

# WEB SERVICES AND EMERGENT ORGANIZATIONS: OPPORTUNITIES AND CHALLENGES FOR IS DEVELOPMENT\*

**Mauro Bello**

Dipartimento di Scienze dell'Economia e della Gestione Aziendale  
Università Cattolica del Sacro Cuore, Milano, Italy  
mbello@mi.unicatt.it

**Maddalena Sorrentino**

Dipartimento di Scienze dell'Economia e della Gestione Aziendale  
Università Cattolica del Sacro Cuore, Milano, Italy  
mso@mi.unicatt.it

**Francesco Virili**

Dipartimento Impresa e Lavoro  
Università di Cassino, Italy  
francesco.virili@eco.unicas.it

Dipartimento di Scienze dell'Economia e della Gestione Aziendale  
Università Cattolica del Sacro Cuore, Milano, Italy  
francesco.virili@mi.unicatt.it

## ABSTRACT

*We are living in exciting times, in which technological innovation and new forms of organization are advancing at a very fast pace. On the organizational side, the lack of stability of the so-called "emergent" organizations, those that are in continuous evolution and transformation, represents a big challenge for Information Systems Development practices and technologies: in a fast changing organization, there may be no "optimal" set of specifications for an Information System and even the traditional concept of the Information System life cycle may need to be replaced by one involving the idea of continuous development. On the technological side, the diffusion of Internet-based platforms and, in particular, the recent introduction of the so-called "Web Services" technological standard for dynamic component-based software development may represent a potentially interesting opportunity to build continuously changing Information Systems. In this contribution we start exploring this territory, focusing on the concepts of emergent systems and continuous development, explaining the major characteristics of Web services technology and the potential uses of it, and delineating the essential paths of a study in progress in which the final research objective is to evaluate the impact of*

---

\* While the article was written by the three authors, section 1 was contributed by Maddalena Sorrentino and section 3 was contributed by Mauro Bello; Francesco Virili wrote sections 2 and 4. Finally section 5 was contributed by all the authors.

*this new technological standard on Information Systems development practices. The first research stage, planned and depicted here in broad outline, would be to select an appropriate methodological framework and a few case studies for a preliminary qualitative evaluation of the adoption of Web services technology.*

## 1. INTRODUCTION

In a recent contribution, Truex, Baskerville and Klein (1999) pose an appealing and thought provoking question: are current Information Systems Development (ISD) practices still appropriate in today's organizations? They observe that one of the main assumptions in Information Systems Development in the last forty years has been organizational stability, an attribute that is often lacking in continuously changing, unstable, highly innovative business environments. In the presence of new forms of organizations that are dynamically evolving in a process of continuous transformation, the traditional Information Systems life cycles, essentially based on a static definition of requirements and the fixing of a supposedly "optimal" set of specifications may result in a disproportionate, rigid, uncomfortable costume that may make it difficult to move in and adapt to the changing environment. From this point of view, the advent of new technologies that facilitate "continuous redevelopment" practices is surely welcome.

We argue that the recently launched "Web services" technological standard may facilitate a migration to new Information Systems development practices more appropriate to the needs of "emergent" organizations. This recently defined technological framework has been conceived to make it possible to build, publish, retrieve and dynamically integrate software components into existing systems, using the Internet as communication media between the software components providers and the target systems.

In principle, this technology may have great potential to make it possible for existing systems to dynamically change and adapt to evolving business conditions: it would be possible to search for, integrate and execute at run time a specific software component, depending on the needs of the moment. In practice, much unexplored territory still has to be covered to make this vision real.

The aim of this contribution is to examine the first steps that have been taken in this direction. Our research question is: what is the potential role of web services for building dynamically adaptive Information Systems?

In the introduction we try to give an answer to this question by describing the first elements of a research study that is still in its embryonic stage. In section 2 we illustrate the Truex, Baskerville and Klein (1999) view of the needs of emergent organisations and the consequent shift from traditional development practices to a "continuous redevelopment" process. In this context we observe that the notion of "emergent" organization may be characterized in terms of contextual and structural variables or in accordance with alternative frameworks of analysis, though the actual selection of concrete cases of "emergent" organizations may well raise serious questions and problems in relation to such approaches. An alternative approach, based on a metaphor drawn from linguistics, distinguishes between a "deep structure" and an "emergent structure" in an Information System (Truex and Baskerville, 1998). This metaphor invites us to focus our attention on the "surface structure" of an Information System, i.e. the part that is much more exposed to environmental changes and that is much more liable to be subject to a process of continuous redevelopment.

