To appear in: Practical Handbook of Internet Computing, M.P. Singh, ed. 2004 CRC Press.

Information Systems (in the Internet Age)

Eric Yu
University of Toronto

Abstract
Internet computing is changing the nature and scope of information systems (IS). Most IS methods and techniques were invented before the advent of the Internet. What will the world of information systems practice be like in the age of the Internet? What methods and techniques will be relevant? We review the world of information systems in terms of processes and products, qualities, social structures, and the role of automation. Given the rapid adoption of Internet thinking not only among technical professionals, but in the public consciousness, we outline the prospects and challenges for information systems in the emerging landscape. In particular, we highlight the need for richer modelling abstractions to support the diversity of services and modes of operation in the new age of world-wide open network information systems.

1 Introduction

How will Internet computing change the world of information systems?

Since the widespread commercial availability of computing technologies, information systems have been the dominant application area of computing. Organizations large and small, private and public, have come to rely on information systems for their day-to-day operation, planning, and decision making. Effective use of information technologies has become a critical success factor in modern society. Yet, success is not easily achieved. Many of the failures occur not in the technology, but in how technology is used in the context of the application domain and setting [Lyytinen, 1987; Standish, 1995]. Over the years, many methods and techniques have been developed to overcome the challenges in building effective information systems.

For many segments of society, the Internet has already changed how people work, communicate, or even socialize. Many of the changes can be attributed to information systems that now operate widely over the Internet. Internet computing is changing the scope and nature of information systems and of information systems work.

What opportunities, problems and challenges does Internet computing present to the information systems practitioner? What makes the new environment different? Which existing techniques continue to be applicable and what adaptations are necessary? What new methods and techniques are needed for information systems in the new reality of the Internet world?

Information systems (IS) is a multi-faceted field, and requires multi-disciplinary perspectives. In this chapter, we will only be able to explore some of the issues from a particular perspective – primarily that of information systems engineering, with an emphasis on the interplay between the technical world of system developers and programmers on the one hand, and the application or problem-domain world of users, customers, and stakeholders on the other. This perspective highlights some of the key issues of information systems as the bridge between raw technology and the application domain.

The chapter is organized as follows. Section 2 considers the world of IS practice before the advent of the Internet. In section 3, we ask how the world of users and applications are seen through the eyes of the IS practitioner, pre-Internet. Section 4 focuses on the new environment for information systems, brought about by Internet computing. Section 5 considers the implications and challenges for IS practice and research. Since conceptual abstractions are at the heart of information systems engineering, we focus in Section 6 on the kinds of abstractions that will be needed in the Internet age. We close in Section 7 with a summary and conclusions.

2 The World of Information Systems

Let us first consider the world of information systems practice, focusing on methods and techniques that have been in use since before the Internet.

What kinds of tasks and processes do information systems professionals engage in? What products do the processes produce? What quality concerns drive their daily work and improvement initiatives? How is the division of work organized among professional specialties, and within and across project organizations and industry sectors? Which areas of work can be automated, and which are retained as human tasks?

2.1 Processes

The predominant overarching organizing concept in most information systems curricula is that of the system development lifecycle [Gorgone et al., 2002].

The overall process of creating and deploying an information system is broken down into a number of well-defined interdependent processes. These typically include planning, requirements elicitation, analysis, specification, design, implementation, operations and support, maintenance and evolution. Verification and validation, including testing, is another set of activities that needs to be carried out in parallel with the main production processes. Some of the lifecycle activities involve participation from users and stakeholders. For example, technical feasibility and business priorities and risks are reviewed at predefined checkpoints. When externally provided components or subsystems are involved, there are processes for procurement and integration. Processes are also needed to manage the information content –during system development, as in defining the schemas, and during operation, as in ensuring information quality [Vassiliadis et al., 2001].

Systematic process is therefore a central concept in the field, imported initially from practices in large scale engineering projects. The systematic approach is used to control budget, schedule, resources, and opportunities to change course, e.g., to reduce scope, or to realign priorities. Nevertheless, lack of systematic process continues to be a concern, as a contributing factor to poor quality or failure of software and information systems. Substantial efforts are used to institutionalize good practices in processes, through standards, assessment and certification, and process improvement initiatives (e.g., Capability Maturity Model Integrated (CMMI) [Chrissi et al., 2003], ISO 9000 [ISO, 1992]).

Many IS projects adopt methodologies offered by vendors or consulting companies which prescribe processes in detail, supported by associated tools. Prescriptive processes provide guidance and structure to the tasks of system development. They may differ in the stages and steps defined, the products produced at each step, and how the steps may overlap or iterate (e.g., the waterfall model [Royce, 1970], the spiral model [Boehm, 1988], the Rational Unified Process [Kruchten, 2000]). While prescriptive processes aim to create order out of chaos, they are sometimes felt to be over-restrictive, or require too much effort and time. Alternative approaches that have developed over the years include rapid prototyping, Joint Application Development (JAD) [Wood and Silver, 1995], Rapid Application Development (RAD) [McConnell, 1996], and more recently agile development [Cockburn, 2001]. All of these make use of a higher degree of human interaction between developers and users and stakeholders.

2.2 Products

Complementary to and intertwined with processes are the products that they produce. These include products and artefacts that are visible to the end-user, such as executable code, documentation, training material, as well as intermediate products that are internal to the system development organization. When more than one organization is involved in the creation and maintenance of a system, there are intermediate products that are shared or flow across them.

Most of the products are informational – plans, requirements, specifications, test plans, designs, budgets and schedules, work breakdowns and allocations, architectural diagrams and descriptions, and so on. Some products are meant for long-term reference and record keeping, while others are more ephemeral and for short-term coordination and communication.

These informational products are expressed or encoded in a variety of modelling schemes, languages, and notations. Information modelling techniques continues to be a central area of research [Brodie et al., 1984; Webster, 1988; Loucopoulos and Zicari, 1992; Boman et al., 1997; Mylopoulos, 1998]. Widely used techniques include Entity-Relationships (ER) modeling [Chen, 1976], Integrated Definition for Function Modeling (IDEF0) [NIST, 1993] (based on the Structured Analysis and Design Technique (SADT) [Ross and Shoman, 1977]), and the Unified Modeling Language (UML) [Rumbaugh et al., 1999].

Large system projects involve many kinds of processes producing a great many types of information products related to each other in complex ways. Meta-modeling and repository technologies (e.g., [Brinkkemper and Joosten, 1996; Jarke, 1998; Bernstein et al., 1999]) are often used to manage the large amounts and varieties of information produced in a project. They support retrieval, update, and coordination among project team members. Meta-models define the types of processes and products and their inter-relationships. Traceability from one project artefact or activity to another is one of the desired benefits of systematic project information management [Ramesh and Jarke, 2001].

2.3 Qualities

While processes and products constitute the most tangible aspects of IS work, less tangible issues of quality are nevertheless crucial for system success. Customers and users want systems that not only provide the desired functionalities, but also a whole host of non-functional requirements that are often conflicting – performance, costs, delivery schedules, reliability, safety, accuracy, usability, and so on. Meeting competing quality requirements has been and remains a formidable challenge for software and information systems professionals [Boehm and In, 1996]. Not only are system developers not able to guarantee correctness of large systems, they frequently fail to meet non-functional requirements as well. Many of the issues collectively identified as the software crisis years ago are still with us today [Gibbs, 1994].

