss is the one who stands the best chances of climbing up the hierarchy whereas the most zealous but ‘less

connected' employees will be passed over in the absence of meritocracy" (interview with D.Barber).

"Attitude or respect" drops a hint about the tectonic politics that characterise the majority of organisations and the issue of funding for a project to get started. "When projects get big enough to need dedicated management, whoever is holding the purse strings is going to reject the notion that it is a loosely-organised anarchy" (interview with C.Browne). In contrast, when it comes to Linux, the necessary investment is just a person's time.

Perhaps, the greatest problem is the nature of the organisation. "An organisational copyleft will be extremely hard to be accepted by shareholders in the era of well-guarded "intangibles" as a potential competitive advantage. On the contrary, organisations answer to shareholders and shareholder pressure is not likely to result in co-operative values and intra-industry sharing of information" (interview with G.Prasad).

Another restraining factor may be the regulatory arrangements that often mandate that an enterprise should operate in a "centralised hierarchy" manner. These 'arrangements' may stem from a body like the FAA telling the organisation that they must follow certain procedures to comply with the law or, a union telling management that they need to have clearly defined job descriptions and hierarchy showing the exact delegation of authority and responsibilities within the organisation (interview with C.Browne).

Also the adoption of a new organisational model must be considered with regard to the project's level of complexity. "For a project so complex yet so focused as Linux, decentralisation turned out to be a major driving force behind its success. But when it comes to a much simpler task, decentralisation might only undermine organisational coordination" (interview with K.Kuwabara, also see Appendix III: Communication Networks).

Critics claim that non-virtual, centralised organisations are better at storing know-how – they serve as a repository of knowledge – and that 'social capital' is what keeps an organisation or a network together⁹. As far as Linux is concerned, the ideal reservoir of

⁹ see C.Leadbeater, 2000, Living on thin air, Penguin for a more elaborate discussion

knowledge is the freely available source code and the social capital (ie. trust, common values, reciprocal giving) that serves as the 'integration link' and ties together the entire Linux community network is abundant since all the developers share the hacker mentality-ideology.

"In virtually - networked organisations, suitable mechanisms for storing and extracting value (out of knowledge) will emerge for undertaking these two functions. Film production is a good example where know-how is embodied in individuals' skills and their high mobility makes sure that know-how is quickly disseminated throughout the network – thus, value is extracted out of their interaction with the network" (interview with R.Laubacher).

So far and until these mechanisms emerge, apart from a few examples (ie. Linux, Silicon Valley, Hollywood), "knowledge creation is best done in an evolutionary (decentralised) fashion, while knowledge storage is best done in a centralised mode" (interview with K.Kuwabara).

The issue of social capital and mostly trust also presents an obstacle. "One of the interesting trust models – which is how open source works – consists of webs of trust. You trust the people whom the people you trust trust. After all, centralised trust systems have always been inherently risky" (interview with G.Moody). And the absence of trust is not an option as it destroys this model.

CHAPTER 7: CONCLUSIONS

7.1 Epilogue

The virtual organisation in this study is modelled on the Linux project and thus, the virtual organisation cannot be a company but instead a project that is inclusive enough to embrace an entire global community including the end users, the implementers of the core project and the organisations that compete within the boundaries of the industry. The key technological enabler is 'virtuality' – what we have dubbed as a 'virtual roof' – and the key organisational enabler is the mutual understanding from the global community of the long-term benefits that will arise from co-operation and uninhibited sharing of information.

Beyond the shadow of doubt, the greatest problem seems to be the esoteric (exclusive) and short-sighted nature of the organisation at least in the form that prevails today. Today's organisational dominant design is based on the centralised mindset, and this renders the adoption of this model not likely to unveil its full potential. Even worse, when centralisation is bundled with a bureaucratic delegation of authority and responsibility in a status-driven organisation, then the model is impossible to be implemented and flourish.

7.2 Commentary on the objectives of this research

In overall, this paper's objectives have been fulfilled.

Firstly, through the analysis of the Linux Project and its management, we have come to understand this novel organisational model and identified the crucial success factors enabling virtual decentralised collaboration under no central planning.

We have elaborated a managerial framework that can be theoretically applied to any type of virtual decentralised development and also discussed the strengths and limitations that

might stem from the adoption of the organisational model proposed by the Linux Project in other industries.

However, this research is case – specific and may turn out to be an exception rather than the rule. But in the absence of other appropriate examples and previous tested research, the findings presented in this paper can be invaluable in the form of hypotheses for the conduct of additional research.

The author hopes that this study will trigger further research to test the above assumptions.

This paper will come to an end by citing a visionary statement:

“In the automobile industry, each company does its own R&D. Every innovation is patented before it ever reaches the public, which may take five years for the improvements to be incorporated in an actual car after they were originally developed. If the automobile industry started taking on an open source development model with sharing across companies and countries, the cost and prices would eventually drop, innovation and development would speed up and exceptional features would be shared across many makers and models. The auto industry could finally come up with the safe, clean energy car.

The problem is that the car companies do not seem likely to support something that they perceive could put them out of business, even though this would not happen since nothing stops them from developing on their own and incorporating developments from their “open design shop” into their own products”.

Dan Barber “The Open Source Development Model”, accessible at http://mojomlin.com/articles/open_source_model.php?session=vTVi4tc1GfTb

Ultimately, it is a question of developing a new way of looking at competition and customers.

APPENDIX I: MICROSOFT: THE CATHEDRAL¹²

Microsoft is a 'product development' company and the key principles for managing product development are:

-Hire smart people and have small teams: Microsoft employs only 2-3% of all programmers that apply for a job; it is important to have bright people with excellent technical skills, including all managers. In this way, managers both create the products and make technical decisions. However, MS admits that management skills seem to be non-existent among managers as the late shipments witness, despite the corporate focus on shipping products (Cusumano & Shelby, 1995, p.58-71).

Having small teams does not only introduce flexibility but it is necessary in order to cope with the complexity of projects. "Brooks's Law" indicates that "the complexity and communication costs of a (software) project rise with the square of the number of developers, while work done only rises linearly" (Brooks, 1974), hence software should be developed by a closely knit team.

-Product architectures that reduce interdependencies among teams: The software architecture at MS is modular, which means that software is developed like a "Lego" toy. Such Lego-like pieces of code enable the fast assembly of software, cut down on the overall product complexity and make coordination easier. In contrast, buggy (defective) MS products have proven themselves to be not highly modular (Cusumano & Shelby, 1995: p40) and "MS Windows (in its various versions) enforces relatively little separation between the different system components. This has the unfortunate result that if any system component is changed, all programs need to be changed to conform" (Browne, 1998).

-Nearly all product development done at one site (Redmont, Seattle): Having everything located at the same place aims at speeding up the information flow (communication) and facilitating face-to-face communication. Ironically, the famous trial (part of the trial was based on internal MS e-mails) proved that even matters of strategic importance were communicated via e-mail. E-mail is the norm at MS (Drummond 2001).

- **An enormous feedback loop from customers:** MS gives phone calls to a few thousands of people to check whether everything (the product) is working properly and

¹² The term 'Cathedral' refers to centralisation and Raymond (1998) identifies Microsoft as 'The Cathedral'

its 'support services' provide an efficient way to get customer feedback (MS receives daily thousands of calls) (Cusumano & Shelby, 1995 p402). On the other hand, lots of people get irritated and confused when struggling to use MS' support services (Stephenson, 1999, p109-117).

-Organise around business units and constantly move people around: "Business Units aim at having the programmers more focused on specific customers and competitors, and moving people from one project to another (internal reorganisations) maximizes learning or/and allows burned-out employees a chance to get recharged in new areas, which stokes competitive fires and keeps the giant company nimble". However, "the real reason might be that MS reorgs are the result of massive political tectonics and power grabbing between the Windows and Windows NT groups" (Drummond, 2001 p95) and between the Systems and Applications Division (Cusumano & Shelby 1995, p52).