In section 3 we briefly describe Web Services technology; even at first sight, there is little doubt as to its potentially deep impact on the entire software industry and on software development methodologies. In section 4 we present a broad, tentative outline of the design and methodological features of a research project that will be put into practice through a number of "test-drive" case

studies. Finally, in section 5, we evaluate the expected outcomes and offer some preliminary conclusions.

## 2. EMERGENT ORGANIZATIONS AND IS DEVELOPMENT

Truex, Baskerville and Klein (1999) argue that, given the assumption of organizational stability that has permeated IS development theories over the last forty years, Information Systems have typically been designed and implemented in organizations with the following five goals in mind:

1. formal analysis and design activities should be performed with great attention and, if necessary, at great length, in order to minimize maintenance activities;
2. user satisfaction should be pursued;
3. a complete set of abstract requirements should be created;
4. system specifications should be derived from the abstract requirements;
5. rigorous advance planning should inform the whole ISD process.

Are these goals still valid in the absence of organizational stability? The authors suggest not, introducing a concept of "emergent" organizations, i.e. organisations that "can be said to be in a state of constantly seeking stability, while never achieving it. [...] (Such organisations are) always in process; they are never fully formed." (Truex et al., 1999, p.117). Given the fact that the main features of such organizations are continuously evolving, "Having low-maintenance, stable information systems means the organization is continuously battling against its constraining information systems as it adapts to an ever-changing environment. In this scenario, IS will inhibit rather than facilitate organizational change" (*ibidem*, p.118). Some alternatives to the traditional concept of the Information System development life cycle, including those of prototyping, end-user development and open systems connectivity, have been used with some success to enhance ISD flexibility in changing environments, but they are still inadequate for emergent organizations, because they are mainly focused on determining a "final" solution iteratively or incrementally. In an emergent organization a "final" solution does not exist, and a continuous redevelopment perspective should be adopted. Obviously, in terms of this perspective, a low maintenance, stable system would constrain the organization, inhibiting its adaptability (*stable system drag*) and producing a negative impact on its performance. To illustrate the concept, Truex, Baskerville and Klein represent graphically the typical IS life cycle in a stable organization (Figure 1), where the higher costs of analysis and design can be seen to be justified by lower maintenance costs in the following phases. By contrast, in emergent organizations (see Figure 2) even a significant expenditure on the initial analysis/design process, would probably not compensate for the higher maintenance efforts (and costs) provoked by the continuous redevelopment process. Therefore, according to the authors, it is reasonable to surmise that a reduction in the relative level of expenditure on the initial analysis/design phases (dotted line) would be desirable, given that it would have little impact on the rest of the ISD process.

In a very thorough and stimulating analysis, the authors go on to extend their reasoning to all the five goals of the traditional ISD process mentioned above: for emergent organizations, "lengthy analysis and design are poor investments"; moreover, "user satisfaction is improbable [...]; abstract requirements are largely imaginary; [...]; complete and unambiguous specifications are ineffectual [...]; new systems projects denote ISD failure" (*ibidem*, p. 119-120). The last of these statements deserves particular attention here: in a stable organization, an Information System traditionally has a quite long, but limited lifespan. After a certain period (usually several years) the system starts to become obsolete, with rising maintenance costs. As represented on the right hand side of Figure 1, this is the period during which the system is replaced. If we shift to a concept of continuous redevelopment, illustrated in Figure 2, a system should never become obsolete, because it evolves dynamically with the organization. For this reason there should never be a new system replacing an old one. If it happens, it is a sign of the "utter failure of an existing computer based IS" (*ibidem*, Truex

*et al.*, 1999, p.120). In the rest of the analysis the authors redefine the goals of Information Systems Development, proposing a novel view in which systems, being subject to a process of continuous adjustment and evolution, are never fully specified.

