Increasing the benefit of analysis:

The case of systems that support communication

Dov Te'eni[1], Roy Gelbard[2], Matti Sade[1]

[1] Tel-Aviv University
P.O. Box 39040
Tel Aviv 69978
ISRAEL

[2] Bar-Ilan University
Ramat Gan, 52900
ISRAEL

teeni@tau.ac.il
r_gelbard@yahoo.com
sadematti@hotmail.com

Abstract: Researchers in academia and practitioners in the field are concluding that object orientation (OO) supports well the stages of technical design and programming but not the stages of functional analysis and high-level design. Some further suggest that the UML diagrams cannot be used in systems analysis if it is relies on, as it usually should, feedback from the client because of their complexity. We advocate here an approach to analysis that concentrates on a select set of high impact areas of information systems in organizations but develops more sophisticated and theory based models of these focus areas. We demonstrate how this approach can produce a set of rich representations that enable improved modeling based on domain knowledge. We apply these representations to the design of systems that support organizational communication, providing a list of design principles that emerge from this discussion.

INTRODUCTION

Systems analysis has traditionally examined user requirements and modeled the corresponding system functionality. While academia seems to hold on to this view, we believe that in practice the field is dissatisfied with this approach and consequently is gradually giving up on investing effort and money in applying costly methodologies of analysis. Indeed, Avison & Fitzgerald [AF03] have recently asked whether we are witnessing the demise of systematic methodologies for systems development. We argue that it is not only a matter of reducing the effort needed for systems analysis, but it is important to simultaneously increase the benefit from analysis. In particular, we propose a focused effort in analysis that concentrates on high impact areas of information systems in organizations. We demonstrate how this approach can produce a set of rich representations that enable improved modeling based on domain knowledge. We believe that the combination of a reduced set of representations, on the one hand, and richer representations, on the other hand, enables a better understanding of information requirements and potential impact of information systems on organizational performance.

The paper is built as 3 steps (in three separate sections below), each step making a point with specific examples that we have developed:

- Systems analysis today, in the context of the current prevalence of the Object Orientation (OO) approach, is regarded as a relatively low-payoff activity (compared with say programming). To examine this claim, we report on a field study in progress of software development projects in Israel, revealing the practices and thoughts of highly qualified developers about systems analysis.

- We believe analysis should focus on four areas in which there is the greatest potential for benefit. Moreover, in these areas detailed representations of functional and contextual information can serve as a basis for applying domain knowledge to extend the implications of current analysis.

- Incorporating domain knowledge into the analysis produces new insights for design. To examine this claim, we present a theoretical approach to the development of systems that support organizational communication. We take one of the four areas designated above (organizational structure) and develop an abstract representation of structure as it relates to organizational communication.

THE PRACTICE OF SYSTEMS ANALYSIS AND POSSIBLE EXPLANATIONS

Academic studies [Re03], [AS03], [Ii96] and analysts' surveys, such as the Gartner convention for systems analysis and design, describe a similar situation, by which the OO approach strongly supports the design and programming stages of system development but poorly supports system analysis: "The design is good, the analysis is poor" [Du01]. Additionally, it is claimed there that the weakness results from the fact that "UML representations have not been effective in large-scale projects for context and communication." [ibid.]

Why is this the case? Taking a cost-benefit view of the developers' decision to use OO tools, suggests that either the costs are too high or the benefits too low. Indeed, Agarwal & Sinha [AS03] find that only the Class Diagram and Interaction Diagram are significantly perceived as user friendly, i.e., use of these tools is relatively easy, comfortable and clear. It is safe to assume that for users UML diagrams are even less easy-comfortable-clear to use. On the other hand, In examining the practice of software engineering, Reifer [Re03], raises concerns about benefits too, pointing at several issues that should be investigated to uncover the potential impact of software development activities. Among Reifer's major conclusions on areas undergoing research of special importance in analysis are formal modeling, process simulations and domain specific modeling, and the major areas in need of research are the systems context of the software and the emphasis on user behavior models.

Motivated by the discussion above, we examine the situation at the analysis stage in a variety of commercial software projects, attempting to examine the array of actual considerations behind low usage, not just of CASE tools, but also of the OO methodology. Is the low rate of usage due to technical limitations or due to methodological limitations? Do the methodological limitations arise from perceived problems specifically related to the various representation methods (the various UML diagrams)? Or from more basic reasons related to cost-benefit aspects of using the aforementioned methodology as is?