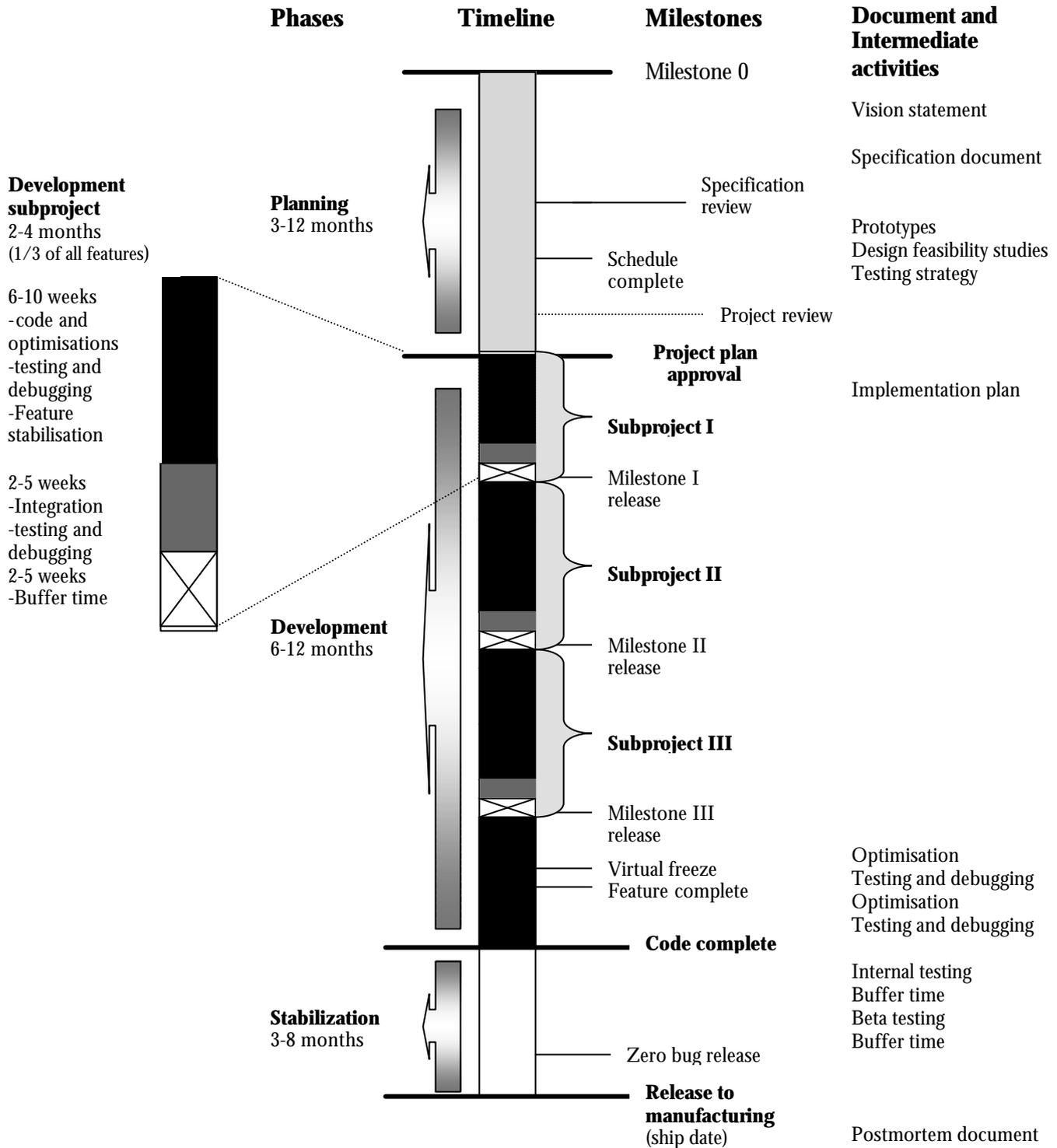
-A development process that allows large teams to work like small teams: For this purpose, MS has developed the "**Synch and Stabilize**" process (Figure I & Table I). The Synch-and-stabilize process was born in 1990 to deal with quality and project management problems that MS had encountered and resulted in a series of late and buggy (defective) product shipments. The main ideas behind this process are: a) breaking up a project into subprojects (milestones), b) doing daily builds of products, c) documenting everything to avoid doing the same mistake twice (Cusumano & Shelby 1995, p190-320).

Synch-and-stabilize: The Development model

MS has adopted the Synch-and-stabilize development model instead of the conventional Sequential development model (Table II). Sequential development treats development and testing as separate phases that are done one after the other, whereas at Synch-and-Stabilize, development and testing are done in parallel.

This approach has several advantages: a) learning is enhanced as work is done in parallel, b) greater flexibility arises when dealing with 'problems' (like defects and architectural issues), c) better quality and control are ensured because of "daily builds" and "milestones stabilizations", d) large teams are enabled to work like small teams. MS also claims that this model adds to the overall corporate responsiveness/adaptation because this development approach is an 'evolving system', and that it strengthens the focus on shipping products (Cusumano & Shelby 1995, p406-433).

Figure I: Synch-and-Stabilize Life Cycle for Program Management, Development and Testing (12-24 Months from Milestone 0 to Manufacturing)



Source: Adapted from M. Cusumano & R. Selby, Microsoft Secrets, (1997) - interview with Dave Moore, Director of development, 3/17/93; and Microsoft Corporation, Office Business unit, "Scheduling Methodology and Milestones Definition, unpublished internal document, 9/1/89

Table I: Overview of Synch-and-stabilize Development Approach

<p>Planning phase: define product vision, specification, and schedule</p>
<p>-Vision Statement: Product and program management use customer input to identify and prioritise product features -Specification document based on vision statement, program management and development group define feature functionality, architecture, component interdependencies -Schedule and Feature team formation: based on specs document, program management coordinates schedule and arranges feature teams that each contain approximately 1 program manager, 3-8 developers, and 3-8 testers (who work in parallel 1:1 with developers)</p>
<p>Development phase: Feature development in 3 or 4 sequential subprojects that each result in a milestone release</p>
<p>Program managers coordinate evolution of specification. Developers design, code and debug. Testers pair up with developers for continuous testing.</p> <p>-Subproject I First 1/3 of features: Most critical features and shared components -Subproject II Second 1/3 of features -Subproject III Last 1/3 of features: Least critical features</p>
<p>Stabilization Phase: Comprehensive internal and external testing, final product stabilization, and ship</p>
<p>Program managers coordinate “the economic web” and monitor customer feedback. Developers perform final debugging and code stabilisation. Testers recreate and isolate errors.</p> <p>-Internal testing : Thorough testing of complete product within the company -External testing: Thorough testing of complete product outside the company by “beta” sites such as OEMs, ISVs, and end-users. -Release preparation: prepare final release of “golden master” disks and documentation for manufacturing</p>

Source: Same as Figure I

Table II: Synch-and-Stabilize Vs. Sequential Development

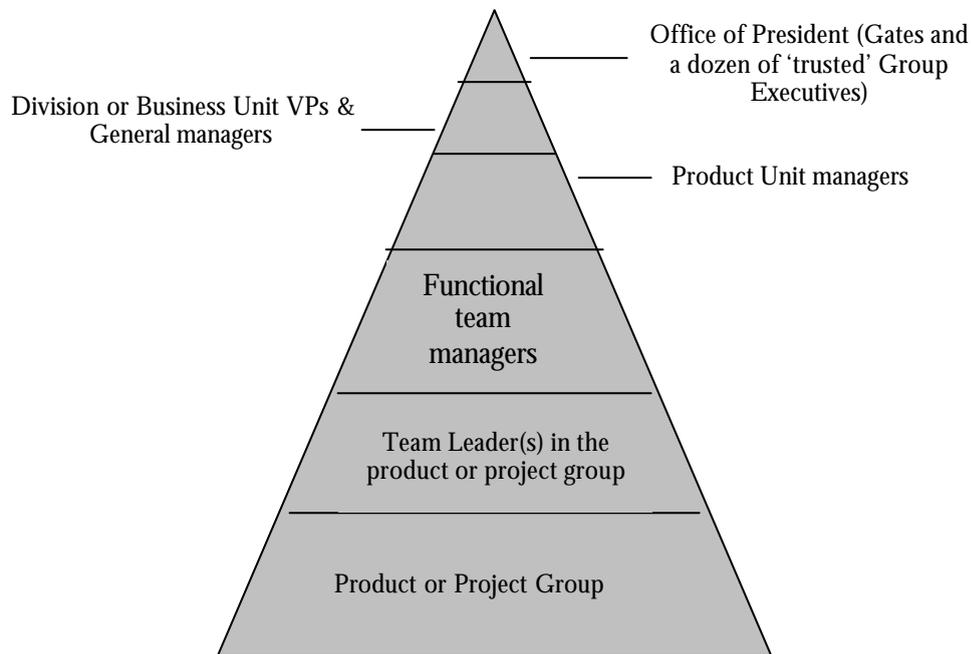
<u>Synch-and-Stabilize</u>	<u>Sequential Development</u>
Product development and testing done in parallel	Separate phases done in sequence
Vision statement and evolving specification	Complete “frozen” specification and detailed design before building the product
Features prioritised and built in 3 or 4 milestone subprojects	Trying to build all pieces of a product simultaneously
Frequent synchronizations (daily builds) and intermediate stabilisations (milestones)	One late and large integration and system phase at the project’s end
“Fixed” release and ship dates and multiple release cycles	Aiming for feature and product “perfection” in each project cycle
Product and process design so that large teams work like small teams	Working primarily as a large group of individuals in separate functional departments

Source: M. Cusumano & R. Selby, Microsoft Secrets, (1997), p407

Structure, Culture and Management

The corporate chain of control resembles the following pyramid (Figure II). Generally, MS’s structure is fluid and changes in structure are not rare but employees are discouraged to bypass (upwards) the hierarchical layers (because their superiors will take it as a serious offence) and especially going directly to Gates (Drummond, 2001 p39). Gates is having the final word for every decision made maybe because so far he is the only one who combines a business vision-insight and technical understanding (Cusumano & Shelby 1995, p70).