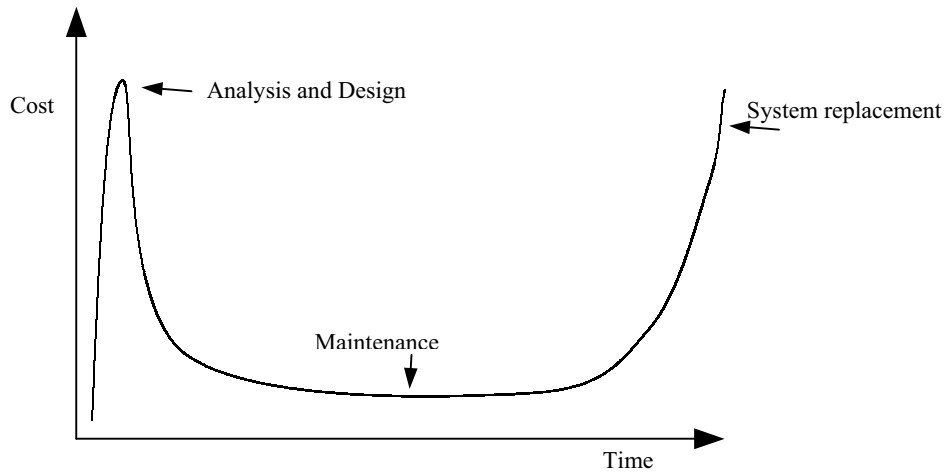


Figure 1. Typical IS life cycle economy. Source Truex et al, 2001.

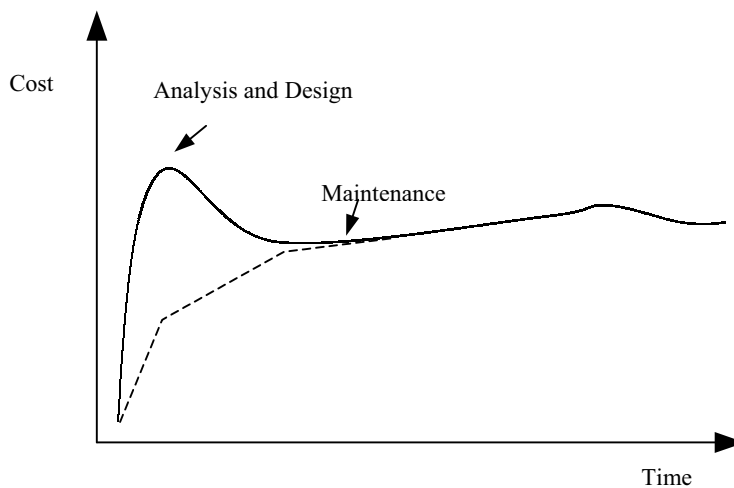


Figure 2. IS life cycle economy in emergent organizations. Source: Truex *et al.*, 2001.

The ideas of Truex, Baskerville and Kline are fascinating and provocative, but rather difficult to test in practice: one of the first points that needs to be clarified is the precise characteristics of an "emergent" organization. The authors write that emergent organizations "include many of today's commercial and governmental organizations" (*ibidem*, Truex *et al.*, 1999, p.117). One could try to apply some concepts and theories in organizational studies: for example, a contingency theory approach might be adopted, starting from the classical distinction between a mechanistic and an organic paradigm, introduced by Burns and Stalker (1961), and building on more recent contributions, like, among others, the analysis of Courtright (1989), where the organizational evolution and adaptation capabilities typical of the organic paradigm are taken into consideration. The more recent, well-known framework of Galbraith (1994) may be of help for the characterization in that it might be developed in terms of the five context variables described there (Strategy, Environment, Technology, Dimensions, Organizational Culture). In addition, the classical structural variables, like specialization, formalization and centralization (Pugh *et al.*, 1968) might be integrated into such an approach. A radically different approach to explain and describe adaptive, emergent forms of organization, is the "phenomenological" perspective adopted by Ciborra and Pugliese (1997): in this model technology, being the object of interpretation and sense-making by organisations, has here a more complex role. The relationship between

technology and organizational structure is bi-directional: the organization applies and interprets technology, trying to make sense of it, and in the course of this process it evolves into new forms and structures.