We suspect the latter, i.e., a sense that effort in analysis does not seem to be cost effective in terms of time investment and resources needed in comparison with the benefit of the obtained products. Accordingly, we designed a survey of project mangers and senior system analysts to unveil their use of the OO methodology and tools and the reasons behind their decisions to use (or not used) them.

The sample included 40 projects in 27 software companies. The projects ranged from up to one man-year projects (12 of the 40 projects) to more than 50 man-year projects (6), with projects consisting of up to 3 people in a project (13) to projects with over 50 people working concurrently (3). The software companies were in diverse industries (government, hi-tech, software houses etc.) and produced both software products for general use and dedicated systems.

The last author followed a predefined questionnaire during a semi-structured interview. Here are some examples of questions:
What is included in the stage of collecting customer requirements? How is this stage conducted and how are the requirements documented? When the interviewer decides that the answers are not detailed enough, he proceeds with more pointed questions such as what type of documents are prepared at this stage? Are there ready-made templates for documentation of requirements? Do you use any CASE tools for this stage? Who is involved in this stage?

Similarly, for each of the other analysis and design activities, we devised the appropriate questions. Moreover, we included open questions that probed the developers to reasons behind their decisions and comments.

The main results are summarized below. Table 1 shows at what stage projects defined objects, which of course is basis for OO. The distribution of projects indicates clearly that only 12.5% of the projects engaged in any sort of OO analysis.

	Projects	%
Full at analysis: All objects are fully identified and characterized in the analysis stage.	2	5%
Partial at analysis: Some of the objects or partial description of them is analyzed in the analysis stage.	3	7.5%
Full at design: All object are introduced and fully defined in other design stage.	10	25%
Partial at design: Only partial descriptions of the objects are defined in the design stage.	9	22.5%
None: Objects introduced only in programming.	16	40%

Table 1 - Object Definition in Analysis and Design

Table 2 shows the number of projects that used each of the visual representations. Fifty three percent of the projects used class diagrams and fifty six percent represent business processes using one of the appropriate diagrams (e.g., activity, sequence and collaboration). All other diagrams are hardly used.

Diagram	Usage	
	Projects	%
Use Case	1	3%
Package	13	33%
Activity	8	20%
Class)	21	53%
Sequence	9	23%
State Transition	7	18%
Collaboration	5	13%
Flowchart)	11	28%
DFD	5	13%

Table 2 - Diagrams usage

Furthermore, in response to the open questions why people underutilized OO diagrams we can point at four major reasons: 1. The time consumed in drawing diagrams is too great relative to the project schedule.; 2. The customers can hardly understand the diagrams and certainly not their implications; 3. The diagrams are not comprehensive because part of the requirements can be detected only later in the lifecycle; and 4. In projects involving relatively high user interaction, the screen designs capture all the necessary details.

A NEW DIRECTION FOR ANALYSIS – CONCENTRATING ON HIGH IMPACT REPRESENTATIONS

As in earlier studies, our survey reveals the extremely low rate of using OO analysis methods. Thus, it is not surprising that the usage rate of respective CASE tools is also low. And why is this?

We assume here that the organization (and also the individual developer) aspire to operate effectively so as to achieve their goals for an economical price in a reasonable time. In systems analysis, the main expense is the analysts' time. However, it turns out that the analysis goals cannot be clearly defined: the exact components for inclusion in the analysis nor the depth and detail required about each object are defined. If we compare the systems analysis stage to the architectural planning stage in a civil engineering project, one finds that the unclear methodology characterizing the planning stage of the information system does not exist at all in the architectural planning stage of a construction project.

Moreover, the outcome of the architectural planning (e.g., lot plan, construction plan, electrical, water, air-conditioning, carpentry, and gardening plans and quantities) most successfully uphold the following two criteria:

1. **Clarity and specificity** that can be understood by the customer as well as by the construction engineer

2. **Completeness** that promises full coverage of all the of construction components so there is no need to expand explanations or make changes in the implementation stage later.

The outcome of system analysis does not uphold these two criteria and, moreover, the analysis methodologies themselves (including the OO approach) do not provide quality indices and control measures, which are necessary to examine whether the two criteria are upheld. Clarity and specificity of the expression are necessary for the establishment of any communication channel, and, in particular, in communication channels between the analysis team and the customer, between analysis team and development team, and among members of the analysis team itself.

The clarity and specificity in the architectural analysis are achieved by a representation that allows an exact description of the many possible channels and forms of expression [Bj92]. That is to say: (1) visual representation language (2) explicit and unambiguous expressions (3) clear rules for usage of each of the expressions and for combining of the expressions.