Figure II: Microsoft's Scalar Chain of Control



Based on M.A. Cusumano & R.W. Selby, Microsoft Secrets, (1995) p.41-52

MS's culture is anti-bureaucratic and developers are been given large amounts of freedom (but only over evolving features and experimenting with designs and not over strategic decisions). Control is almost non-existent and it is mostly enforced by the Synch and stabilize process (Cusumano & Shelby 1995, p405). "Adhering to rules and regulations, respecting formal titles, or cultivating skills in political infighting are not regarded as so important by the administration and employees (as incredibly smart and arrogant individuals) also do not want to be told what to do but are willing to be allowed to discover what to do" (Cusumano & Shelby 1995, p69-71).

"MS is a company where titles often don't mean as much as credibility, and thus, being blunt is a way to establish dominance. The company is rife with pecking-order gamesmanship, such as not answering e-mail or chronically arriving late to meetings" (Drummond,2001 p68) and in all, politics reign (at software development) in MS. "Opportunistic predators and competitors often "kill or kidnap sick or newborn"

technologies. Survival of the fittest is systemic ...internecine backstabbing did not evaporate in the presence of great intelligence and wealth, it became more brutal” (Drummond, 2001 p212). Insiders argue that Gates himself is responsible for this culture of conflict in two ways: by being arrogant (“Gates is famous for ridiculing someone’s idea just to see how he or she defends a position”) and by employing the brightest people and inducing them to grow arrogant and assertive (Drummond 2001).

Learning

Fresh employees do not go through a formal training programme but they learn on the job. Usually, they are being appointed to work together with an already experienced developer so as tacit knowledge that resides in the latter’s head is being effectively transferred to ‘fresher’ (Cusumano & Shelby 1995).

MS takes advantage of the knowledge it has accumulated by exploiting emerging mass markets and establishing its products as standards. But at an organisational level, learning is restricted. “Communication frequently suffers as a result of the inner corporate politics and even privileged employees have trouble getting information from inside Microsoft, a reflection of the long-standing schism between the company’s marketing staff and its legion of programmers” (Drummond, 2001 p35). MS even blocks widespread sharing (of their own source code) within the company (Valloppillil 1998; Nadeau).

Learning from customers is also limited since there is not effective two-way communication between developers and customers. Lots of people who have used MS’s ‘help/support services’ found it problematic and of limited help.

Innovation

MS is an innovative company in many respects. Gates has played a key role in establishing software as a “copyrighted” good ([Open Letter to Lobbyists, 1976](#)). MS has dominated the market by licensing the rights (of its operating system) which has proven to be an innovative move. But as far as product innovation is concerned, MS is a laggard.

Analysts claim that MS finds it difficult to balance being technology-driven with being consumer-driven and this results to great difficulty to move from incremental innovation to truly radical innovation or invention (Cusumano & Shelby 1995, p424).

After all, MS's competitive strategy is to design products for mass markets and then improve them incrementally by enhancing existing features or adding new ones (Cusumano & Shelby 1995, p403). Perhaps it this 'incremental evolution' product approach that impedes radical innovation: "The company has a very dramatic focus on its profitable business. I'm not saying that's bad. But it does preclude you from doing any dramatic thinking, doing any dramatic innovation" (Drummond, 2001 p69)....to the extent that several employees manipulate their inferiors to be given a chance to create something really novel (Drummond, 2001 p286- 287).

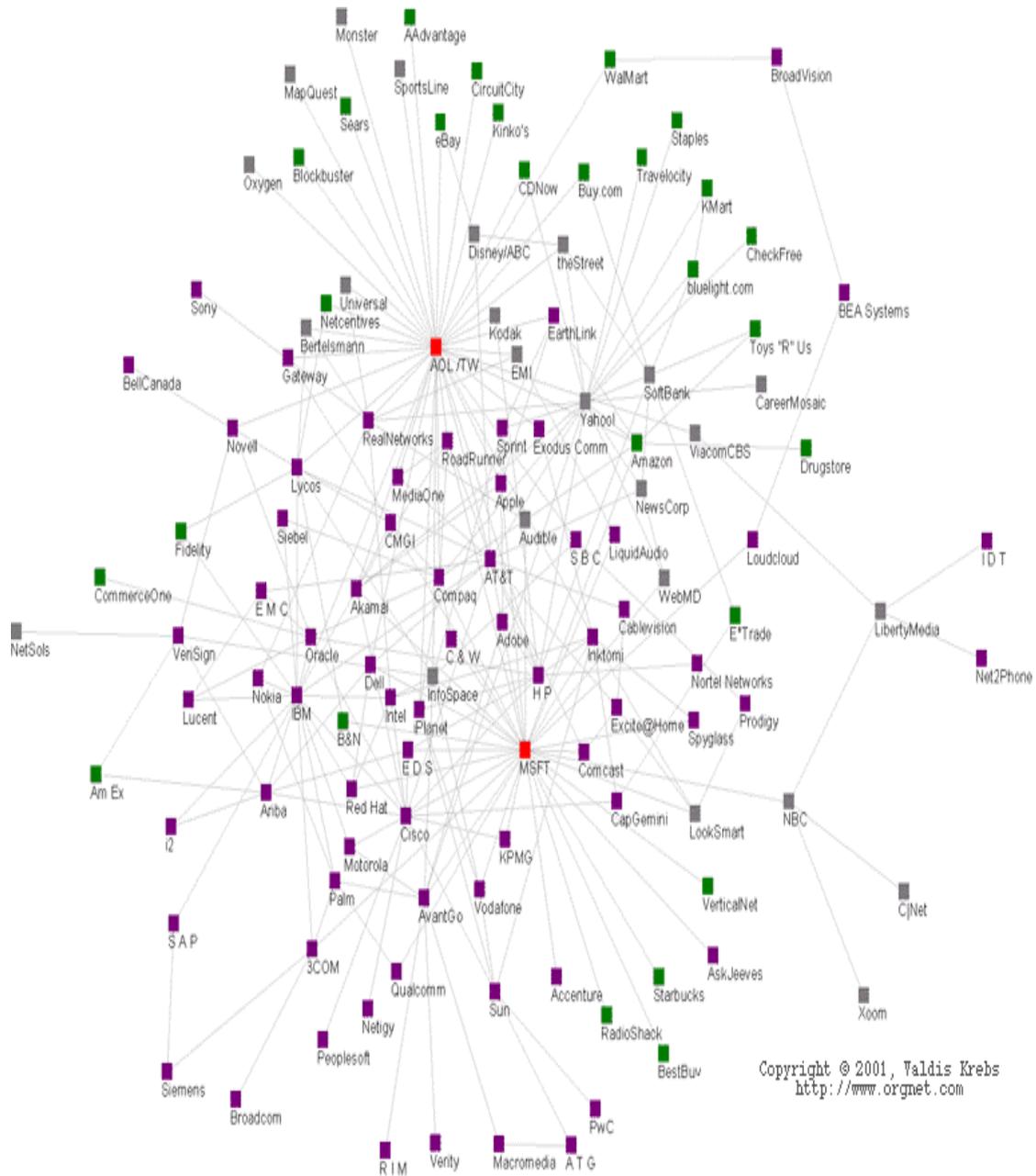
Management of the Economic Web

MS's management of its Economic Web is contradictory. At first, it is so successful that MS has achieved such a heavy lock-in that the Windows operating system family has been dominating the desktop market and until recently had been the unquestionable leader in the corporate and sever market that all 'software manufacturers' make sure that their products can be ported to the MS platform (run on Windows) without any additional effort from MS's side. Figure III shows that MS is the central hub in one of the largest and most powerful networks of organisational partnerships.

On the other hand, it was authoritative management of their economic web that led to the infamous trial as MS sought to conquer as many software-related markets (by providing complementary products) as possible and force the rest participants of the web to drop their own products and adopt MS's.

MS's aggressiveness – "a relentless pursuit of future markets" (Cusumano & Shelby 1995, p402) is harmful to the overall economic web and by trying to do so (by leveraging its technology to capture other software-related markets) has irritated many big industry players like Sun Microsystems, Netscape, HP and Oracle to name a few.

Figure III Microsoft holds the most powerful position in a gigantic network of corporate partnerships



Source: www.orgnet.com

APPENDIX II: INCREASING RETURNS

Conventional economic understanding of how markets and businesses operate is based on the assumption of diminishing returns: products or companies that get ahead in a market eventually run into limitations, so that a predictable equilibrium of prices and market share is reached. But history has proven that this is not always the case.