The underlying theoretical framework of Ciborra's work is based on the theory of structuration (Giddens, 1984). In this context, Ciborra discusses two new modalities of organizational design: "bricolage" and "second level learning". The idea of bricolage is the intuitive, in some way artistic recombination of available resources in accordance with a contingent situation and the needs pertaining to it. Many information systems have actually emerged in this way, and not in accordance with the classical, well-defined and planned system development process (Ciborra, 1991). Second level learning is the result of a radical innovation, obtained by using the available resources in creative and unusual ways. According to this view, system design should consist in promoting tactics, stimuli and paradoxes aimed at driving minds and behaviours out of traditional contexts and into new action schemes. The author provocatively describes this new way of "planning" strategic information systems with seven oxymorons, the first four of which refer to bricolage and the last three to second level learning: value bricolage strategically; design tinkering; establish systematic serendipity; thrive on gradual breakthroughs; practice unskilled learning; strive for failure; achieve collaborative inimicability. To thresh out the significance of this approach, we might point out that the "establish systematic serendipity" statement, for example, stresses the importance of systematically achieving innovation by catering for serendipity or, in other words, consciously making provision for the emergence of casual, unexpected, positive results obtained without a specific initial intent. This concept of serendipity is also found in some of the most recent organizational studies, where it is presented as one of the characteristics of innovative organizational forms (Grandori 1999, p. 536). Ciborra succinctly summarises the value of the overall framework: "These seven oxymorons can represent a new "systematic" approach for the establishment of an organizational environment where new information - and thus new systems can be generated. Precisely because they are paradoxical, they can unfreeze existing routines, cognitive frames and behaviors; they favor learning over monitoring and innovation over control" (Ciborra, 1991, p. 289).

While all the different approaches and theories cited above may be of help in identifying and investigating some of the distinctive traits of "emergent" organizations, it is highly unlikely that they alone will furnish a definitive solution to the problem of distinguishing between emergent and non-emergent organisations. One way to solve this puzzle may be to adopt a radically different point of view. In a recent paper Baskerville and Klein (1998) shift the focus from the distinction between emergent and non-emergent organizations to the individuation, within an organization, of a "deep" structure and an "emergent" structure, an approach inspired by the well-known studies (pioneered by Noam Chomsky) on emergent grammars and deep structure in linguistics. The concept is not new to the IS field: Wand and Weber (1995), for example, distinguish in an Information System a deep structure (the inner model), a surface structure (the interface) and a physical structure (the technologies), but the Baskerville and Klein contribution represents a comprehensive and deep investigation of the concepts and metaphors borrowed from linguistics. Adopting this view, we may focus our investigation on the "emergent" part of the Information System that is supposed to exist to a certain extent in any organization. Our planned empirical study, described in section 4, is designed in accordance with this hypothesis so as to give a first answer to our research question.

### **3. AN EMERGING STANDARD: WEB SERVICES**

#### **3.1 Introduction**

In April 2001, some 52 IT companies and 'power users' participating in the W3C consortium (including Microsoft, IBM, HP, Sun, SAP and Boeing) took part in a workshop in San Jose, California convened to advise the W3C on further actions to be taken in relation to Web services. All of the

participants published their 'position papers' (<http://www.w3.org/2001/01/WSWS>), discussing their peculiar views on how to introduce the new technology.

Web services are self-contained, modular business process applications that Web users or Web connected programs can access over a network (via a standardized XML-based interface) and in a platform-independent and language-neutral way [4] [5]. This makes it possible to build bridges between systems that otherwise would require extensive development efforts. Web services are designed to be published, discovered, and invoked dynamically in a distributed computing environment. By facilitating real-time programmatic interaction between applications over the Internet, Web Services may allow companies to more easily exchange information, leverage information resources, and integrate business processes.

In practice, a Web service is a reusable software component (i.e. a small functionality, a little 'piece' of an application) that can be written by somebody (for example, a software vendor) and published and that can then to be retrieved and dynamically used within an existing application by somebody else (for example, an IS developer). Adopting this framework, companies in the future will be able to buy their information technologies as services provided over the Internet, rather than owning and maintaining all their hardware and software (Hagel and Seely Brown, 2001). The functionalities that can be implemented by Web services have virtually no limits, ranging from major services like storage management and customer relationship management (CRM) down to much more limited services such as furnishing a stock quote and checking bids for an auction item.