Research sub-specialties have developed to come up with specific techniques to address each of the many identified areas of quality or non-functional requirements– performance, reliability, and so forth. However, many qualities are hard to characterize, e.g., evolvability, reusability. When multiple requirements need to be traded off against one another, systematic techniques are needed to deal with the synergistic and conflicting interactions among them. Goal-oriented approaches (e.g., [Chung et al., 2000]) have recently been introduced to support the systematic refinement, interaction analysis, and operationalization of non-functional requirements. On the project management level, institutionalized software process improvement programs (such as CMMI) target overall project quality improvements. Quality improvements need to be measured, with results feeding back into new initiatives [Basili and Caldiera, 1995].

2.4 Social Structures

Most information systems require teams of people to develop and maintain. The organization of projects into process steps and artefacts implies a social organization among the people performing the work, with significant degrees of task specialization. Some tasks require great familiarity with the application domain, while others require deep knowledge about specific technologies and platforms. Some require meticulous attention to detail, while others require oversight and vision.

A well functioning people organization is as important as technical capabilities for project success [Weinberg, 1998; DeMarco and Lister, 1999]. Every work product requires time and effort to produce. So whether they get produced, and to what quality, depends on motivation, reward structures, priorities, as well as on personnel capabilities. Yet the social organization is often implicit in how processes and products are organized, rather than explicitly designed, since there are few aids beyond generic project management tools.

Processes are judged to be too heavy (excessive regimentation) or too light (chaotic) based on the perceived need for human creativity, initiative, and flexibility for the task at hand. Factors influencing the determination of social structure include project and team size, familiarity of the application domain, maturity of the technologies, as well as socio-cultural and economic factors. Industry categories and structures (e.g., ERP vendors vs. ERP implementers) and human resource categories (database designers vs. database administrators) are larger social structures that specific project social structures must operate within.

The social nature of IS work implies that its structure is a result of conflicting as well as complementary goals and interests. Individual and groups come together to cooperate to achieve common objectives, but they also compete for resources, to pursue private goals, and can have different visions and values. Processes and products that appear to be objectively defined are in fact animated by actors with initiatives, aspirations, and skills.

The human intellectual capital perspective [Nonaka and Takeuchi, 1995] highlights the importance of human knowledge and ingenuity in systems development. While considerable knowledge is manifested in the structure of processes and products, a great deal of knowledge remains tacit in human practices and expertise. There are limits on how much and what kinds of knowledge can be made explicit, encoded in some language or models, and systematically managed.

In reflecting on the practices of information systems and software development as professional disciplines, authors acknowledge the human challenges of the field [Banville and Landry, 1992; Humphrey, 1995].

2.5 Automation

The quest for higher degrees of automation has been a constant theme in information systems and software engineering. The large amounts of complex information content and relationships, the need for meticulous detail and accuracy, the difficulty of managing large teams, and the desire for ever quicker delivery and higher productivity – all call for more and better automated tools.

Numerous tools to support various stages and aspects of IS work have been offered – from CASE tools that support modelling and analysis, to code generators, test tools, simulation tools, repositories, and so on. They have met with varying degrees of success in adoption and acceptance among practitioners.

Automation relies on the formalization of processes and products. Those areas that are more amenable to mathematical models and semantic characterization have been more successful in achieving automated tool support. Thus, despite great efforts and many advances, information systems work remains labour-intensive and requires social collaboration. Many issues are socio-technical, e.g., requirements elicitation, reuse, agile development, process improvement.

The difficulties encountered with automation in the developer's world may be contrasted with that in the user's world, where automation is the mandate and expectation of the IS practitioner.

3 The World According to Information Systems

Information systems convey and manipulate information about the world. The kind of world (the application setting, the problem domain) that is perceived by the IS analyst is filtered through presuppositions of what the technology of the day can support. In the preceding section, we reflected upon the world of the information systems practitioner in terms of processes and products, qualities, social structures, and automation. Let us now use the same categories to consider how IS practitioners treat the world that they serve – the world that users and stakeholders inhabit.

3.1 Processes and products

The predominant conceptualization of the world as seen by IS analysts is that of processes and products. The main benefit of computers was thought to be the ability to process and store large amounts of encoded information at high speeds and with great accuracy. In early applications, information systems were used to replace humans in routine, repetitive information processing tasks, e.g., census data processing, business transaction processing. The processes automate the steps that humans would otherwise perform. Processes produce information artefacts that are fed into other processes. The same conception can be applied to systems that deal with less routine work, e.g., management information systems, decision support systems, executive information systems, and strategic information systems.

Models and notations, usually graphical – with boxes and arrows – were devised to help describe and understand what processes are used to transform what kinds of inputs into what kinds of outputs, and state transitions. Data Flow Diagrams [DeMarco, 1979], SADT [Ross and Shoman, 1977], Entity-Relationships modeling [Chen, 1976], and UML [Rumbaugh et al., 1999] are in common use. These kinds of models shape and constrain how IS analysts perceive the world of the application domain [Curtis et al., 1992].

We note that processes and products in the developer's world are treated somewhat differently than those in the user's world. In the latter, attention is focused on those that are potentially automatable. In the former, there is an understanding that a large part of the processes and products will be worked on by humans, with limited degrees of automation. We will return to this point in Section 3.4.

3.2 Qualities

Most projects aim to achieve some improvement or change in qualitative aspects of the world – faster processing, fewer delays, information that is more accurate and up-to-date, lower costs, and so forth. In section 2.3, we considered the pursuit of quality during a system development project. Here we are concerned with the quality attributes of processes and products in the application domain in which the target system is to function. Many of the same considerations apply, except now the IS professional is helping to achieve quality objectives in the client's world.

Quality issues may be prominent when making the business case for a project, and may be documented in the project charter or mandate. However, the connection of these high level objectives to the eventual definition of the system in terms of processes and products may be tenuous. Quality attributes are not easily expressible in models that are used to define systems, since the latter is defined in terms of processes and products. Quality concerns may appear as annotations or comments in accompanying text (e.g., bottleneck, missing flow). Furthermore, a model typically describes only one situation at a time, e.g., the current system as it exists, or a proposed design. Comparisons and alternatives are hard to express, as are pros and cons and justifications of decisions. These kinds of information, if recorded at all, are recorded outside of the modelling notations. Some qualities can be quantified, but many cannot. Specialized models can be used for certain quality areas (e.g., economic models, logistical models), but analyzing cross-impacts and making tradeoffs among them is difficult, as noted in Section 2.3. Design reasoning is therefore hard to maintain and keep up to date when changes occur.

3.3 Social structures

Information systems change the social structures of the environment in which they operate. In performing some aspects of work that would otherwise be performed by people, they change how work is divided and coordinated. Bank tellers take on broader responsibilities as customer service representatives; phone inquiries are funnelled into centralized call centres; and data entry tasks had moved from clerical pools to end-users and even to customers. Each time a system is introduced or modified, responsibilities and relationships are reallocated, possibly contested and renegotiated. Reporting structures, and other channels of influence and control, are realigned. The nature of daily work and social interactions are altered. Reward structures and job evaluation criteria need to be readjusted.