The completeness in the architectural analysis, which ensures full coverage of all construction components, is achieved by: (1) richness of expressions/ building blocks (2) explicit attention to each aspect of constructions (lot, structure, electricity, water, air-conditioning, carpentry, and gardening)

Both the functional and the OO approach also use visual representations for communication (UML diagrams, DFD etc.). These diagrams represent the system and its varied characteristics. But are the expressions in this visual representation sufficiently unambiguous? Are the rules of their usage and their possible combinations all defined? Is the representation language rich enough? And do they indeed deal with and cover all the system components? Unfortunately, the answers to these questions are mostly negative.

First of all, the traditional Functional approach and the OO approach do **not** deal with the following system components: user interface, business logic, hierarchy and organizational structure, hardware infrastructure, software infrastructure, communication infrastructure, technical constraints and operational constraints. Furthermore, the **language is incomplete**, as it does not supply the building blocks and rules necessary to describe and represent many fundamental aspects of the system.

Regarding the ambiguous rules of usage, one can find duplications in the manner of usage of different diagrams, in the Functional and in the OO approaches. For example, in the Functional approach it is possible to describe data items in both the DFD Diagrams and the ERD. In the OO approach, it is possible to describe business processes with Activity Diagrams and Sequence Diagrams. Such ambiguity in analysis products raises questions on the **completeness of the analysis** products and the internal links between them as to if and how consistency is maintained. Furthermore, most representations do not have clear decomposition and stopping rules.

The OO approach in fact defines the necessary links between analysis components. However, a survey of the situation in the field shows that the actual analysis products do not contain many of those diagrams. In addition, the degree of detail changes between organizations, projects, diagrams and even between objects in the same diagram. If the actual analysis results do not include all the diagrams defined in the methodology, then they are incomplete. It is fit to restrict the matter and note the existence of clear and ordered decomposing rules and stopping rules when it comes to data entities representation, either with Entity Relation Diagram or in the static aspect of Class Diagram. These rules are defined in accordance with the 5.5 normalization forms (1NF-5NF + BCNF). However, it must be emphasized that these normal forms apply only to the static aspect of data entities and do not support decomposing rules or stopping rules for the dynamic aspects of data entities (Methods).

It is therefore no surprise that the system analysis products, in both the Functional and the OO approaches, are not only incomplete but are neither explicit nor unambiguous. Thus, as noted above, we believe the low usage of the OO approach in analysis is due to cost-effective aspects. In other words, the organizations' policies and practices of using these tools and methodologies reflect first and foremost cost-effective aspects. In the absence of clarity, in the absence of the coverage of all the system components, in the absence of decomposing and stopping rules, the ability to methodologically use these tools, in a cost-effective manner, is apparently low and thus not feasible for the organization /project.

Cost - Effective Oriented Approach

In light of this, current work recommends the consolidation of an alternate analysis methodology, a methodology that is focused only in those analysis components that can produce products that satisfy the following principles:

- A product whose production process withstands the cost-effective demands, i.e., a product composed of fewer component types (such as smaller number of UML diagram types) but in which the contribution of each component is greater.

- A clear, explicit and unambiguous product, so that there is not a gap between the representation and its actualization in the programming stage.

- A product whose construction is based upon clear decomposing and stopping rules.

- A product that supports examination and elucidation options.

- A product that adopts RAD principles (Rapid Application Development).

Examination of the Functional approach, the OO approach and RAD values, in light of the aforementioned principles, enables defining a cost-effective analysis methodology, based on the following four areas of analysis:

1. Representation of organizational structure (e.g., Organizational chart).
2. Representation of information (e.g., ER diagram only with no dynamic aspects)
3. Representation of user models (e.g., images of screens)
4. Representation of business processes (e.g., simulated process diagrams)

These areas are consistent with our empirical findings. Representation of information fits those 53% using class diagram. Representation of business processes comes along with those 56% using activity, sequence or collaboration diagrams. Other diagrams were hardly used. Representation of user models fits dominant response to the open question, why people underutilized OO diagrams, saying that in projects involving relatively high user interaction, the screen designs capture all the necessary details.

Each of these areas will require new and sophisticated tools for representation. We believe they will be developed in the future and will probably be tailored towards different types of systems. In the next section, for instance, we develop the representation of organizational structure in the case of communication support systems. However, for now, we indicate for each area the kind of commonly used diagrams and other tools to demonstrate our suggestions. They will usually apply to data intensive systems.