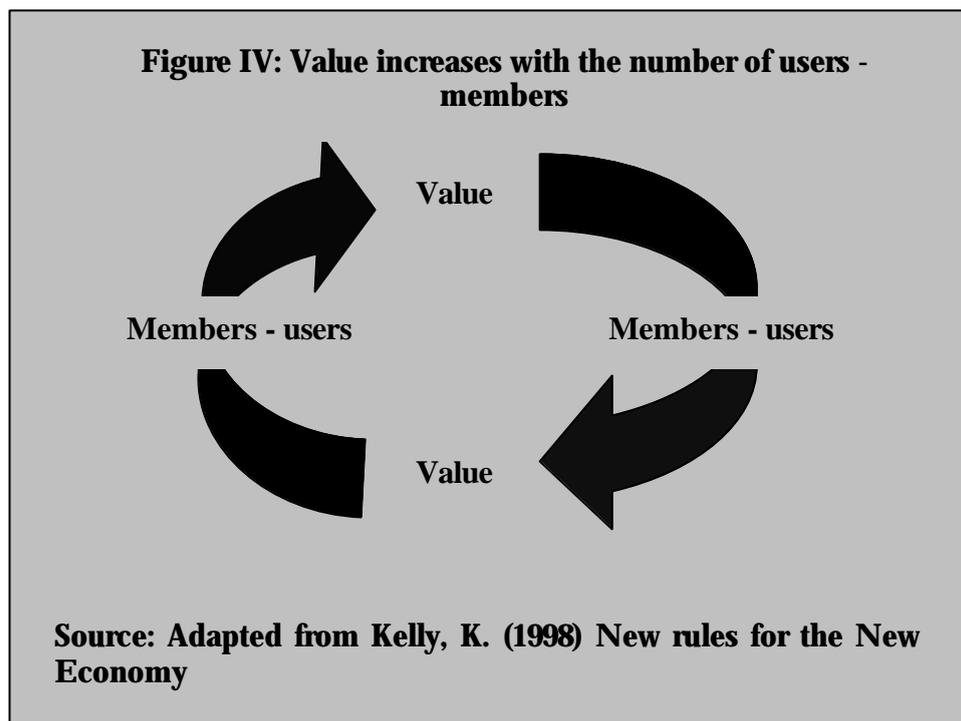
W. B. Arthur identifies 'increasing returns' to be the tendency for that which is ahead to get further ahead, for that which loses advantage to lose further advantage. In other words, "if a company or a technology – one of many competing in a market – gets ahead by chance or clever strategy, increasing returns can magnify this advantage, and the product or company or technology can go on to lock in the market" (Arthur, 1996, p1). Therefore, the value decreases rather than increases with scarcity.

For some time, economists have been accepting the notion of increasing returns but they restricted their scope only to the good produced with emphasis on the per unit cost of reproduction. Hence, since the cost of reproducing software is almost zero, software is not subject to diminishing returns. Whereas, W.B. Arthur proved that increasing returns (or positive feedback) dominate knowledge-intensive industries and form the inner mechanisms of these industries.

He cites the market for operating systems for PCs in the early 1980s as a characteristic example: DOS was born when Microsoft locked up a deal to supply an operating system for the IBM PC. Despite the fact that DOS was derided by computer experts and the fierce competition from other operating systems (mainly Macintosh and CP/M), the growing base of DOS/IBM users encouraged software developers to write for DOS. IBM PC/DOS's prevalence bred further prevalence, and eventually came to dominate a large portion of the market. This is where the 'economic web' comes in. DOS did not lock in the market because of technical superiority but because IBM passively allowed other companies to join its PC web as clones whereas Apple erred in the other direction by closing its Macintosh system to outsiders and hence opted not to create such a web (Arthur, 1996, p6).

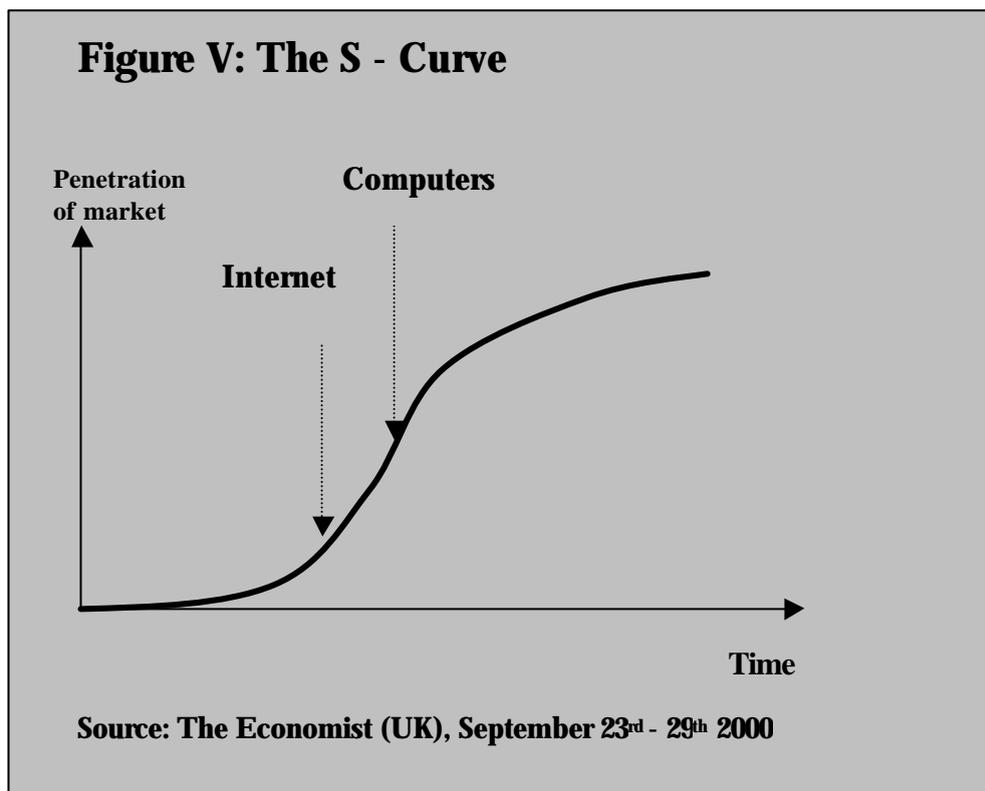
According to B. Arthur, there are three reasons why knowledge-intensive industries operate under increasing returns:

- a) high up-front (ie. R&D) costs,
- b) learning effects: high tech (or knowledge-intensive) products are difficult to use and require training. Therefore, once users have spent time to learn how to use a specific product, technology or platform, it becomes harder for them to switch to another alternative as they will have to devote additional effort and time again and,
- c) Network effects (or network externalities): according to “Metcalfe’s Law”, the experienced utility of belonging to a network increases exponentially with the number of users (Figure IV). If two people out of ten have a telephone, their experienced utility is not significant. But if all of them have a telephone, then it is much more useful. What I mean is that the more diffused the DOS/IBM PC became, the more value was attributed to it by users.



The crucial points are that a technology does not need to be the best to become the ‘dominant design’ and, all technologies follow an unshaped path (see figure V). They are

slow to get going, but once they reach critical mass the technology spreads fast. Usually, technologies typically improve as more people adopt them and gain experience with them. This link is a positive feedback loop: the more people adopt a particular technology, the more it improves, and the more incentive there is for further adoption (Arthur 1990).



APPENDIX III: COMMUNICATION NETWORKS

Regular patterns of information exchange among groups are called communication networks. Communication networks are either deliberately implemented when the group is organised (ie. many organisations adopt a hierarchical communication network that dictates the flow of information among the employees, especially from superiors to subordinates and horizontally to one's peers) or evolve in an evolutionary mode even when there is no formal attempt to specify and organise communication and its flow (without central planning).

In addition, the communication network tends to parallel role, status and attraction patterns. Studies indicate that usually those with higher status initiate and receive more information, as those who are better liked within the group (Aiken & Hage 1968; Bacharach & Aiken 1979; Jablin 1979; Shaw 1964).

The original analyses of communication networks were focused on small groups and conducted as laboratory experiments but recently, researchers have extended the scope of their research to investigate networks in many settings including large business organisations, families, research and development units, university departments and military units (Craddock 1985; Friedkin 1983; Keller & Holland 1983; Monge, Edwards & Kirste 1983; Tutzauer 1985). The findings of these studies attest to the powerful impact of networks on the overall group performance and efficiency and to the members' level of satisfaction. These are summarised below:

Centralisation and Performance

The earliest systematic studies of communication networks were carried out by the Group Networks Laboratory (MIT) in the 1950s. A characteristic study based on that systematic approach was done by Leavitt¹³(1951). This study showed that one of the most important features of a network is its degree of centralisation and additional research has verified it (Leavitt 1951; Bavelas & Barrett 1951; Shaw 1964, 1978).

¹³ For a complete discussion of the study and Leavitt's experiment, see Leavitt 1951 and Forsyth 1990

A centralised network is a structure at which one of the positions is located at the “crossroads” of communications and therefore controls the flow of information (by acting as an intermediary between nodes of the system) and a decentralised network is a structure at which the number of channels at each position – node is roughly equal, so none position is more central than another (all positions – nodes can ‘reach’ the same number of positions – nodes)¹⁴.

These early studies argued that centralised networks are more efficient than decentralised networks when the task is simple but when the task is more complex, the decentralised networks outperformed the centralised ones (Shaw 1964, p126).