The importance of social factors in information systems have long been recognized (e.g., [Kling, 1996; Lyytinen, 1987]). Many systems fail or fall into disuse not because of technical failure, but in how the technology is matched to the social environment. Alternative methodologies have been proposed that pay attention to the broader context of information systems, e.g., Soft Systems Methodology [Checkland, 1981], ethnographic studies of work practices [Goguen and Jirotka, 1994], Participatory Design [Muller and Kuhn, 1993], Contextual Design [Holtzblatt and Beyer, 1995], and so on. Each has developed a following, and have produced success stories. Workplace democracy approaches have a long history in Scandinavia [Ehn, 1988].

Nevertheless, despite the availability of these alternative methods, social issues are not taken into account in-depth in most projects. When an information system operates within an organizational context, the corporate agenda of the target system dominates – e.g., to improve productivity and profitability. Users who are employees are expected to fit their work practices to the new system. While users and other stakeholders may be given opportunities, to varying degrees, to participate and influence the direction of system development, their initiatives are typically limited.

Existing modelling techniques, most of which focus on process-and-product, are geared primarily to achieving the functionalities of the system, deferring or side-stepping quality or social concerns. For example, in the Structured Analysis paradigm, people and roles that appear in “physical” data flow diagrams (DFDs) are abstracted away in going to the “logical” DFD, which is then used as the main analysis and design vehicle [DeMarco, 1979]. Actors in UML Use Case Diagrams [Rumbaugh et al., 1999] are modelled in terms of their interactions with the system, but not with each other. Given the lack of representational constructs for describing social relationships and analyzing their implications, IS practitioners are hard pressed to take people issues into account when considering technical alternatives. Conversely, stakeholders and users cannot participate effectively in decision making when the significance and implications of complex design alternatives are not accessible to them. It is hard for technical developers and application domain personnel to explore, analyze, and understand the space of possibilities together.

3.4 Automation

The responsibility of the IS professional is to produce automated information systems that meet the needs of the client. While the success of the system depends a great deal on the environment, the mandate of the IS professional typically does not extend much beyond the automated system.

In the early 1990s, the concept of business process reengineering overturned the narrow focus of traditional IS projects. Information systems are now seen as enablers for transforming work processes, not just to automate them in their existing forms [Hammer, 1990; Davenport and Short, 1990]. The transformation may involve radical and fundamental change. Process steps and intermediate products judged to be unnecessary are eliminated, together with the associated human roles, in order to achieve dramatic efficiency improvements and cost reductions. IS therefore has been given a more prominent role in the redesign of organizations and work processes. Yet IS professionals do not have good techniques and tools for taking on this larger mandate. Many BPR efforts failed due to inadequate attention to social and human issues and concerns. A common problem was that tacit knowledge among experienced personnel is frequently responsible for sustaining work processes, even though they are not formally recognized. Existing IS modelling techniques, based primarily on a mechanistic view of work, are not helpful when one needs to take a socio-technical perspective to determine what processes can be automated or eliminated or reconfigured.

4 What's new with Internet Computing?

Why can't the practice of information systems carry on as before, as outlined in the preceding two sections? What parameters have changed as a result of Internet computing?

From a technology perspective, the Internet revolution can be viewed, simplistically, as one in connectivity, built upon a core set of protocols and languages – TCP/IP, HTTP, and HTML or XML. With their widespread adoption through open standards and successful business models (e.g., affordable connection fees, free browsers), the result, from the user's point of view, is a worldwide, borderless infrastructure for accessibility to information content and services – information of all types (as long as they are in digital format), regardless of what "system" or organization they originate from. Digital connectivity enabled all kinds of information services to co-exist on a common interoperable network infrastructure. The same users can access any service on the network. Service providers have ready access to a critical mass of users, through the network effect of Metcalfe's Law [Gilder, 1993]. Automated services can access, invoke, and interact with each other.

Universal connectivity at the technology level makes feasible universal accessibility at the information content and services level. Internet computing is therefore triggering and stimulating the removal of technology-induced barriers in the flow and sharing of information. Previously compartmentalized information services and user communities are now reaching out to the rest of the world. Information systems, with Internet computing, find themselves broadening in scope with regard to content types, system capabilities, and organizational boundaries:

(i) Information systems have traditionally focused on structured data. The Internet, which gained momentum by offering information for the general public, unleashed the enormous appetite for unstructured information, especially text and images, but also multi-media in general. Corporations and other organizations quickly realized that their information systems capabilities must address the full range of information content, to serve their publics as well as their internal workings. They can do this relatively easily, by embracing the same Internet technology, now rendered for internal use as intranets.

(ii) Users working with information do not want to have to deal with many separate systems each with their own technical idiosyncrasies. Internet computing, by offering higher-level platforms for application building, makes it possible for diverse technical capabilities to appear to the user as a single "system", as in the concept of portals. Thus, Internet computing vastly expands what a user may expect of a "system".

(iii) Most information systems in the past had an internal focus and operated within the boundaries of an organization, typically using proprietary technologies from a small number of chosen vendors. Internet computing is inverting that, both from a technological viewpoint, and from an information services viewpoint. Technologically, the momentum and economics of Internet computing is such that corporate internal computing infrastructures are converting to open Internet standards [IETF; W3C; OpenGroup]. At the information services level, organizations are realizing that much can be gained by opening up their information systems to the outside world – to customers and constituents, to suppliers, partners and collaborators, as in B2B e-commerce and virtual enterprises [Mowshowitz, 1997]. The boundaries of organization have become porous and increasingly fluid, defined by the shifting ownership and control of information and flows, rather than by physical locations or assets.

5 Information Systems Challenges in the Internet Age

With the apparently simple premise of universal connectivity and accessibility, Internet computing is changing the field of information systems fundamentally. It is redrawing the map of information systems. As barriers to connectivity are removed, products and processes are being redefined. Quality criteria are shifting. New social structures are emerging around systems both in the user's world and in the developer's world. People's conception of what computers can do, and what they can be trusted to do, are evolving.

5.1 Products and processes

Let us first consider the impact of Internet computing on processes and products in the information system user's world. Over the years, a large organization would have deployed dozens or hundreds of information systems to meet their various business and organizational needs. Each system automates its own area of work processes and products, with databases; forms, reports and screens for input and output. Soon it was realized that these independently developed systems should be interacting with each other directly and automatically.

Thus long before the Internet, numerous approaches have emerged for extending the reach of information processes and products beyond the confines of a single system. For example, information in separate databases often in fact represent different aspects of the same entity in the world. A customer, a purchase, an insurance policy, a hospital stay – each of these has many aspects that may end up in many databases in the respective organization. Database integration techniques were introduced to make use of data across multiple databases. Data warehousing provided powerful tools for understanding trends by enabling multi-dimensional analysis of data collected from the numerous operational databases in an organization. Data mining and knowledge discovery techniques enhanced these analyses.

Enterprise-wide information integration has also been motivated from the process perspective. Business process reengineering stimulated cross-functional linking of previously standalone "stove pipe" systems. Workflow management systems and document management systems were used to implement end-to-end business processes that cut across functional departmental lines.

Different approaches were used to achieve integration or interoperability at various levels. middleware technologies provided inter-process communication at a low level, requiring handcrafting of the interactions on an application-to-application basis. Enterprise application integration (EAI) products offered application level interoperability. Enterprise resource planning (ERP) systems offered integrated package solutions for many standard back-office business processes. Integration is achieved at the business process level by adopting process blueprints from a single vendor [Curran and Keller, 1997]. When systems had disparate conceptual models of the world, meta-modelling technique were used to map across them.