1. Organizational structure representation

The organizational chart is meant to reflect the HR picture in the organization. Thus, the decomposing and stopping rules of this representation are clear and precise. Likewise, this component is implemented as is in the framework of user groups and authorizations in the system. Thus, this product satisfies the demands defined for analysis methodology components.

2. Information representation

Information (including data items and knowledge) is represented at various levels. For example, attributes, classes, structures of classes etc. When dealing with data intensive systems, and as was described above, data items representation using Entity Relation Diagram, is supported by clear and ordered decomposing and stopping rules, defined within the normalization framework. Likewise, this component is implemented as is on the basis of the data. Therefore, this product upholds the demands defined for the components of the analysis methodology.

3. User models representation

Representation of user models, commonly represented by user interfaces in “moving-graphics” format (such as that achieved by combining interface graphics with tools such as Visio, and operation imaging with Power Point) is most effective and economical, even though it is not of use later in the programming stage. Imaging based on interface graphics implements, in effect, the RAD principles and supplies the communications platform for the elucidation and integration of the manner of operating the desired system.

Although today, there is no way to generate the interface code from these graphics, this product clearly and specifically reflects the system interfaces since decomposing rules are equivalent to common principles of the use of UI Controls. And the stopping rule is by achieving an accurate description of the desired interface. In other words, by achieving a description that accurately defines the development required from the programmers. Therefore the components of this analysis withstand of the demands defined for the components of analysis methodology.

4. Business processes representation

Graphic representation of user interfaces provides the communication platform for elucidation and integration of the operation method of the desired system. With that, such a graphic representation, like UI prototype implemented with some programming language-environment, does not support - neither in simulation of the flow of business processes nor in the simulated analysis of the bottlenecks and loads expected in every process, according to the range of assumptions regard the environment and regard system resources. Thus, in order to implement such a simulated analysis, there needs to be a representation method supporting both **stages and conditions representation**, such as Process Chart, Activity Diagram or Sequence Diagram; and **simulation capabilities** of each process presented separately as well as the overview of all processes, like in the simulation enabled by BPM tools (Business Process Modeling).

BPM is a good example of implementing this areas because of its simulation capabilities but also because its vocabulary is much richer than the vocabulary of expression in UML diagrams such as the Process Chart, Activity Diagram and Sequence Diagram. Note for example BPM's expressions for Event frequency, “Production capacity”, Definition of the business logic, Identification of the program component, responsible for each event/activity/relationship and so on. Thus, adding such characteristics to Functional and OO representation method is necessary for clarity and completeness. Adding simulation abilities is necessary to enable better planning of service architecture and of system resources. With these additions, this component would also withstand the demands defined for the components of analysis methodology.

In conclusion, we recommend focusing on these four areas to increase the cost-effectiveness of analysis. Other aspects can be deferred to later stages. Whatever does not affect the database schema, user interfaces and business process architecture can be dealt at later stages. Only in this manner we will be able to establish a quicker, more accurate and effective systems analysis process.

A THEORY BASED APPROACH TO COMMUNICATION SUPPORT SYSTEMS

In this section we apply one of the four areas of analysis designated in the previous section to an analysis of information systems that support organizational communication, which we call them communication support systems (CSS). Although our discussion has so far drawn on the development of specific information systems, we feel the examination of a general type of systems such as CSS is especially illuminating. We concentrate on representing organizational structure as part of the analysis, touching only briefly on some of the other areas of analysis. In particular, we wish to highlight the potential impact of capitalizing on domain knowledge (in this case, a model of organizational communication) in modeling the information systems. We therefore first introduce an abstract representation of organizational structure, then the theory of organizational communication and its implications, and finally the impact of applying the theory on the representation.

The representation of organizational structure

The representation of organizational structure should be taken in its broadest sense in order to capitalize on its impact on information systems. An organizational chart showing formal roles in the organization is a very common but limited representation of structure. Te'eni and Schwartz [TS06] represent the structure of the organization in terms of relationships between (human and non-human) actors of the organization. These relationships can be represented on five dimensions: space, time, knowledge, inter-personal relations and functional interdependencies.

- *Space* concerns the relationship in space between actors, e.g., the geographical distribution of the organization. CSS often stretch the span of the organization.

- *Time* concerns the distribution of work or actors along time, e.g., do several workers perform work simultaneously, do projects consist of tasks organized on some time line, etc. IT helps overcome time constraints and span the organization.