Satisfaction

Since the number of peripheral positions in a centralised network exceeds the number of central positions, the overall level of satisfaction is lower than in a decentralised network (Shaw 1964; Eisenberg, Monge & Miller 1983; Krackhardt & Porter 1986).

Communication in Hierarchical Networks

For reasons of efficiency and control, many organisations adopt hierarchical communication networks. In such networks, information can either pass horizontally among peers or vertically, up and down from superiors to subordinates and vice versa (Jablin 1979). However, upward communications differ greatly from downward communications (Browning 1978; Katz & Kahn 1978) and upward communications are fewer in number, briefer and much more guarded than downward communications.

Evidence shows that in large organisations the upward flow of information may be impeded by the low-status members’ reluctance to send information that might reflect unfavourably on their skills and performance (Bradley 1978; Browning 1978).

¹⁴ For a more elaborate definition of centralised and decentralised communication networks, see Bavelas 1948, 1950; Freeman 1977; Grofman & Own 1982; Moxley & Moxley 1974)

In practice, this means that good news will travel quickly up the hierarchy, whereas the top management will be the last to learn bad news¹⁵. Further studies indicate that employees are likely to distort information when they are not satisfied with their job because they are not interested in helping the organisation to fulfil its objectives (O'Reilly 1978).

¹⁵ See Jablin 1979 for a detailed review of communication in hierarchical organisations

APPENDIX IV: INTERVIEWEES

Dan Barber is the main co-ordinator/maintainer of “Mojolin” which is a ‘Virtual Roof’ for the open source community and projects. He is probably the only person in the open source community that proposes the adoption of the open source development model in industries other than software

Chris Browne is the author of “Linux and Decentralized Development” which is highly regarded analysis of the strengths and weaknesses of decentralisation in the software industry

Chris Dibona volunteers as the Linux International webmaster and is also the Linux International grant development fund coordinator. He is proud to work as the Director of Linux Marketing for VA Research Linux systems (<http://www.varesearch.com>) and is the Vice President of the Silicon Valley Users Group (the world’s largest at <http://www.svlug.org>)

His writings have been featured in *The Vienna Times*, *Linux Journal*, *Tech Week*, *Boot Magazine* (now *Maximum PC*), and a number of online publications. Additionally, he has edited (together with Sam Ockman and Mark Stone) the book “Open Sources – Voices from the open source revolution” which is a key book on the Open Source movement. [This is the description given in Dibona 1999, no modifications have been made]

Matt Haak works for Intekk Communications which is a leading provider of Internet-based data storage solutions and web applications.

Ko Kuwabara has provided a fascinating sociological account of Linux and the open source community in his “Linux: A Bazaar at the Edge of Chaos”. It is one of the first (and certainly the most known) attempts to explain the ‘Linux phenomenon’ with Complexity Theory

Robert Laubacher is a leading researcher for Massachusetts Institute of Technology's Sloan School of Management's "Initiative on Inventing the Organizations of the 21st Century". The paper "The Dawn of the E-lance Economy" (Harvard Business Review, 1998, Sep-Oct) he has authored with Thomas Malone was one of the main influences for this study.

Glyn Moody is an eminent journalist-researcher and he has written extensively for Linux and open Source in *The Financial Times*, *The O'Reilly Network*, *the Guardian*, *The Economist*, *Wired*, *Computer*, *New Scientist* and he holds a column in *Computer Weekly*. His book "Rebel code: Linux and the Open Source Revolution" (2001) is regarded as the best history of the open source movement and an excellent example of research as he has interviewed all the key figures of the movement.

Ganesh Prasad is a leading open source figure and his "Examining some pseudo-economic arguments about open source" and "The Manager's Practical guide to Linux" are the most popular and economically sound efforts to bring corporate managers closer to open source software and particularly, the Linux OS.

Richard Stallman (RMS) 15 years ago, he started the GNU project, to protect and foster the development of free software. A stated goal of the project was to develop an entire operating system and complete sets of utilities under a free and open license so that no one would ever have to pay for software again.

In 1991, Stallman received the prestigious Grace Hopper Award from the Association for Computing Machinery for his development of the Emacs editor. In 1990 he was awarded a MacArthur Foundation fellowship. He was awarded an honorary doctorate from the Royal Institute of Technology in Sweden in 1996. In 1998 he shared with Linux Torvalds the Electronic Frontier Foundation's Pioneer award.

He is now more widely known for his evangelism of free software than the code he helped create.

Like anyone utterly devoted to a cause, Stallman has stirred controversy in the community he is a part of. His insistence that the term "Open Source software" is specifically designed to quash the freedom-related aspects of free software is only one of the many stances that he has taken of late that has caused some to label him an extremist. He takes it all in stride, as anyone can testify who has seen him don the garb of his alter ego, Saint GNUtias of the Church of Emacs.

Many have said, "If Richard did not exist, it would have been necessary to invent him." This praise is an honest acknowledgment of the fact that the Open Source movement could not have happened without the Free Software movement that Richard popularizes and evangelizes even today.

In addition to his political stance, Richard is known for a number of software projects. The two most prominent projects are the GNU C compiler (GCC) and the Emacs editor. GCC is by far the most ported, most popular compiler in the world. But far and wide, RMS is known for the Emacs editor. Calling Emacs editor an editor is like calling the Earth a nice hunk of dirt. Emacs is an editor, a web browser, a news reader, a mail reader, a personal information manager, a typesetting program, a programming editor, a hex editor, a word processor, and a number of video games. Many programmers use a kitchen sink as an icon for their copy of Emacs. There are many programmers who enter Emacs and don't leave to do anything else on the computer. Emacs, you'll find, isn't just a program, but a religion, and RMS is its saint.[this is the exact description given in Dibona 1999, no modifications have been made]

'Debian'* distributors:

Michael McConnel of Eridani Star System (UK)

Philip Hands of Philip Hands (UK)

Kajetan Hinner of Hinner EDV (Germany)

Ikarios (France)

Bitbone (Germany), Cetus Technology (Australia), LSL Australia (Australia), Jurgen Dumont of Linux Shop (Belgium), Libranet (Canada), Emanuel McGuyver of Vedova Computing (Italy), Philip Charles of Philip Charles (New Zealand), Max Tux (UK), Cheap*Bytes (US), Linux Central (US), StarLinux (Argentina), Bill Bennet of Computer Helper Guy (Canada)

Version 1.0 September 2001

Copyright © 2001 George Dafermos. You may forward these notes electronically to anyone for non-commercial reasons. Please send me any comments that might help to improve future versions.

Georgedafermos@zybermail.com

* 'Debian' is the most technical-oriented, stable, 'hardcore open source' version of the GNU/Linux OS and in fact the 'Open Source Definition' is based on the 'Debian Definition'.

BIBLIOGRAPHY

- Abbate, J. (1999) *Inventing the internet*, MIT Press
- Adler, P.A. & Adler, P. (1998) *Observational Techniques in Denzin, N.K. & Lincoln, Y.S. (Eds) Collecting and Interpreting Qualitative Materials*, Sage
- Aiken, M. & Hage, J. (1968) Organizational interdependence and intraorganizational structure, *American Sociological Review*, 33, 912-930
- Agre, P. (1994) *Networking on the Network*, accessible at <http://eserver.org/internet/Networking-on-Network.txt>
- Ansoff, I. (1965) *Corporate Strategy*, Penguin
- Arthur, J.B. (1990) Positive Feedbacks in the Economy, *Scientific American*, vol 262, February
- Arthur, J.B. (1996) Increasing Returns and the New World of Business, *Harvard Business Review*, July-August
- Axelrod, R. & Cohen, M.D. (1999) *Harnessing Complexity*, The Free Press
- Babcock, C. (2001) Open source code: A corporate building block, *Interactive Week*
- Bacharach, S.B. & Aiken, M. (1979) the impact of alienation, meaninglessness, and meritocracy on supervisor and subordinate satisfaction, *Social forces*, 57, 853-870
- Baldwin, C.Y., Clark, K.B., (1997) Managing in an Age of Modularity, *Harvard Business Review*, Sept-Oct
- Barber, D. (2001) The open source Development Model - is it applicable to other industries?, accessible at http://mojolin.com/articles/open_source_model.php?session=vTVi4tc1GfTb
- Barbrook, R. (1998) The Hi-Tech Gift Economy, *First Monday*, accessible at www.firstmonday.org/issues/issue3_12/barbrook/index.html
- Barnard, C.I. (1938) *The Functions of the Executive*, Cambridge, MA: Harvard University press
- Bavelas, A. & Barrett, D. (1951) An experimental approach to organization communication, *Personnel*, 27, 367-371
- Becker, H.S. (1956) Interviewing medical students, *American Journal of sociology*, 62, 199-201
- Belson, D. (1994) *The Network Nation Revisited*, accessible at www.stevens-tech.edu/~dbelson/thesis/thesis.html
- Bennett, H.S. (1980) *On becoming a rock Musician*, Amherst: University of Massachusetts Press
- Berners-Lee, T. (2000) *Weaving the Web*, texere Publishing
- Bezroukov, N. (1999) A second look at the Cathedral and the Bazaar, *First Monday*, accessible at www.firstmonday.org/issues/issue4_12/bezroukov/index.html
- Bovet, D. (1999) Value Webs: The next business revolution, *Logistics!*, Spring/Summer 1999, Mercer Management Consulting, accessible at www.mercer.com
- Bradley, P.H. (1978) Power, status, and upward communication in small decision-making groups, *Communication Monographs*, 45, 33-43
- Bramel, D. & Friend, R. (1981) Hawthorne, the myth of the docile worker, and class bias in psychology, *American Psychologist*, 36, 867-878
- Bricklin, D. (1999-2001) *The cornucopia of the Commons: How to get volunteer labor*, accessible at www.bricklin.com/cornucopia.htm
- Brooks, F. (1974) *The Mythical Man-Month*, Addison Wesley
- Browne, C.B. (1998) *Linux and Decentralized Development*, accessible at www.firstmonday.org/issues/issue3_3/browne/index.html
- Browning, L. (1978) A grounded organizational communication theory derived from qualitative data, *Communication Monographs*, 45, 93-109
- Bryman, A. (1988) *Quantity and Quality in Social Research*, London: Unwin Hyman
- Bulkeley, W. (2001) Linux gains corporate respectability, *ZDNet*, accessible at www.zdnet.com/filters/printerfriendly/0.6061.270.5804-2.00.html
- Burgess, R.G. (1984) *In the field: An Introduction to Field Research*, London: Allen & Unwin
- Burns, T. & Stalker, G.M. (1961) *The Management of Innovation*, Tavistock
- Burns, T. (1963) Industry in a New Age, *New Society*, 31 January
- *Business 2.0* (2001) "Change at the core: How Apple polished its rusty supply chain", accessible at www.business2.com/articles/mag/print/0.1643.14257.FF.html
- Chandler, A. (1966) *Strategy and Structure*, Doubleday