Internet computing technologies come as a boon to the mishmash of technologies and approaches that have proliferated in the IS world. By offering a common network computing and information infrastructure that is readily accessible to everyone – regardless of organizational and other boundaries – the integration and interoperability challenges that organizations had been confronting individually at an enterprise level is now being addressed collectively on a world-wide scale [Yang and Papazoglou, 2000]. Organizations that had already been opening up their operations to the external world through IS-enabled concepts such as supply chain management, customer relationships management (CRM) and virtual enterprise now have the momentum of the whole world behind them. Inter-organizational interoperability initiatives (also known as B2B e-commerce) no longer need to be done from scratch between partner and partner, but are done by entire industries and sectors through consortiums that set standards for business application level protocols, e.g., Rosettanet [Rosettanet], ebXML [ebXML], HL7 [HL7], UN/CEFACT [UN/CEFACT], OASIS [OASIS], and BPMI [BPMI].

Once the interaction protocols are set up, processes in one organization can invoke automated services in another without human intervention [WebServices]. In an open world, anyone (individuals or organizations, and their information systems) has potential access to the full range of products and services offered on the open network, in contrast to the closed proprietary nature of interactions pre-Internet. "End-to-end" process redesign can now be done not only from one end of an organization to another, but across multiple organizations, through the customer, and back.

To support flexible open interactions, products and services increasingly need to be accompanied by rich meta-data, e.g., using XML and its semantic extensions [Berners-Lee et al., 2001]. Catalogues, directories are needed for locating desired products and services. Brokers, translators, and other intermediaries are needed [Wiederhold and Genesereth, 1997].

Internet computing is stimulating coordinated use of multimedia and multi-channel user interactions. The same user – a sales representative, a student, or a community services counsellor – may be drawing on material that combine text, images, voice, music, and video on a desktop, laptop, PDA, mobile phone, or other devices. There will be increasing demands to enable higher level automated processing of digital information in all formats. The semantic web initiative, for example, aims to enhance semantic processing of web content through formal definitions of meanings (ontologies) for various subject domains and communities [Gomez-Peres and Corcho, 2002; SemanticWeb].

In terms of products and processes, the challenges brought about by Internet computing can be summarized as one of diversity. Standardization is one way to overcome excessive proliferation of diversity. Yet, in an open world, the capacity to cope with diversity must be there to recognized the inherent need to differentiate, and not to inhibit innovation. So the great challenge is to have processes that can interoperate seamlessly, and products that are intelligible and useful to its intended users.

Given these recent transformations in the user's world, the character of work in the developer's world has seen rapid changes in the past decade or so. Development work that used to be organized vertically – from requirements to design to implementation – are now dealing increasingly with horizontal interactions, coordination, and negotiations. Each layer in the developer's world – from business process analysis to architectural design to implementation platforms – must address interaction with peers, coping with diversity and interoperability at that level [Bussler, 2002]. As a result, each level is working with new kinds of information artefacts (e.g., using new languages and meta-models [Mylopoulos et al., 1990]) and new development processes (e.g., understanding and negotiating peer level protocols and interactions) [Isakowitz et al., 1998].

5.2 Qualities

In the Internet world, when we are pursuing quality goals such as faster processing, greater accuracy and reliability, better usability, and so forth, we are dealing with processes that cut across many systems and organizations, and information products from many sources. Unlike in the traditional world of closed systems, Internet computing implies that one may need to rely on many processes and products over which one has limited control or influence. Achieving quality in an open network environment requires new techniques not in common use in the traditional environment.

For example, if the product or service is commodity-like, one can switch to an alternate supplier when the supplier is unsatisfactory. This presupposes efficient market mechanisms, with low transaction costs. There needs to be accurate descriptions of functionalities as well as quality attributes, using metrics that allow meaningful comparison by automated search engines and shopbots. This may involve third-party assessors and certifiers of quality, and regulatory protection and legal recourse when obligations are not met. The situation is complicated by the dynamic nature of Internet collaborations, where automated processes can come together for one transaction fleetingly, then in the next moment go their own ways to participate in new associations. When market mechanisms fail, one would need to establish more stable associations among players based on past experiences of trust [Rosenbloom, 2000; Falcone et al., 2001].

As for developers, due to the open network environment, one can expect special emphasis on certain non-functional requirements such as scalability, reliability, usability, security (including availability, integrity, and confidentiality), time-to-market, costs, and performance. Design tradeoffs may be more challenging, as the designer attempts to cater to market-based dynamically changing clientele, as well as stakeholders in more stable longer-term relationships.

Design techniques have traditionally been weak in dealing with quality or non-functional requirements. With Internet computing, there is the added need to support the more complex decision making involving competing demands from multiple dynamically configured stakeholders.

5.3 Social structures

Traditional information systems that are function-specific and narrowly focused imply that there are well-defined human roles and responsibilities associated with each one, e.g., planning vs. execution; product lines vs. geographic regions. Internet computing, by facilitating ready access to a wide range of information system capabilities, enables much greater flexibility in social organizational arrangements. When a common platform is used, learning curves are reduced, and movement across roles and positions are eased. For example, as more routine tasks are automated, the same personnel can monitor a wider scope of activities, respond to problems and exceptions, and engage in process improvement and redesign. More fundamental changes are occurring at the boundaries of organizations. The Internet has made the online consumer/citizenry a reality. Many transactions (e.g., catalogue browsing and ordering, banking and investments, tax filing, proposal submissions) are now handled online, with the user directly interacting with automated information systems. The organization is effectively pushing some of its processes to the customer's side. Similar boundary renegotiations are taking place among suppliers and partners. These shifts in boundaries are changing internal organizational structures as well as broad industry structures. New business models are devised to take advantage of newly created opportunities [Timmers, 2000]. Disintermediation and re-intermediation are occurring in various sectors of society and business. Organizations are experimenting with different kinds of decentralization, re-centralization, market orientation, as well as internal coordination mechanisms. Citizen groups are organizing their activities differently using chat rooms and other web-based media.

Many of the social organizational relationships are being shifted into the automated realm, as software agents act on behalf of their human counterparts, as alluded to in the preceding sections. New partners may be found via automated directory services (e.g., [UDDI]).

Social dynamics is therefore becoming an important subject matter in the analysis and design of information systems in the Internet age. Unfortunately, there are few techniques in the information system practitioner's toolbox that takes social structures and dynamics into account [deMichelis, 1998].

The social organization of system development organizations are also rapidly changing, most directly resulting from changes in development processes and the types of artefacts they produce. New professional categories arise as specialized knowledge and skills are sought. New dependencies and relationships among teams and team members need to be identified and negotiated. Education and training, upgrading, obsolescence – these and other labour market and human resources issues are often critical for project success.

Larger changes analogous to those happening in the user's world are also happening in the developer's world. Industry structures are changing. Technology vendors are specializing or consolidating. Component creators and service providers are springing up to take advantage of the Internet computing platform. Outsourcing or in-sourcing, proprietary vs. open, commercial off-the-shelf (COTS) systems, open source development – all of these alter the dynamics of information systems work. The adoption of certain system architectures has direct significance for the social structures around it.