- *Knowledge* concerns management and application of knowledge, assuming here that knowledge is known by actors. It includes the strategic issues of what knowledge is stored, how it is distributed among actors (including information systems) and how it is managed. IT drastically changes our capacity to manage knowledge.

- *Interpersonal relations* concern the relationships, attitudes, collegiality and so on between actors. This is usually an informal dimension of organization but nevertheless may be instrumental in determining the feasibility and quality of communication. IT has mixed effects on relations that must be considered by the analyst.

- *Functional interdependencies* characterize the organization of roles and allocation of tasks to actors, and the consequent interdependencies between tasks and actors. Some organizations are tightly coupled, while others are not; some are hierarchically organized, others are flat, and so on. IT often flattens the organization but sometimes enables centralization.

These five dimensions of organizational structure provide a richer understanding of its interplay with communication.

In order to fully capitalize on the use of CSS we must first understand its complex impact on the specific actions that actors perform in the organization, including their interdependencies. But we must also understand the impact of CSS on organizational structure, which in turn affect the actions. This dynamic picture complicates our analysis but is necessary to gain a more realistic view of the multiple impacts of CSS.

One example of the interaction between CSS, Organizational structure, and communication can be seen with respect to the growing importance of virtual organizations as discussed by Schwartz, Divitini, and Brasthevik [SDB00]. CSS (based on Internet, intranet, and extranet technologies) foster an environment in which both the virtual characteristics of the organization (e.g., capacity to serve online) and its communication activities (e.g., support services to customers) are enhanced. But virtual characteristics and communication create their own cycle of growth in which virtual characteristics increase the ability to communicate, and the increased communication ability in turn fuels the organization's ability to adopt and assimilate virtual characteristics. Such indirect (and often unintended) changes are not new but they can be identified systematically only when the analyst has this expanded view of the organization. For example, Van Dijk et al. [VDB03] discuss unintended changes in the strategic roles and activities of actors in supply chains and networks following the introduction of inter-organizational systems in the food industry.

Organizational communication theory and its design implications

The domain knowledge we wish to apply to the analysis of communication support systems (CSS) is communication theory. Te'eni [Te06] shows how such domain knowledge informs design. According to Habermas [Ha84], goal oriented communication takes place within a context of values, norms and resources. In order for goals to be achieved, coordination between communicators is necessary, as is their commitment to act appropriately. Each communicative act consists of three components: its propositional content (core message), its context or background information and the speaker's intention. Intentions are of different forms such as assertions (represent real-world states), directives (attempts to get the hearer to act), and expressions (feelings and attitudes). The role of context is to support the core message. The intentions, drawn at a higher level, govern the content and its context. How can these components be designed to produce more effective communication?

According to Habermas, comprehensible, trustworthy and appropriate communication is at once an act of building mutual understanding and relationships between the communicators. Mutual understanding is necessary for agreement and task accomplishment, and relationships are necessary for gaining commitment between communicators. All too often, CSS ignore one or the other, for example, ignoring relationships in structured transactions and mutual understanding in unstructured IRC. This discussion results in Principle 1 (see Table 3).

#	Principle of CSS design
1	Design must simultaneously consider enhancing mutual understanding and promoting relationships between communicators
2	Design should support adaptive behavior, including the contingent use of alternative communication strategies, alternative message forms and alternative media.
3	Design should monitor complexity and alert communicators to a high propensity of communication breakdowns, including those related to dimensions of organizational structure.
4	Design should support multiple levels of communication and easy travel between levels.
5	Organizational memory should consist of speech act components, situations, norms and values. .
6	Organizational memory should consist of associative information, accessible through multiple media, and represented in multiple forms, allowing for indeterminate and emergent views too

Table 3 – CSS Design Principles

In a similar fashion, the other five principles are developed [Te06]. Importantly, these principles can be related to the five areas of analysis designed earlier. In particular, the principles are tied to organizational structure (e.g., Principles 1, 3), information items (Principles 5, 6) and user models (Principle 2). (As this is a generic system to support communication business processes are reduced to communication processes and technical feasibility is not relevant.)

Clearly, this example demonstrates an unorthodox thinking in systems analysis. It would be very different for a standard transaction processing system (see examples of in the previous discussion of the four areas of analysis). Nevertheless, the approach is the same. For instance, here we talk of information organized as organizational memory, in transactions we talk about information in databases. The idea is that the domain knowledge informs the characterization and representation of information. For instance, in order to support communication effectively, it must include the information needed to support adaptive, appropriate and multi-level communicative action (principle 6 developed by Boland et al. [BTT94]).