- Chandler, A (1977) *The Invisible Hand: The Managerial Revolution in American Business*, Harvard University Press
- Chandler, A., Hagstrom, P. & Solvell, O. (1998) *The Dynamic Firm: The role of technology, strategy, organization, and regions*, Oxford University Press
- Chassell, R.J. (2000) *Shaping Collaborative ICT Development and Initiatives for Global Prosperity*, accessible at www.gnu.org/philosophy/ICT-for-prosperity.html
- Cicourel, A. (1974) *Theory and Method in a Study of Argentine Fertility*, NY: Wiley
- Clarke, T & Clegg, S. (1998) *Changing Paradigms*, HarperCollinsBusiness
- Coleman, J.S. (1969) *Relational analysis: The study of organizations with survey methods in Etzioni A. (2nd ed) A sociological reader on Complex Organisations*, NY: Holt, Rinehart and Winston
- *Computer Weekly* (2001) Interview with R.M. Stallman, April 19, p36
- Constant, D., Kiesler, S. & Sproull, L. (1996) The kindness of strangers: on the usefulness of weak ties for technical advice, *Organization Science*, vol 17, no2, p119-135
- Cook, S.W. (1981) Ethical implications, in Kidder, L.H. (ed) *research Methods in social relations* (4th ed), NY: Holt, Rinehart & Winston
- Corbin, M. (1971) 'Problems and procedures in interviewing' in Pahl, J.M., *Managers and their Wives*, London: Allen Lane
- Cox, A. (1998), *Cathedrals, Bazaars and the Town Council*, accessible at www.slashdot.org/features/98/10/13/1423253.shtml
- Craddock, A.E. (1985) Centralised authority as a factor in small group and family problem solving, *Small Group Behavior*, 16, 59-73
- Crozier, M. (1964) *the Bureaucratic Phenomenon*, Chicago: University of Chicago Press
- Cusumano, M (1985) *The Japanese automobile industry: technology and management at Nissan and Toyota*, Harvard University Press
- Cusumano, M.A. & Selby, R.W. (1995) *Microsoft Secrets*, HarperCollins
- Dalton, M. (1959) *Men Who Manage*, NY: Wiley
- David, E.E. Jr & Fano, R.M. (1965) "some thoughts about the social implications of the accessible computing", excerpts reprinted in *IEEE Annals of the History of Computing*, vol14, no2, p36-39
- Davidow & Malone (1992) *The Virtual Corporation*, HarperCollins
- Davies, S. & Meyer, C. (1998) *Blur: the speed of change in the connected economy*, Capstone, Oxford
- Dawson, P & Palmer, G. (1995) *Quality Management: the theory and practice of implementing change*, Melbourne: Longman
- Dempsey, B.J., Weiss, D., Jones, P. & Greenberg, J. (1999) A Quantitative Profile of a Community of Open Source Linux developers, accessible at www.ibiblio.org/osrt/develpro.html
- Dibona (ed) (2000) *Open Sources*, O'Reilly
- Ditton, J. (1977) *Part-time Crime: an ethnography of fiddling and pilferage*, London: McMillan
- Douglas, J.D. (1976) *Investigative Social Research*, Newbury park, CA:Sage
- Drucker, P (1993) *Post-Capitalist Society*, Butterworth-Heinemann
- Drummond, M (2001) *Renegades of the Empire*, Three Rivers Press
- Duncan, A. & Hull, S. (2001) *Oracle & Open Source*, O'Reilly Publications
- *The Economist UK* (2001), Survey: Software, April 14th – 20th
- *The Economist UK* (2000) November 18th – 24th
- *The Economist (UK)* (2000) September 23rd - 29th
- Eisenberg, E.M. Monge, P.R. & Miller, K.I. (1983) Involvement in communication networks as a predictor or organizational commitment, *Human Communication Research*, 10, 179-201
- Etzioni A. (2nd ed) *A sociological reader on Complex Organisations*, NY: Holt, Rinehart and Winston
- Evans, P. & Wurster, T. (1999) *Blown to Bits*, Harvard Business School Press
- Fayol, H (1949) *General and Industrial Management*, Pitman (French original published in 1916)
- Festinger, L., Riecken, H.W., & Schachter, S. (1956) *When Prophecy Fails*, Mineapolis: University of Mineapolis Press
- Finch, J. (1984) "It's great to have someone to talk to": the ethics and politics of interviewing women, in Bell, C. & Roberts, H.(Eds) *Social Researching: Policies, Problems, Practice*, London: Routledge and Kegan Paul
- *First Monday* (1998), Interview with Linus Torvalds: what motivates free software developers?, accessible at www.firstmonday.org/issues/issue3_3/torvalds/index.html
- Fontana, A (1977) *The last frontier: the social meaning of growing old*, CA: Sage

- Fontana, A. & Frey, J.H. (1998) Interviewing – The Art of Science in Denzin, N.K. & Lincoln, Y.S. (Eds) *Collecting and Interpreting Qualitative Materials*, Sage
- Forsyth, D.R. (1990) *Group Dynamics*, 2nd ed., Brooks/Cole Publishing
- Franke, R.H. (1979) The Hawthorne experiments: Re-view, *American Sociological review*, 44, 861-867
- Franke, R.H. & Kaul, J.D. (1978) The Hawthorne Experiments: First statistical interpretation, *American sociological Review* 43, 623-643
- Frey, J.H. (1993) Risk Perception associated with a high level nuclear waste repository, *Sociological Spectrum*
- Friedkin, N.E. (1983) Horizons of observability and limits of informal control in organizations, *Social Forces*, 62, 54-77
- Gherardi, S (1997) Organisational Learning, in Sorge, A. & Warner, M. (Eds) *International Encyclopedia of Business and Management: The Handbook of Organizational Behavior*, London: International Thompson Business Pree
- Ghosh, R. & Prakash, V.V. (2000) The Orbiten Free Software Survey, *First Monday*, accessible at www.firstmonday.org/issues/issue5_7/ghosh/index.html
- Hacki, R. & Lighton, J. (2001) the Future of the Networked Company, *McKinsey Quarterly* no3, available at www.mckinseyquarterly.com
- Hagel, J.III (1996) Spider versus Spider, *McKinsey Quarterly* no1, available at www.mckinseyquarterly.com
- Hall, R. (1993) *The Soul of the Enterrprise: Creating a dynamic vision for American manufacturing*, Harper Collins
- Hames, R.D. (1994) *The Management Myth: Explaining the essence of future organization*, Sydney: Business & Professional Publishing
- Hames, R.D. (1997) *Burying the 20th Century: new paths for new futures*, Sydney: Business & Professional Publishing
- Handy, C. (1989) *The Age of Unreason*, Harvard Business School Press
- Handy, C. (1995) The virtual Organisation, *Harvard Business Review*, May-June
- Hare, A.P. & Naveh, D. (1986) Conformity and Creativity: Camp David, *Small Group Behavior*, 17, 243-268
- Hargreaves, S. (1999) Playing the open Source Game, accessible at www.tabula.demon.co.uk/games.html/
- Hayes, R.H., Wheelwright, S.C.& Clark, K.B. (1988) *Dynamic Manufacturing: Creating the Learning Organisation*, Free Press
- Hedlund, G (1993) in eds. Ghoshal, S. & Westney, E., *Organisation Theory and the Multinational Corporation*, St Martin's Press, New York
- Hiltz, S.R. & Turoff, M. (1978) *The Network Nation: Human Communication via computer*, Addison Wesley
- Hiltz, S.R. & Turoff, M. (1993) *The Network Nation: Human Communication via computer*, MIT Press
- Hodgetts, R.M., Luthans, M.F., & Lee, S.M. (1994) New Paradigm Organizations: From total Quality to Learning to World-Class, *Organization Dynamics*, vol 22, issue 3, Winter, p5-19
- Holland, J (1975) *Adaptation in natural and artificial systems: an introductory analysis with implications to biology, control and artificial intelligence*, MIT Press
- Holmstrom, B. (1999) Managerial Incentive Problems: A dynamic perspective, *Review of Economic Studies*, 66, 169-182
- Jablin, F.M. (1979) Superior-subordinate communication: the state of the art, *Psychological Bulletin*, 86, 1201-1222
- Janis, I.L. (1963) Group identifications under conditions of external danger, *British Journal of Medical Psychology*, 36, 227-238
- Janis, I.L. (1972) *Victims of groupthink*, Boston:Houghton-Mifflin
- Janis, I.L. (1983) Groupthink, in Blumberg H.H., Hare A.P., Kent V., Davis M.F. (eds), *Small groups and social interaction*, vol2, p39-46, NY:Wiley
- Janis, I.L. (1985) International Crisis Management in the Nuclear Age, *Applied social Psychology Annual*, 6, 63-86
- Janis, I.L. (1989) *Crucial Decisions: Leadership in policymaking and crisis management*, NY: Free Press
- Janis, I.L. & Mann, L. (1977) *Decision making: a psychological analysis of conflict, choice and commitment*, NY: Free Press
- Johnson, J. (1976) *Doing Field Research*, NY: Free Press
- Katz, D. & Kahn, R.L. (1978) *The Social Psychology of Organizations* (2nd ed), NY: Wiley