As in the user's world, some processes in the developer's world will be carried out by software agents, with the social dynamics carried over into the automated realm. Again, there is little theoretical or practical support for the information systems practitioner facing these issues.

5.4 Automation

With Internet computing, the broad range of information system capabilities are now accessible to the user on a single consistent platform. The ability of these functions and capabilities to interoperate creates a powerful synergistic effect, because they can make use of information that's already in digital form and machine processable. Automated functions can invoke each other at electronic speeds. For example, programmed trading of commodities and financial instruments have been operating for some time. It is feasible to have medical test results sent to and responded to by one's family physician, specialist, pharmacist, and insurer all within seconds rather than days or weeks, if all the "processing" is automated. Governments can potentially collect electronic dossiers on the activities and movements of citizens for tax collection and law enforcement. Almost all knowledge work in organizations will be conducted through computers, as the technological support for searching, indexing, cross-referencing, multi-media presentation, and so forth, become routine expectation. More and more documents and other information content are “born digital” and will remain digital for most of their lifecycle.

In the past, what gets automated is decided for each “system” within a well-defined context of use. Significant investments and efforts were required for each system because each application system required its own underlying computing support (vertical technology stack) and operational procedures (including data entry and output). Cost-benefits analysis leads to automation only in selected areas or processes, typically based on economic and efficiency criteria. This is typically done by system analysts at the early stages of system definition, with the application system as the focal point and unit of analysis.

The Internet has turned the tide in automation. As we are witnessing, the concept of an isolated application system is dissolving. Information content – public or private - may pass through numerous systems on the network, invoking processing services from many operators and developed by many system vendors (e.g., via web services). Because of the synergy and the network effect, it will be irresistible in economic or efficiency terms to automate [Smith and Fingar, 2002]. The investments have already been made; the technology infrastructure is there; and the content is already in machine processable form. It will take a conscious effort to decide what not to automate.

The decision on what to automate requires difficult analysis and decision making, but is crucial for the success and sustainability of systems. There will often be a clash of competing interests among stakeholders, involving issues of trust, privacy, security, reliability, vulnerability, risks and payoffs. Even economic and speed advantages that are the usual benefits are not necessarily realizable in the face of potential downsides. One needs to understand broad implications and longer term consequences – heavily interconnected networks imply many far-reaching effects that are not immediately discernable.

With the digital connectivity infrastructure in place, one has to face decisions on the degrees of automation. Informational processing can range from the minimal (e.g., message transmission and re-presentation at the destination, with no processing in between) to the sophisticated (extracting meaning and intent and acting upon those interpretations). But even in messaging services, traffic patterns can be monitored and analyzed. So the analysis of what a system or service should do and should not do is much more complicated than in the pre-Internet world. They will involve complex human and machine processes as well as conflicting interests from many parties and perspectives.

The same forces apply to the developer’s world. The increased demand from the great variety of information system capabilities will lead to pressure for more automation. When automation is raised to the level such that technical details can be hidden from the user, the entire development can be pushed into the user’s world.

Much human knowledge and experience cannot be made explicit and codified symbolically. Where and how tacit knowledge and human judgement interact and combine with automated machine processes remains a difficult design challenge.

Information system practitioners have few tools that can support the analysis of these issues and to help make these important decisions for users, service providers, and for society.

6 Conceptual Abstractions for Information Systems

The practice of information systems is based on conceptual abstractions with well-defined properties. Abstractions focus attention to aspects of the world that are relevant for information systems development.

As we saw in Section 3, in order to develop information systems to serve in some application setting, the richness of that setting needs to be reduced through a set of modelling abstractions. The characterization of the user's world need to be expressed in some models that can be analyzed, leading to decisions about what aspects of that world will become the responsibility of the intended system. During systems development (section 2) the models are translated, through a series of steps, each time through a different set of abstractions, from ones that describe the user's world (e.g., travel plans and bookings), to ones that describe the machine's world (data and operations in computers that store those plans and execute those bookings) [Jarke et al., 1992]. At each stage or level of translation, analyses are performed to understand the situation; decisions are made on how elements at the current level should translate into or correspond to elements at the next level.

Notations are important to help communication between the world of stakeholders and users on the one hand, and system designers on the other. They need to have sufficient expressiveness to convey the desired needs and requirements. Yet the notations need to be simple and concise enough for widespread adoption and standardization. Furthermore, they need to support analysis and inference, preferably automated, so as to be scalable.

A necessary consequence of using modelling is to restrict what can be said about the world. Aspects of the world that cannot be expressed tend to be left out, and will no longer be the focus of attention during system development. Therefore, the design of notations requires a difficult balance [Potts and Newstetter, 1997]. With too much detail, one can get bogged down; with too little, one can get the wrong system that does not do what is needed or intended. Whatever is chosen, the modeling techniques used shape the analyst's perception of the world.

As we surveyed the user's and developer's worlds, we noted that not all the relevant aspects are equally well supported by existing modelling abstractions. Processes and products are the mainstay of most existing modelling techniques, but qualities and social structures are not well captured. Therefore those issues are not systematically dealt with in mainstream methodologies.

In the Internet age, it will be especially important to have conceptualizations and abstractions that relate concerns about social relationships and human interests to the technical alternatives in systems design, and vice versa.

The preceding sections revealed that, with Internet computing, information systems are now expected to deal with a much wide range of conceptualizations than before. We will consider the abstractions for expressing what, when, where, how and why, and who.

6.1 Conceptualizing "what" and "when"

The "what" refers to things that exist, events that occur, and properties and relationships that hold. These aspects of the world are most heavily addressed in existing modeling schemes. An online transaction needs to identify products bought and when payments take effect. Patient records need to distinguish different kinds of diseases and symptoms and document the nature and timing of treatments.

Product and process proliferation triggered by Internet computing will test the limits of these modelling techniques. Current work in ontologies [Guarino and Welty, 2002] are revealing subtleties and limitations in earlier work. Knowledge structuring mechanisms such as classification, generalization, and aggregation [Greenspan et al., 1994] that has been in used in object-oriented modeling will be used extensively. The global reach of Internet computing is likely to push each classification and specialization scheme to the limits of its applicability, for example, to organize the types and features of financial instruments that are becoming available on the global investment marketplace, which may be transacted electronically. Meta-modeling techniques are especially relevant for working with the conceptual structures spanning multiple domains or contexts [Nisssen and Jarke, 1999], for example, classification of medical conditions by physicians as opposed to insurance companies, and in one country or culture versus another.

The "what" and "when" cover the static and dynamic aspects of the world. Time is not always explicitly represented in dynamic models, but may appear as sequence or precedence relationships, e.g., coordinating multi-step financial transactions that traverse many systems, countries, time zones, and organizations. Internet computing brings more complex temporal issues into play. Multiple systems cooperating on the network may operate on different time scales, interact synchronously or asynchronously at different periods. They will have different development and evolution lifecycles that require coordination. Conventional modeling techniques typically deal with only first-order dynamics, as in process execution and interaction. Second or higher order dynamics, such as change management, are usually not well integrated in the same modeling framework. When Internet computing is relied upon as a platform for long term continuity, there will be processes that have time horizons extending into years and decades (e.g., inter-organizational workflow, managing the impacts of legislative change). Over long time horizons, process execution and process change will have human and automated components, involving users and developers. Similarly the long term presentation and preservation of information content over generations of information systems will be significant issues (e.g., identification and referencing of objects, how to make objects interpretable by humans and machines in future generations) [GAO, 2002].