So, on the one hand we have a theory of communication (leading to the 6 principles) and on the other hand we have a rich representation of organizational structure, as part of the cost-effective analysis. We now combine the two. A knowledge-based mailer called kMail [ST00] is an example of applying some of the principles outlined above. This CSS aims to provide contextual information when needed. It builds links to relevant information automatically by parsing outgoing messages to detect possible information that elaborates the message. In order to decide when and how much contextual information should be supplied it must combine domain knowledge (the theory of communication outlined above) with the results of analysis. In particular, it uses the representation of organizational structure to calculate the complexity of the communication. For example, distance in space, distance in time and differences in knowledge are all indicators of communication complexity [Te01] and call for a higher level of contextual information. Note that this result could not be achieved with standard representations of systems analysis that cannot produce a rich representation of organizational structure. The representation is therefore necessary but not sufficient. It is only the application of domain knowledge to the representation yields this extra benefit to analysis.

CONCLUSION

There is a substantial discrepancy between the prescriptions of systems analysis as they appear in our textbooks and the practice of systems analysis. In practice, only a small fraction of the recommended tools and methods are applied. In practice, there are no clear rules for what to represent, at what granularity to represent and how to move from representation to design. In practice, feedback loops from client to developers in order to improve representations are usually defunct. Practitioners see relatively little payoff on the effort involved in detailed representations.

We offered a way of thinking about analysis that has a potentially higher payoff to organizations and the developers. We suggest that analysis limit the range of tools and representations to a subset of four areas that have the highest potential impact on systems definition, namely organizational structure, information, user models and business processes. However, we suggest that the field goes deeper into these four areas to develop more potent representations. With these richer representations we will be able to reap the benefits of more sophisticated modeling and, in particular, apply domain knowledge to determine new insights and new valuable functionality to improve organizational performance.

Bibliography

[AF03] Avison D.F. and Fitzgerlad G. 2003. where now for development methodologies?, Communication of the ACM, Vol.46, No.1, pp.79-82.

[AS03] Agarwal R. & Sinha A.P., (2003), “Object-Oriented Modeling with UML: A Study of Developers’ Perceptions”, Communication of the ACM, Vol.46, No.9, pp.248-256.

[Bj92] Bjoerk, B.C. (1992), “A unified approach for modelling construction information”, Building and Environment. Vol. 27, no. 2, pp. 173-194

[BTT94] Boland Jr., R. J., Tenkasi, R. V. and Te’eni, D. 1994. Designing Information Technology to Support Distributed Cognition. Organization Science, 5(3) 456-477.

[Du01] Duggan J., (2001), "Modeling and Methods: the Keys to the Quality Kingdom", Gartner Analysis & Design Summit, San Francisco, 3-5 December, 2001.

[Ha84] Habermas, J. The Theory of Communicative Action: Reason and Rationalization of Society, 1984, Volume 1, Boston, Beacon Press.

[Ii96] Iivari J., (1996),”Why are CASE tools not used?”, Communication of the ACM, Vol.39, No.10 pp.94-104.

[Re03] Reifer D.J., (2003),”Is the Software Engineering State of the Practice Getting Closer to the State of the Art?”, IEEE Software, November/December, pp.78-83.

[SDB00] Schwartz, D.G., Divitini, M., and Brasethvik, T. On Knowledge Management in the Internet Age. In Schwartz, Divitini, and Brasethvik (eds.), Internet-based Organizational Memory and Knowledge Management, Hershey:Idea Group Publishing, (2000), pp. 1-23.

[ST00] Schwartz, D.G. and Te'eni, D. 2000. Tying Knowledge to Action with kMail, IEEE Intelligent Systems, 15(3) (May-June) 33-39.

[Te01] Te’eni, D. 2001. A Cognitive-Affective Model of Organizational Communication for Designing IT, MISQ. Rev., 25(2) 251-312.

[Te06] Te'eni, D. 2006 Communication Support Systems, Communication of the ACM, Vol.49, No.5, pp. 65-70.

[TS06] Te'eni, D.. and Schwartz, D.G 2006. Knowing to communicate for knowing in the learning organization, Proceedings of the Gerry DeSanctis Conference, Duke University, NC, March 3-5, [VDB03] van Dijk, S., Duysters G., and Beulens, A., Transparency dilemmas, information technology and alliances in agriculture and food industry, Eindhoven Centre for Innovation Studies, Working Paper 03.23 Department of Technology Management, Technische Universiteit Eindhoven, The Netherlands, (November 2003).