Users can access some Web services through a peer-to-peer arrangement rather than by going to a central server. Some services can communicate with other services. This exchange of procedures and data is generally enabled by a class of software known as middleware. Services previously possible only with the older standardized service known as Electronic Data Interchange (EDI) are now likely to become Web services. Besides the standardization and wide availability to users and businesses of the Internet itself, Web services are also increasingly enabled by the use of the Extensible Markup Language (XML) as a means of standardizing data formats and exchanging data.

Through Web services systems can advertise the presence of business processes, information, or tasks to be consumed by other systems. Web services can be delivered to any customer device - e.g., cell phone, (PDA) and PC - and can be created or transformed from existing applications. More importantly, Web services use repositories of services that can be searched to locate the desired function so as to create a dynamic value chain. Web services go beyond software components, because they can describe their own functionality as well as look for and dynamically interact with other Web services. They provide a means for different organizations to connect their applications with one another so as to conduct dynamic e-business across a network, no matter what their application, design or run-time environment.

In theory, with this new software layer, it is possible to build applications without having to know who the users are, where they are, or anything else about them. Users of these applications will be able to source them as easily as they would static data on the Web, operating with complete freedom and without any concern about the format, platform, or anything else. Realistically, however, the issues of affordability, security, trust, performance, efficiency and the real feasibility of runtime software component selection and activation all need to be thoroughly investigated. Nevertheless, the revolutionary aspect of Web services, namely the fact that they are self-integrating with other similar applications, remains true. Until now using traditional software tools to make two e-business technologies work together has required lots of work and planning (i.e. reaching agreement on the standards to pass data, the protocols, the platforms, etc.) With the aid of Web services applications should be able to automatically integrate with each other, wherever they originate, without any additional work.

### 3.2 The architecture

The Web Services architecture, depicted in Figure 3, is based upon the interactions between three roles: service provider, service registry and service requestor [4]. The interactions involve the 'publish', 'find' and 'bind' operations. Together, these roles and operations function in relation to the Web Service software module and its description. In a typical scenario, a service provider hosts a network-accessible software module. The service provider defines a service description for the Web service and 'publishes' it to a service requestor or service registry. The service requestor uses a 'find' operation to retrieve the service description locally or from the service registry; it uses the service description to 'bind' with the service provider and to invoke or interact with the Web service implementation. The service provider and service requestor roles are logical constructs and a Web service can exhibit characteristics of both.

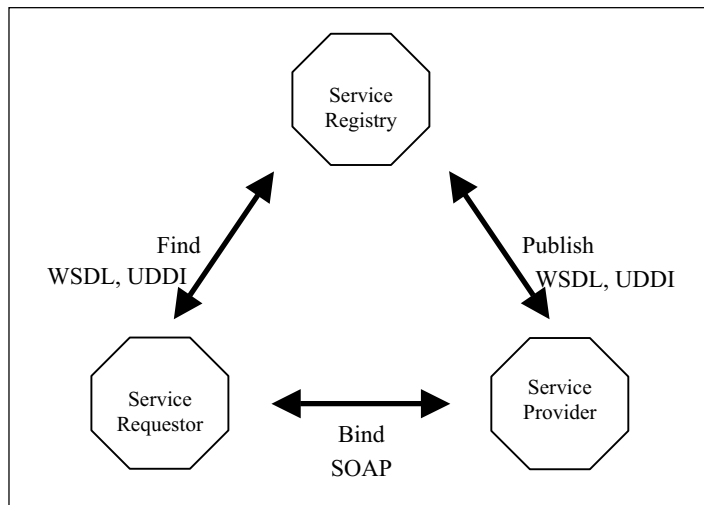


Figure 3. The Web Services Model, from Kreger (2001).