6.2 Conceptualizing "where"

It should be no surprise that the Internet challenges conventional conceptions of geographic space. On the one hand, it enables users to transcend physical space, reaching out to others wherever they are. On the other hand, world wide coverage means that users do come from many different geographic regions and locales, and that these differences are significant, or can be taken advantage of in many applications. Peoples' preferences and interests, linguistic and cultural characteristics, legal frameworks and social values, all of these can be correlated to physical locations. Mobile and ubiquitous computing, silent commerce, and intelligent buildings can make use of fine-grained location awareness. Modeling techniques developed in geographical information systems (GIS) can be expected to find wider applications stimulated by Internet computing, e.g., to offer location sensitive services to users in vehicles, to help visitors navigate unfamiliar territory, or to track material goods in transit. Physical locations will often need to be mapped to jurisdictional territories, e.g., in enforcing building security.

6.3 Conceptualizing "how" and "why"

The distinction between "what" and "how" is often made within software engineering and in systems design. Requirements are supposed to state the "what" without specifying the "how". Here the "what" refers to essential characteristics, whereas the "hows" used to achieve the "what" reflect incidental characteristics that may be peculiar to the implementation medium or mechanisms. This is one of the core principles of abstraction in dealing with large systems. The what and how distinction can be applied at multiple levels or layers, each time focusing on features and issues relevant to that level, while hiding "details" that can be deferred.

Structured analysis techniques (e.g., SADT, DFD) rely heavily on a layered, hierarchical structure for the gradual revealing of details. Although the vertical layering of processes or functions can be viewed as embodying the how (downwards) and why (upwards) for understanding the structure of a system, much of the reasoning leading to the structure is not captured. There is almost always more than one answer when considering how to accomplish something. Yet most modeling techniques only admit one possible refinement in elaborating on the "how". Alternatives, their pros and cons, and why one of them is chosen, typically cannot be described and analyzed within the notation and methodology. The lack or loss of these information in systems descriptions makes system evolution difficult and problematic.

Understanding how and why will be critical in the Internet environment, where systems can be much more dynamic and contingent. Systems are typically not conceived in terms of a single coherent system with a top level overview which can then be decomposed into constituent elements, to be designed and constructed by the same project team. Instead, systems could arise from network elements that come together in real time to participate in a cooperative venture, then dissolve and later participate in some other configuration. There will be many ways to assemble a system from a network of potential participants. Components can come together in real time for short periods to form a cooperative venture.

Designers, or the systems component themselves, must have ways of identifying possible solutions (the hows) and ways for judging which ones would work and work well according to some quality criteria and goals (the whys).

Reasoning about how and why is needed at all levels in systems development, e.g., at the application service level (a navigation system recommends an alternate route based on traffic conditions and user preferences), and at the systems and networks level (a failure triggering diagnostics that lead to system recovery).

Representing how and why has been addressed in the areas of goal-oriented requirements engineering [Mylopoulos et al., 1999; van Lamsweerde, 2000], design rationales [Lee, 1997], the quality movement (e.g., [Hauser and Clausing, 1988], and partly in requirements traceability [Ramesh and Jarke, 2001].

6.4 Conceptualizing "who"

The most underdeveloped aspect is the conceptualization of the notion of "who" to support systems analysis. The Internet environment will bring many actors into contact with each other. There will be individuals, groups, organizations, and units within organizations such as teams, task forces, and so forth. They will be acting in many different capacities and roles, with varying degrees of sustained identity. They will have capabilities, authorities, and responsibilities. Information system entities (e.g., software agents) may be acting on behalf of human actors, taking on some of their rights and obligations.

Traditional information systems tend to exist in closed environments, e.g., within the authority structure of a single organization. Social structures are more easily defined and instituted, e.g., as used in role-based access control techniques in computer security [SACMAT, 2003]. With Internet computing, there can be much greater numbers of participants and roles (both in types and instances), with dynamic and evolving configurations of relationships. Consider, for example, healthcare information systems that connect patients at home to hospitals and physicians, later expanding to community care centres, and eventually to insurers and government regulatory agencies and registries. New configurations can arise from time to time due to innovation (e.g., in business models) or regulatory change. There are complex issues of reliability, trust, privacy, and security, as well as operational responsibilities. There are difficult analysis and tradeoffs, arising from the complex social relationships. Virtuality of the Internet creates many new issues for notions of who, e.g., identity and personae, influence and control, authority and power, ownership and sharing [Mulligan, 2003]. All of these are crucial in analyzing new organizational forms (centralize vs. decentralize, internal vs. external) and social relationships. Notions of community are important in knowledge management, and in managing meaning in conceptual models.

These issues are not well addressed in traditional information systems techniques. Some of them are beginning to be studied in agent-oriented approaches to software systems and information systems [Papazoglou, 2001; Huhns and Singh, 1998]. However much of the work on agent-oriented software engineering is currently focused on the design of software agents [Giunchiglia et al., 2003]. For information systems, more attention needs to be paid to modelling and analyzing conceptions of "who" as applied to complex relationships involving human as well as software agents [Yu, 2002].

7 Summary and Conclusions

Internet computing is changing the world of information systems. Information systems started historically as computer applications designed specifically for a well defined usage setting. They implemented a narrow range of repetitive processes, producing pre-determined types of information products. Most often, these are automated versions of manual processes. Systems development was primarily "vertically" oriented – the main activities or processes were to convert or translate a vision of a new system into functioning procedures (executable code) and populated databases. A system project involves significant investments and lead time because it typically requires its own technology infrastructure, including networking.

With Internet computing, information systems projects will become more and more "horizontal". The larger proportion of the effort will be to coordinate interactions with other system and information resources that already exist or may exist in the future. They will potentially interact with a much wider range of users, with different quality expectations and offerings, and evolving usage patterns. Development work can be more incremental, as new systems are built from ever higher-level platforms and components. These developments will enable information systems professionals to concentrate on the application level, helping users and stakeholders to formulate and understand their problems and aspirations in ways that can take advantage of information technology solutions.

Given the broad spectrum of technological capabilities that are now available on a common infrastructure, with ever higher-level interoperability, information systems are coming into their own as embodying and realizing the wishes and visions of the user's world, instead of reflecting the limitations and inherent structures of the underlying technologies.

The chief limitations in this regard are those imposed by the modeling techniques of the day. As we have reviewed in this chapter, traditional IS techniques have focused on those aspects that lead most directly to the computerization or automation of existing information processes and products, as these are perceived to be the most tangible results of the project investment. The compartmentalized, vertical system development perspective means that the perception and conceptualization of the world is filtered through preconceived notions of what can be automated, based on the technological implementation capabilities of the day. Hence traditional techniques have focused on the modeling and analysis of processes and products, activities and entities, objects and behaviours. Ontologies for analysis and design are well developed for dealing with the static and dynamic dimensions of the world. Much less attention has been paid to the quality and social aspects, even though these are known to be important success factors, and to have contributed to many failures.