- Kauffman, S. (1993) *The origins of order: self-organization and selection in evolution*, NY: Oxford University Press
- Keller, R.T. & Holland, W.E. (1983) Communications and innovators in research and development organizations, *Academy of Management Journal*, 26, 742-749
- Kelly, K. (1998) *New rules for the New Economy*, Fourth Estate
- Khalak, A. (1999) Evolutionary model for open source software: Economic impact, accessible at <http://web.mit.edu/asif/www/ace/html>
- Kollock, P. (1996) The Economies of Online Cooperation: Gifts and Public goods in Cyberspace, accessible at <http://research.microsoft.com/vwg/papers/kollockeconomies.htm>
- Krackhardt, D. & Porter, L.W. (1986) The snowball effect: Turnover embedded in communication networks, *Journal of Applied Psychology*, 71, 50-55
- Krubasilk & Lautenschlager (1993) Forming successful Strategic alliances
- Kuwabara, K. (2000) Linux: A Bazaar at the Edge of Chaos, accessible at www.firstmonday.org/issues/issue5_3/kuwabara/index.html
- Landsberger, H.A. (1958) *Hawthorne revisited*, Ithaca NY: Cornell University
- Laubacher, R.J., Malone, T.W. & MIT Scenario Working Group (1997) Two Scenarios for 21st Century Organizations: Shifting Networks of Small Firms or All-Encompassing “Virtual Countries”?, MIT Initiative on Inventing the Organizations of the 21st Century, accessible at <http://ccs.mit.edu/21c/21CWP001.html>
- Laubacher, R.J., Malone, T.W. (1997) Flexible Work Arrangements and 21st Century Worker’s Guilds,
- MIT Initiative on Inventing the Organizations of the 21st Century, accessible at <http://ccs.mit.edu/21c/21CWP004.html>
- Lawrence, P.R. & Lorsch, J.W. (1967) *Organisation and Environment*, Harvard University Press
- Lazarsfeld, P.F. & Menzel, H. (1969) On the relation between individual and collective properties in
- Leadbeater, C (2000) *Living on thin air*, Penguin
- Leavitt, H.J. (1951) Some effects of certain communication patterns on group performance, *Journal of Abnormal and Social Psychology*, 46, 38-50
- Lee, G.K. & Cole, R.E. (2000) The Linux development as a model of open source knowledge creation, Haas School of Business
- Lerner, J. & Tirole, J. (2000) the Simple Economics of Open Source, accessible at www.people.hbs.edu/jlerner/simple.pdf
- Levy, S (1984) *Hackers*, Penguin
- Lighthouse ‘99 C.S. (1999) *Open source software*, Alliance for Converging Technologies
- Limmerick & Cunningham (1993) *Managing the New Organisation*, Chatswood, New South Wales: Business and Professional Publishing
- Locke, (2000) *Cluetrain Manifesto*, FT.COM
- *LWN* (2000) Interview with E. Raymond, accessible at <http://lwn.net/2000/features/ESR/>
- *O’Reilly Network: Code + Law: an interview with Lawrence Lessig* (2001), accessible at www.openp2p.com/lpt/a//p2p/2001/01/30/lessig.html
- Malinowski, B. (1922) *Argonauts of the Western Pacific*, London: Routledge & Kegan Paul
- Malone & Laubacher, (1998) The Dawn of the E-lance Economy, *Harvard Business Review*, Sep-Oct
- March, J.G. (1991) Exploration and Exploitation in Organizational Learning, *Organization Science*, vol 2, no1, p71-87
- March, J.G. & Simon, H.A. (1958) *Organizations*, Wiley
- Martin, J (1995) *Cybercorp*, Amacom
- Mayo, E. (1945) *the Social Problems of an Industrial Civilization*, Cambridge, MA: Harvard University Press
- McGregor, D. (1960) *the Human side of enterprise*, NY: McGraw-Hill
- McKelvey, M. (2001) *Internet Entrepreneurship: Linux and the Dynamics of Open Source Software*, Centre for Research on Innovation and competition, accessible at <http://les1.man.ac.uk/usercgi/cric/cricpaperdl.asp?>
- Moglen, E. (1999) Anarchism Triumphant: Free Software and the Death of Copyright, *First Monday*, accessible at http://old.law.columbia.edu/my_pubs/anarchism.html
- Monge, P.R. Edwards, J.A. & Kirste, K.K. (1983) determinants of communication network involvement: Connectedness and integration, *Group and Organization studies*, 8, 83-111
- Moody, G (1997) The Greatest OS That (N)ever Was, *Wired*, accessible at www.wired.com/wired/5.08/linux_pr.html