Web Services are based on a series of new standard technologies which meet the essential requirements for their implementation:

- SOAP (to convey) a standard mechanism for sending requests to services and for receiving responses (Barton *et al.*, 2000; Box *et al.*, 2000; Brown *et al.*, 2001; Gudgin *et al.*, 2001a, 2001b; Mitra, 2001);
- WSDL (to specify) a standard way to describe services, with input/output interface specifications and some *meta* information (copyright, version, update URL, etc.) (Christensen *et al.*, 2001);
- UDDI (to advertise and syndicate) a standard means of locating relevant services with the desired characteristics. See Table 1 (ARIBA *et al.*, 2000; Curbera *et al.*, 2001; Ehnebuske, 2001; McKee, 2001).

Interop Stack	Universal Service Interop Protocols (these layers are not defined yet)	UDDI is a “next layer” in an emerging <i>stack</i> enabling rich Web services. UDDI uses standards -based technologies such as TCP/IP, HTTP, XML and SOAP to create a uniform service description format and service discovery protocol.
	Universal Description, Discovery Integration (UDDI)	
	Simple Object Access Protocol (SOAP)	
	Extensible Markup Language (XML)	
	Common Internet Protocols (HTTP, TCP/IP)	

Table 1. UDDI scheme from (ARIBA *et al.*, 2000).

### 3.3 Web services development

In using Web services, as is usual with componentized software applications, the system development process may be fractioned in two major parts: the standard components development and the integration into the target system. Even if, due to the novelty of the platform, a consolidated methodology does not yet exist, we can delineate, so far as the development process of the standard components is concerned, four major phases that might be followed: building, deployment, running and management.

The 'build' phase includes the development and testing of the Web service implementation and the definition of the descriptions for both the service interface and the service implementation. Web services implementations can be provided by creating new Web Services, by transforming existing applications into Web Services or by composing new Web Services from other Web Services and applications.

The 'deploy' phase includes the publication of the service interface and service implementation definition to a service requestor or service registry and the deployment of the executables for the Web service in an execution environment (typically, a Web application server).

During the 'run' phase, the Web service is available for invocation. At this point, the Web service is fully deployed, operational and network-accessible from the service provider. At this point the service requestor can perform the find and bind operations.

The 'manage' phase covers the ongoing management and administration of the Web service application. Security, availability, performance, quality of service and business processes must all be addressed.

On the user side, the deployment of Web services in existing systems should not require any special effort or resources for application integration. This fact will certainly have great importance for developers in that it will have a dramatic effect on the way they design and implement Information Systems. In particular, it will involve them in focusing on how such systems may be adapted dynamically to respond to new business needs or organisational changes. The whole IS development process may well be radically transformed. As outlined by Lyytinen *et al.* (1998): '... the distinctions between 'internal' and 'external' applications have greyed. The impact of this greying is both the altering and the broadening of design considerations such as availability, security, support and access for all applications. In response to these issues new mechanisms and methods of application assembly are emerging. [...] These are a far cry from the application-oriented, data flow diagramming, functional design and bespoke application days of yore. Against these changes, the role of the software developer necessarily changes. Some will manufacture components; the majority will facilitate their adaptation, choice, understanding and use'. (*ibidem*, p.248).

To what extent will this process affect the traditional Information Systems Development methodologies? The research study depicted in the next section is an attempt to shed light on this question.

## 4. THE RESEARCH STUDY DESIGN AND METHODOLOGY

To evaluate the capacity of Web services to facilitate evolutionary development and to investigate how such a developmental process might differ from traditional Information System Development practices, we believe that it would be of particular interest to adopt an exploratory analysis based on a few "test drive" cases. Currently we are conducting a close exploration of this new field with a view to obtaining more information and insights on how to extend our research. In particular, we are investigating the possibility of conducting a longitudinal study where the temporal dimension may help us to understand the system dynamics. We anticipate that the whole project, including the choice

of methodological framework, and the selection, observation and discussion of the case studies, will last about 1 to 2 years.

As far as the specific object of the study is concerned, we are considering selecting a few ISD-based applications of Web services used in an information intensive industry (most probably, banking and finance). In several European countries the banking industry has recently undergone tumultuous innovation, in the course of which there has emerged a growing interest in Web-based distribution channels. As such, this industry appears a very suitable candidate for the study.