In the horizontal world brought about by massive networking, quality and social dimensions will come to the fore. Modeling techniques must cover the full range of expressiveness needed to reasoning about the *what, where, when, how and why*, and especially the *who* of information systems in their usage and development contexts. Refined characterizations of the notions of *who* will be crucial for tackling the human and social issues that will increasingly dominate systems analysis and design. Privacy and security, trust and risks, ownership and access, rights and obligations, these issues will be contested, possibly down to level of transactions and data elements, by a complex array of stakeholders in the open networked world of Internet computing.

Development processes and organizations are benefiting from the same advances experienced in user organizations. Systems development work is taking advantage of support tools that are in effect specialized information systems for its own work domain. As the level of representation and analysis is raised closer to and becomes more reflective of the user's world and their language and conceptual models, the developer's world blends in with the user's world, providing faster and tighter change cycles achieving more effective information systems.

Despite connectivity and potential accessibility, the networked world will not be of uniform characteristics or without barriers. There will continue to be differentiation and heterogeneity in technical capabilities as well as great diversity in information content and services. Internet computing allows information systems to transcend many unwanted technological barriers, yet it must allow user communities to create, maintain, and manage boundaries and identities that reflect the needs for locality and autonomy. Techniques that support the management of homogeneity within a locality and heterogeneity across localities will be a crucial challenge in the Internet age.

Acknowledgements. The author is grateful to the editor and Julio Leite for many useful comments and suggestions.

8 References

Claude Banville and Maurice Landry (1992) Can the Field of MIS be Disciplined? In: Galliers, R. (Hg.): Information Systems Research. Issues Methods and Practical Guidelines. London et al.: Blackwell, S. 61-88.

Victor R. Basili and G. Caldiera (1995) "Improve Software Quality by Reusing Knowledge and Experience," Sloan Management Review 37, 1. Fall, 55-64.

Tim Berners-Lee, Jim Hendler and Ora Lassila (2001). The Semantic Web. Scientific American, May, 2001.

Philip A. Bernstein (2001) Generic model management – a database infrastructure for schema management. Proc. CoopIS 01, Trento, Italy, LNCS 2172, Springer, 1-6.

Philip A. Bernstein, Bergstraesser, T., Carlson, J., Pal, S., Sanders, P., Shutt, D. (1999) Microsoft repository version 2 and the open information model. Information Systems 24, 2:71-98.

Barry Boehm (1981). Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall.

Barry Boehm (1988) A spiral model of software development and enhancement. IEEE Computer, May, 61-72.

Barry Boehm and Hoh In (1996) "Identifying Quality-Requirement Conflicts," IEEE Software, March, 25-35.

Christoph Bussler (2002) P2P in B2BI. Proceedings of the 35th Hawaii International Conference on System Sciences. IEEE Computer Society. Vol. 9: 302-311.

Sjaak Brinkkemper and S. Joosten (1996) Method Engineering and Meta-modelling: editorial. Information and Software Technology, 38(4):259.

Michael Brodie, John Mylopoulos, and Joachim Schmidt, eds. (1984) On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages, Springer-Verlag.

Magnus. Boman, Janis Bubenko, Paul Johannesson, and Benkt Wangler (1997) Conceptual Modeling, Prentice Hall.

BPMI. The Business Process Management Initiative. www.bpmi.org

Peter B. Checkland (1981) Systems Thinking, Systems Practice. John Wiley, Chichester, UK.

Peter Chen (1976) "The Entity-Relationship Model: Towards a Unified View of Data", ACM Transactions on Database Systems 1(1). 9-36.

Marybeth Chrissis, Mike Konrad, Sandy Shrum (2003) CMMI : Guidelines for Process Integration and Product Improvement. Addison-Wesley.

Lawrence Chung, Brian Nixon, Eric Yu, and John Mylopoulos (2000) Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers.

Alistair Cockburn (2001) Agile Software Development. Addison-Wesley.

Thomas Curran and Gerhard Keller (1997) SAP R/3 Business Blueprint: Understanding the Business Process Reference Model. Pearson Education.

Bill Curtis, Mark Kellner, and James Over (1992) "Process Modelling," Communications of the ACM 35(9), September. 75-90.

Thomas H. Davenport and J. E. Short (1990) "The New Industrial Engineering: Information Technology and Business Process Redesign," Sloan Management Review, 1990, 31 (4), pp. 11-27.

Thomas DeMarco (1979) Structured Analysis and System Specification, Prentice Hall.

Thomas DeMarco and T. Lister (1999) Peopleware, 2nd Edition, New York, Dorset House.

Giorgio deMichelis, Eric Dubois, Matthias Jarke, Florian Matthes, John Mylopoulos, Michael Papazoglou, Joachim W. Schmidt, Carson Woo, and Eric Yu (1998). A three-faceted view of information systems: the challenge of change. Communications of the ACM 41, 12: 64-70

ebXML. www.ebxml.org

Pelle Ehn (1988) Work-Oriented Development of Software Artifacts, Arbetslivscentrum, Stockholm.

Rino Falcone, Munindar P. Singh, and Yao-Hua Tan (2001) Trust in Cyber-Societies: Integrating the Human and Artificial Perspectives, Lecture Notes in Artificial Intelligence 2246. Springer.

General Accounting Office (GAO) (2002) Information Management: Challenges in Managing and Preserving Electronic Records. Report number GAO-02-586. GAO, Washington, DC, USA.

W. Wayt Gibbs (1994) Software’s Chronic Crisis. Scientific American (International edition) September 1994. 72-81.

George Gilder (1993) Metcalfe's Law and Legacy. Forbes ASAP, September 13, 1993. http://www.gildertech.com/public/telecosm_series/metcalf.html

Fausto Giunchiglia, James Odell, Gerhard Weiß, eds. (2003) Agent-Oriented Software Engineering III, Third International Workshop, Bologna, Italy, July 15, 2002. Lecture Notes in Computer Science 2585. Springer.

Joseph Goguen and Marina Jirotka, eds. (1994) Requirements Engineering: Social and Technical Issues. London: Academic Press.

A. Gomez-Peres and O. Corcho (2002) Ontology languages for the semantic web. IEEE Intelligent Systems 17, 1, 54-60.

John T. Gorgone, et al. (2002) IS 2002 - Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Association for Computing Machinery (ACM), Association for Information Systems (AIS), Association of Information Technology Professionals (AITP). http://www.acm.org/education/is2002.pdf

Sol J. Greenspan, John Mylopoulos, Alexander Borgida (1994) On Formal Requirements Modeling Languages: RML Revisited. International Conference on Software Engineering. ACM Press. 135-147.

Nicola Guarino and Christopher A. Welty (2002) Evaluating ontological decisions with OntoClean. Communications of the ACM 45(2): 61-65.

Michael Hammer (1990) Reengineering Work: Don't Automate, Obliterate . Harvard Business Review, July. 104-112.

J.R. Hauser and D. Clausing (1988) The house of quality. Harvard Business Review (3), May, 63-73.

HL7. www.hl7.org

Karen Holtzblatt and H.R. Beyer (1995) Requirements Gathering: The Human Factor. Communications of the ACM, 38(5): 30-32.

Michael Huhns and Munindar P. Singh (1998) Readings in Agents. Morgan Kaufmann, San Francisco.

Watts Humphrey (1995) A Discipline for Software Engineering, Addison-Wesley, Reading, MA.

IETF. The Internet Engineering Task Force. http://www.ietf.org/

Tomas Isakowitz, Michael Bieber, Fabio Vitali (1998) Web information systems. Communications of the ACM 41(7): 78-80.