- Moody, G. (2001) Rebel Code: Linux and the Open Source Revolution
- Moon, J.Y. & Sproull, L. (2001) Essence of Distributed Work: the case of the Linux kernel, *First Monday*, accessible at www.firstmonday.org/issues/issue5_11/moon/index.html
- Morgan, G. (1994) Imaginization
- Nadeau, T. Learning from Linux: OS/2 and the Halloween Memos – Part 1/Halloween 1, accessible at www.os2hq.com/archives/linmemo1.htm
- Nadeau, T. Learning from Linux: OS/2 and the Halloween Memos – Part 2/Halloween 2, accessible at www.os2hq.com/archives/linmemo2.htm
- Nadeau, T. Learning from Linux: OS/2 and the Halloween Memos – Part 3/Halloween 3, accessible at www.os2hq.com/archives/linmemo3.htm
- Naisbitt, J. (1982) Megatrends, Ny: Warner Books
- Nalebuff, B.J. & Brandenburger, A.M. (1997) Co-opetition, HarperCollins
- Naughton, J. (2000) A brief history of the future: from radio days to Internet years in a lifetime, Overlook Press
- Negroponte, N. (1995) Being Digital, NY: Knopf
- Nohria, N. & Berkely, J.D. (1994) The Virtual Organisation: Bureaucracy, Technology, and the Implosion of Control, in Heckscher, C. & Donneillon (Eds) The Post-bureaucratic Organization: New Perspectives on Organizational Change, Sage, p108-28
- Oakley, A. (1981) 'interviewing women: a contradiction in terms', in Roberts, H. (ed) Doing Feminist Research, London: Routledge and Kegan Paul
- Oram (ed) (2001) Peer-to-Peer: Harnessing the power of disruptive technologies, O'Reilly
- O'Reilly, C.A. (1978) The intentional distortion of information in organizational communication: a laboratory and field investigation, *Human Relations*, 31, 173-193
- O'Reilly (2001) Remaking the P2P Meme Map, in Oram (Ed) Peer-to-Peer: Harnessing the power of disruptive technologies, O'Reilly
- Peters, T. & Austn, N. (1985) A Passion for Excellence, HarperCollins
- Prasad, G. (2001) Open Source-onomics: Examining some pseudo-economic arguments about open source, accessible at www.freeos.com/articles/4087
- Quinn, J.B., Anderson, P. & Finkelstein, S. (1996) Managing Professional Intellect: getting the most out of the best, *Harvard business Review*, vol 74, March -April
- Quinn, J.B., Baruch, J.J. & Zien, K.A. (1996) Software-based Innovation, *Sloan Management Review*
- Quinn, J.B. & Hilmer, F.G. (1994) Strategic Outsourcing, *Sloan Management Review*, Summer 1994
- Quinn, J.B. & Paquette, P.G. (1990a) Technology in Services: Creating Organisational Revolutions, *Sloan Management Review*, Winter 1990
- Quinn, J.B. & Paquette, P.G. (1990b) Technology in Services: Rethinking Strategic Focus, *Sloan Management Review*, Winter 1990
- Radloff, R. & Helmreich, R. (1968) Groups under stress: Psychological research in SEALAB II, New york: Irvington
- Rasmussen, P. (1989) Massage Parlor Prostitution, NY: Irvington
- Raymond, E. (1998a) A Brief History of Hackerdom, accessible at <http://earthspace.net/~esr/faqs/hacker-hist.html>
- Raymond, E. (1998b) The Cathedral and the Bazaar, *First Monday*, accessible at www.firstmonday.org/issues/issue3_3/raymond/index.html
- Raymond, E. (1998c) Homesteading the Noosphere, *First Monday*, accessible at www.firstmonday.org/issues/issue3_10/raymond/index.html
- Raymond, E.S. (1999a) A response to Nikolai Bezroukov, *First Monday*, accessible at www.firstmonday.org/issues/issue4_11/raymond/index.html
- Raymond, E. (1999b) The Magic Cauldron, accessible at www.tuxedo.org/~esr/writings/magic-cauldron/magic-cauldron.txt
- Reid, E.M. (1991) Electropolis: Communication and Community on Internet Relay Chat, University of Melbourne, Master's Thesis
- Reynolds, P.D. (1979) Ethical Dilemmas and Social Science Research, Jossey-Bass
- Rheingold, H. (1993) The Virtual Community: Homesteading on the Electronic frontier, NY: Addison-Wesley
- Ridderstrale, J. & Nordstrom, K. (2000) Funky Business: talent makes capital dance, FT.COM
- Roberts, J. (2000) From Know-How to Show-How? Questioning the Role of Information and Communication Technologies in Knowledge Transfer, *Technology Analysis & Strategic Management*, vol 12, no. 4
- Roethlisberger, F.J. & Dickson, W.J. (1939) Management and the Worker, Cambridge, MA: Harvard university Press

- Rothwell, R. (1992) Successful Industrial Innovation: critical factors for the 1990s, Edward Elgar
- Scanorama, September, 1995
- Roy, D. (1960) 'Banana Time: Job satisfaction and informal interaction', *Human Organization*, vol 18, no4, p158-168
- Schneier, B. (2000) Secrets and Lies, Wiley
- Schwartz, M.S. & Schwartz, C.G. (1955) Problems in Participant Observation, *American journal of sociology*, 60, 343-354
- Schwartz, H. & Jacobs, J. (1979) Qualitative Sociology: A method to the madness, NY: Free Press
- Scott, R.W. (1969) Field Methods in the study of organizations, in March, J.G. (ed) Handbook of Organisations, Chicago: Rand McNally & Company, p272-282 and reprinted in Etzioni A. (2nd ed) A sociological reader on Complex Organisations, NY: Holt, Rinehart and Winston, p558-576
- Senge, P (1992) The fifth discipline, Random House
- Shaw, M.E. (1964) Communication Networks in Berkowitz (ed) Advances in Experimental Social Psychology, NY: Academic Press
- Shaw, M.E. (1978) Communication networks fourteen years later in Berkowitz, L. (ed) Group Processes, Academic Press
- Shirky, C. "In Praise of Evolvable Systems", accessible at www.shirky.com/writings/evolve.html
- Shirky, C. "View source..Lessons from the Web's massively parallel development", accessible at www.shirky.com/writings/view_source.html
- Shirky, C. "The Interest Horizons and the Limits of Software Love", accessible at www.shirky.com/writings/interest.html
- Shirky, C. (2000) Darwin, Linux, and Radiation, *Business 2.0*
- Shy, (2001) The Economics of Network Industries, Cambridge University Press
- Sieber, S.D. (1973) 'The integration of fieldwork and survey methods', *American sociological Review*, vol 78, no 6, p1335-39
- Stallman, R.M. (1999) The GNU Operating System and the Free Software Movement in Dibona (Ed) Open Sources, O'Reilly
- Stallman, R. Why software should not have owners, accessible at www.gnu.org/philosophy/why-free.html
- Stephenson, N. (1999) At the beginning...was the command line, Avon
- Stewart, T.A. (1997) Intellectual Capital: the new wealth of organisations, Doubleday/Currency, N.Y.
- Spradley, J.P. (1979) The Ethnographic Interview, NY: Holt, Rinehart & Winston
- Stones, C.R. (1982) A community of Jesus people in South Africa, *Small Group Behavior*, 13, 264-272
- Stutz, M. (2000) Applying Copyleft to Non-Software information, accessible at www.gnu.org/philosophy/nonssoftware-copyleft.html
- Tapscott, D. & Caston, A. (1993) Paradigm Shift: Promise and Peril in the age of networked intelligence, McGraw Hill
- Tapscott, D. (1996) The Digital Economy, McGraw Hill
- Taylor, F.W. (1911) Principles of Scientific Management, Harper & Row
- The Last Dinosaur and the Tarpits of Doom: How Linux smashed Windows (1999), accessible at <http://mug.org/~cynbe/rants/lastdino.htm>
- Thompson, H. (1985) Hell's angels, NY: Ballantine
- Torvalds, L. (1999a) The Linux edge, *Communications of the ACM*, vol42, no4, pp38-39
- Torvalds, L. (1999b) The Linux edge, in Dibona, C. (Ed) Open Sources, O'Reilly, p101-119
- Tuomi, I. (2001) Internet, Innovation, and Open Source: Actors in the network, *First Monday*, p12-13, accessible at www.firstmonday.org/issues/issue6_1/tuomi/index.html
- Tutzauer, F. (1985) Toward a theory of disintegration in communication networks, *Social Networks*, 7, 263-285
- Valloppilli, V. (1998) open source software: A (new?) development methodology, also referred as the Halloween document, unpublished working paper, Microsoft Corporation, accessible at www.opensource.org
- Wakeford, J. (1981) 'From methods to practice: a critical note on the teaching of research practice to undergraduates', *Sociology*, vol 15, no4, p505-512
- Wax, R. (1960) Twelve years later: An analysis of field experiences, in Adams, R.N. & Preiss, J.J. (Eds) Human Organization Research, Homewood IL: Dorsey
- Weber, M (1947) The Theory of Social and Economic Organisation, Free Press, translated and edited by Henderson, A.M. & Parsons, T. (German original published in 1924)

- Weber, M. (1968) Economy and Society: An outline of interpretative sociology, NY: Bedminster Press
- Weber, S. (2000) The Political Economy of Open Source Software, BRIE Working Paper 140, accessible at <http://opensource.mit.edu/online-papers.php>
- Webb, S. & Webb, B. (1932) Methods of social Study, London; longman, Green
- Wheeler, D.A. (2001) Why open Source software/Free Software? Look at the Numbers!, accessible at www.dwheeler.com/oss_fs_why.html
- White, M.J. (1977) Counternormative behavior as influenced by deindividuating conditions and reference group salience, *Journal of Social Psychology*, 103, 73-90
- Whyte, W.F. (1943) Street Corner Society, Chicago: University of Chicago Press
- Young, R. (1999) Giving it Away, in Dibona C. (Ed) Open Sources, O'Reilly
- Zelditch, M. Jr (1969) Can you really study an army in the laboratory? in Etzioni A. (2nd ed) A sociological reader on Complex Organisations, NY: Holt, Rinehart and Winston
- Zweig, F. (1948) Labour, Life and Poverty, London: Gollancz

Internet

www.caldera.com
www.collab.net
www.cw360.com
www.cybergeography.org
www.debian.org
www.davenet.userland.com
www.forrester.com
<http://freshmeat.net>
www.gnu.org
www.idc.com
www.li.org
www.linux.com
www.linux.org
www.linuxjournal.com
<http://linuxtoday.com>
www.linux-mandrake.com
www.lwn.net
www.mckinseyquarterly.com
www.mojolin.com
www.newsforge.com
www.newscientist.com
www.openmagazine.net
www.opensource.org
www.opensource.mit.edu
www.oreilly.com & www.oreillynet.com
www.orgnet.com
www.ovum.com
www.redhat.com
www.researchoninnovation.org
www.slashdot.org
www.sourceforge.net
www.suse.com
www.tuxedo.org
www.valinux.com
www.wired.com

Usenet Newsgroups

News:comp.os.linux.advocacy
News:comp.os.linux.alpha
News:comp.os.linux.announce
News:comp.os.linux.apps
News:comp.os.linux.development.apps
News:comp.os.linux.development.system
News:linux.debian.devel
News:uk.comp.os.linux