We anticipate that in general the object under analysis will not consist of an entire Information System but only of part of one, clearly a part implemented with Web services technologies. The qualitative analysis will essentially consist of a description of the functionalities of the new software, the development process and the role played by the new technology (Web services), all of which will be addressed with a view to how it compares with traditional ISD paradigms. Although we have not yet identified in a definitive manner an appropriate evaluation methodology, we are considering the adoption of the four-tiered framework recently introduced by Iivari *et al.* (2001). This has the advantage of being deep, comprehensive and open to the dynamic addition of new methodologies. On the other hand, we are concerned that this framework could turn out to be too structured and "well defined" to be able to offer an explanation of a radically new paradigm of ISD. In keeping with the illuminating analysis of Lyytinen *et al.*, (1998), we might regard Web services as an emerging standard in the so-called "internetwork computing architecture", a broad concept denoting "[...] all the technological components and associated standards that organize Internet based data transfer and computing, as well as the increasingly dense resulting digital interconnectivity between individuals and organizations" (*ibidem*, p. 241). What Lyytinen and his co-authors clearly state is that "software development changes *radically* in InterNCA" (*ibidem*, p. 248, proposition D4 - the italics are ours), adding that, as a consequence, researchers should "critically assess their current research methods and standards so that they can scale up with interNCA development" (*ibidem*, p. 250, proposition R1), and that they "should drop research approaches and topics that not meet the needs of the InterNCA platform" (*ibidem*, p. 251, proposition R2). In keeping with this view, we intend to adopt a flexible approach to the evaluation, using the framework proposed by Iivari (Iivari *et al.*, 2001) as a starting point and integrating the information acquired and classified by it with unstructured qualitative observation aimed at better describing the ISD process and the resulting system.

## **5. EXPECTED OUTCOME AND CONCLUSIONS**

What we have before us is a vast unknown territory, with only a few initial signposts to start the exploration. From our planned preliminary observations, based on the analysis of a few case studies, we expect to draw enough information to better focus and structure our future work. Together with Lyytinen *et al.* (1998), we foresee great and exciting opportunities for further research and understanding: "IS researchers must take a keen look at how large-scale and dramatic changes take place in infrastructure and how they impact on development forms. For example, studies of the impact of new protocols of object migration on legacy system maintenance alone would be worthy in any top-tier journal and would greatly add to our relevance" (*ibidem*, p. 251). Unfortunately, in such situations there is a great deal to discover but only a few consolidated tools and methodologies to drive the exploration. We have to proceed by trial and error, using qualitative observation and insights as well as the traditional evaluation and classification frameworks. The expected outcome in this phase is just to obtain information and insights on how to better focus and structure our research in order to be able, in future work, to more rigorously answer our research question and better understand the impact of Web Services technology on the Information Systems Development methodologies. In addition, new streams of research may be generated by this preliminary work, like, for example, an analysis of the use of Web Services technology and its impact on the cost structure in a networked organization environment (See, for example, Ring (1999) and references therein.). To achieve this ambitious goal,

we are putting our trust in the interaction and help of the IS community, confident that it will be possible to achieve not only organizational, but also ... "research" serendipity.

We would like to thank the anonymous reviewers who have commented this paper and Marco Cavallari who has given us helpful advice on earlier drafts.