ISO (1992) ISO9000 International Standards for Quality Management. Geneva, International Organization for Standardization.

Matthias Jarke, John Mylopoulos, Joachim Schmidt, and Yannis Vassiliou (1992) "DAIDA: An Environment for Evolving Information Systems," ACM Transactions on Information Systems 10(1), 1992, 1-50.

Matthias Jarke, K. Pohl, K. Weidenhaupt, K. Lyytinen, P. Martiin, J.-P. Tolvanen, M. Papazoglou (1998) Meta modeling: a formal basis for interoperability and adaptability. In Krämer, Papazoglou, Schmidt (eds.): Information Systems Interoperability, Wiley RSP, 229-263.

Rob Kling (1996) Computerization and Controversy: Value Conflicts and Social Choices. 2nd ed. Academic Press.

Philippe Kruchten (2000) The Rational Unified Process: An Introduction (2nd ed.) Addison-Wesley.

Axel van Lamsweerde (2000) Requirements Engineering in the Year 00: A Research Perspective 22nd International Conference on Software Engineering, Limerick, ACM Press.

Jintae Lee (1997) Design Rationale Systems: Understanding the Issues. IEEE Expert 12(3): 78-85

Kalle Lyytinen (1987) Different Perspectives on Information Systems: Problems and their Solutions. ACM Computing Surveys, vol 19, No 1, pp. 5-44.

Pericles Loucopoulos and R. Zicari, eds. (1992) Conceptual Modeling, Databases and CASE: An Integrated View of Information System Development, Wiley.

Steve McConnell (1996) Rapid Development: Taming Wild Software Schedules. Microsoft Press.

Abbe Mowshowitz (1997) Virtual Organization - Introduction to the Special Section. Communications of the ACM 40(9): 30-37

Michael J. Muller and Sarah Kuhn (1993) Participatory design. Communications of the ACM, 36(6) June. 24-28.

Deirdre K. Mulligan (2003) Digital rights management and fair use by design. Special issue. Communications of the ACM 46(4): 30–33.

John Mylopoulos (1998) "Information Modeling in the Time of the Revolution", Invited Review, Information Systems 23(3/4): 127-155.

John Mylopoulos, K. Lawrence Chung, and Eric Yu (1999) “From Object-Oriented to Goal Oriented Analysis”, Communications of the ACM, January 1999. 42(1): 31-37.

John Mylopoulos, Alex Borgida, Matthias Jarke, and Manolis Koubarakis (1990) "Telos: Representing Knowledge About Information Systems," ACM Transactions on Information Systems, 8(4): 325-362.

Hans W. Nissen and Matthias Jarke (1999) Repository support for multi-perspective requirements engineering. Information Systems. 24(2):131-158.

NIST (1993) Integrated Definition for Function Modeling (IDEF0). 1993, National Institute of Standards and Technology.

Ikujiro Nonaka and Hirotaka Takeuchi (1995) The Knowledge-Creating Company, Oxford University Press.

OASIS. Organization for the Advancement of Structured Information Standards. http://www.oasis-open.org/

OpenGroup. The Open Group. http://www.opengroup.org/

Michael Papazoglou and Gunther Schlageter, eds. (1998) Cooperative Information Systems: Trends and Directions. Academic Press.

Michael Papazoglou (2001) Agent-oriented technology in support of e-business. Communications of the ACM 44(4): 71-77.

Colin Potts and Wendy Newstetter (1997) Naturalistic Inquiry and Requirements Engineering: Reconciling Their Theoretical Foundations. Proceedings Third IEEE International Symposium on Requirements Engineering, Annapolis, January 1997. 118-127

Bala Ramesh and Matthias Jarke (2001) Towards reference models for requirements traceability. IEEE Transactions on Software Engineering. 27(1) 58-93.

Andrew Rosenbloom (2000) Trusting technology: introduction to special issue. Communications of the ACM, 43(12) December.

Douglas T. Ross and D. Schoman (1977) “Structured Analysis for Requirements Definition,” IEEE Transactions on Software Engineering 3(1), Special Issue on Requirements Analysis, January, 6-15.

W.W. Royce (1970) Managing the Development of Large Software Systems: Concepts and Techniques. Proceedings of WESCON, IEEE Computer Society Press, Los Alamitos, CA.

James Rumbaugh, Ivar Jacobson, and Grady Booch (1999) The Unified Modeling Language Reference Manual. Addison-Wesley.

SACMAT (2003) 8th ACM Symposium on Access Control Models and Technologies, June 2-3, 2003, Villa Gallia, Como, Italy, Proceedings.

SemanticWeb. The Semantic Web community portal. http://semanticweb.org/

Howard Smith and Peter Fingar (2002) Business Process Management: The Third Wave. Meghan-Kiffer Press.

Standish Group (1995) "Software Chaos", http://www.standishgroup.com/chaos.html.

Lucy Suchman (1995) Making work visible. Communications of the ACM 38(9) 56 -64.

Paul Timmers (2000) Electronic Commerce: Strategies and Models for Business-to-Business Trading. John wiley & Sons.

UDDI. www.uddi.org

UN/CEFACT. United Nations Centre for Trade Facilitation and Electronic Business. http://www.unece.org/cefact/

Panos Vassiliadis, Christoph Quix, Yannis Vassiliou, Matthias Jarke (2001) Data Warehouse Process Management. Information Systems, 26(3) 205-236.

W3C. The World Wide Web Consortium. http://www.w3.org/

WebServices. WebServices.org portal. http://www.webservices.org/

Dallas E. Webster (1988) “Mapping the Design Information Representation Terrain,” IEEE Computer 21(12):8-23.

Jerry Weinberg (1998) The Psychology of Computer Programming, Silver Anniversary Edition, New York, Dorset House.

Gio Wiederhold and Michael R. Genesereth (1997) The Conceptual Basis for Mediation Services. IEEE Expert 12(5): 38-47

Jane Wood and Denise Silver (1995) Joint Application Development. John Wiley & Sons. 2nd ed.

Jian Yang and Mike P. Papazoglou (2000) Interoperation Support for Electronic Business. Communications of the ACM 43(6): 39-47

Eric Yu and John Mylopoulos (1994) "Understanding 'Why' in Software Process Modeling, Analysis and Design," Proceedings Sixteenth International Conference on Software Engineering, Sorrento.

Eric Yu and John Mylopoulos (1994) “From E-R to ‘A-R’ -- Modeling Strategic Actor Relationships for Business Process Reengineering”, Proceedings Thirteenth International Conference on the Entity-Relationship Approach, Manchester, December 1994; P. Loucopoulos (Ed.), Lecture Notes in Computer Science 881, Springer-Verlag. 548-565.

Eric Yu, John Mylopoulos, and Yves Lespérance (1996) “AI Models for Business Process Re-engineering,” IEEE Expert 11(4).

Eric Yu (1997) “Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering," Proceedings IEEE International Symposium on Requirements Engineering, Annapolis, Maryland, 226-235, January 1997.

Eric S. K. Yu (2002) Agent-Oriented Modelling: Software versus the World. In: Michael Wooldridge, Gerhard Weiß, Paolo Ciancarini (Eds.): Agent-Oriented Software Engineering II, Lecture Notes in Computer Science 2222 Springer: 206-225.