## REFERENCES

- ARIBA, MICROSOFT and IBM (2000). *UDDI Technical White Paper*, [http://www.uddi.org/pubs/Iru\\_UDDI\\_Technical\\_White\\_Paper.PDF](http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.PDF).
- BARTON, J. J., THATTE, S. and H. FRYSTYK NIELSEN, (2000), SOAP Messages with Attachments, *W3C Note*, <http://www.w3.org/TR/SOAP-attachments>.
- BOX, D., et al., (2000), Simple Object Access Protocol (SOAP) 1.1, *W3C Note*, <http://www.w3.org/TR/SOAP>.
- BROWN, A., et al., (2001), SOAP Security Extensions: Digital Signature, *W3C Note*, <http://www.w3.org/TR/SOAP-dsig/>.
- BURNS, T., and G.M. STALKER (1961). *The management of innovation*, Tavistock, London.
- CHRISTENSEN, E., CURBERA, F., MEREDITH, G. and S. WEERAWARANA, (2001). Web Services Description Language (WSDL) 1.1, *W3C Note*, <http://www.w3.org/TR/wsdl>.
- CIBORRA, C. and Pugliese, S. (1997). La tecnologia, in Costa G. and Nacamulli R.C.D. (eds.), *Manuale di Organizzazione Aziendale, Vol. II: La Progettazione Organizzativa*, UTET (Ita)
- CIBORRA, C. (1991). From thinking to tinkering: the grassroots of Strategic Information Systems, *Proceedings of ICIS 1991, International Conference on Information Systems*.
- COURTRIGHT, J.A., FAIRHURST, G.T. and L. E. ROGERS (1989). Interaction Patterns in Organic and Mechanistic Systems. *Academy of Management Journal*, 32, 773-802.
- CURBERA, F., EHNEBUSKE, D. and D. ROGERS, (2001), Using WSDL in a UDDI Registry 1.05, *UDDI Working Draft Best Practices Document*, <http://www.uddi.org/pubs/wsdlbestpractices-V1.05-Open-20010625.pdf>.
- EHNEBUSKE, D., (2001), UDDI Version 2.0 Data Structure Reference, UDDI Open Draft Specification, <http://www.uddi.org/pubs/DataStructure-V2.00-Open-20010608.pdf>.
- DAFT, R. L. (2001), *Organizations Theory and Design*, South-Western College Publishing.
- GALBRAITH, JAY R. (1994). *Competing with flexible lateral organizations*, Addison-Wesley, Reading, Mass.
- GIDDENS, A., (1984). *The Constitution of Society: Outline of a Theory of Structuration*, University of California Press, Berkeley.
- GLASS, G., (2000) *The Web services (r)evolution, Part 1 – Applying Web services to applications*, IBM developer works. November 2000.
- GRANDORI, A., (1999). *Organizzazione e Comportamento Economico*. Il Mulino, Bologna (in Italian).
- GUDGIN, M., HADLEY M., MOREAU, J. J. and H. FRYSTYK NIELSEN, (2001a), SOAP Version 1.2 Part 1: Messaging Framework, *W3C Working Draft*, <http://www.w3.org/TR/soap12-part1/>.
- GUDGIN, M., HADLEY M., MOREAU, J. J. and H. FRYSTYK NIELSEN, (2001b), SOAP Version 1.2 Part 2: Adjuncts, *W3C Working Draft*, <http://www.w3.org/TR/soap12-part2/>.
- HAGEL, J. III, SEELY BROWN, J., (2001), Your Next IT Strategy, *Harvard Business Review*, October 2001, 105–113.
- IIVARI, J., HIRSCHHEIM, R. and H.K. KLEIN (2001). A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches. *Journal of Management Information Systems*, 17(3), 179-218.
- KREGER, H., (2001), *Web Services Conceptual Architecture*, IBM Software Group, May 2001.
- LYYTINEN, K., ROSE, G. and R. WELKE (1998). The Brave New World of development in the internetworking computing architecture (interNCA): or how distributed computing platforms will change systems development. *Information Systems Journal*, 8, 241-253.
- MCKEE, B., (2001), UDDI Version 2.0 API Specification, UDDI Open Draft Specification, <http://www.uddi.org/pubs/ProgrammersAPI-V2.00-Open-20010608.pdf>.

- MITRA, N., (2001), SOAP Version 1.2 Part 0: Primer, W3C Working Draft, <http://www.w3.org/TR/soap12-part0/>.
- PUGH, D.S., HICKSON, D.J., HININGS, C.R. and C. TURNER (1968), Dimensions of Organization Structure, *Administrative Science Quarterly*, 13, 65-91.
- RING, P.S. (1999). The costs of networked organization, in Grandori, A. (ed.), *Interfirm Networks*, Routledge, pp. 237-262.
- TRUEX, D.P. and BASKERVILLE, R. (1998). Deep structure or emergence theory: contrasting theoretical foundations for information systems development. *Information Systems Journal*, 8, 99-118.
- TRUEX, D.P., BASKERVILLE, R. and H. KLEIN (1999). Growing Systems in Emergent Organizations. *Communications of the ACM*, 42(8), 117-123.
- WAND, Y. and R. WEBER (1995). On the deep structure of information systems. *Information Systems Journal*, 5, 203-223.