Discovering Problem-Solving Knowledge in Business Emails
Traceability in Software Design Using Computer Mediated Communication

Rauscher Francois, Matta Nada and Atifi Hassan
Institute ICD/Tech-CICO, University of Technology of Troyes,
12 rue Marie Curie, BP. 2060, 10010 Troyes Cedex, France

Keywords: Knowledge Management, Traceability, Project Memory, Pragmatics, Problem Solving.

Abstract: This article addresses issues related to problem solving knowledge detection in business emails. The objective is to discover if there are traces of knowledge left in emails and how to detect it. Finding patterns and structure of interest in emails corpora is a difficult task, but it is a necessity for companies in order to keep traces and learn lessons from previous projects. Especially in software development project that can be very costly and time consuming. In this paper, we present a general method for discovering problem solving requests in professional emails using mixed techniques from pragmatics analysis and knowledge engineering. This study takes place in the project memory work aiming at the traceability and structuration of knowledge in daily business environment.

1 INTRODUCTION

Computer mediated communication is ubiquitous in Software design projects. Email is used for project coordination, but also for design, implementation and tests. Especially with currents agile development methods, it is very common to interact through computer meditated communication like email, instant messaging and other collaborative tools in order to express functional needs, notify of issues and take appropriate decisions.

The knowledge produced in this types of interactions is often buried inside email boxes, hence being volatile and not easily reused. Project Memory (Matta et al, 2000) aims at describing organizational and cooperative dimensions of knowledge created during the life-cycle of a project. Our objective is to discover if we can structure and extract knowledge from professional emails in order to trace some phases and decisions inside a project memory. In particular problem solving of a project. Altough some works have been done on emails related to topics classification or spam detection (Jindal, 2007), it's rarely used in the context of knowledge management. The use of email is often confined to coordination and planning tasks (Wasiak et al, 2011), (Matta et al, 2010) or in case of legals issues. Enron case turned into a rich corpus for research about communication and social networks (Diesner, 2005).

The paper is organised as follows. Principles of problem solving, software design methods, and Project Memory are presented in Section 2, followed by some related works on emails and pragmatics analysis in Section 3. In Section 4, we propose an hybrid method using Knowledge engineering, NLP and pragmatic linguistic to discover meaningful features and context from emails. Then evaluations and results are reported in Section 5.

2 OVERVIEW

Problem solving plays a central role in software design projects. At first in the initial analysis that leads to specification documents, and then during the life-cycle of the project with the implementation, debugging and testing. This is especially true if the development follow an agile method, with several roundtrips from design to delivery.

Project Memory focus on keeping "project definition, activities, history and results" (Tourtier, 1995). Problem Solving is an essential part of design rationale as it tackles with problem definition, suggestions, and decision.

Francois R., Matta N. and Hassan A..
Discovering Problem-Solving Knowledge in Business Emails - Traceability in Software Design Using Computer Mediated Communication.
DOI: 10.5220/0005073601860193
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2014), pages 186-193
ISBN: 978-989-758-050-5

Face to face meetings are commonly used in office to manage projects and do collaborative work, but other mediated communications usages are increasing like phone, emails or instant messaging. However, information in emails is volatile, unstructured and distributed among users email boxes, making it difficult to trace and keep for corporate memory.

2.1 Problem Solving

Theory of Human Problem Solving was developed from the work of Newell and Simon (Newell et al, 1972) and provided the basis of much problem solving research. According to Hardin (Hardin, 2002), "Any problem has at least three components: givens, goal and operation". This general definition from problem solving theory is bringing keys elements into light:

- Givens: information and facts presenting context;
- Goal: desired end state;
- Operations : actions to be performed to reach end state;

In our present study, related to software development, we will focus more on givens and goal, i.e. the "problem recognition" part, the operations being part of the solution. When designing software complex problem solving arise more easily, because the tasks are abstracts and often not well structured as opposed as designing a real world artifact.

Problem solving in professional project aims at transforming knowledge into business value (Gray, 2001). This usually involve two types of knowledge: declarative (about facts, events, and objects) and procedural (knowing how to do things).

2.2 Software Development Process

Methodologies in software development evolved quickly in the last two decade from classic waterfall model to Extreme Programming and Agile methods (Beck et al, 2001).

Agile development is iterative and incremental with continuous delivery. As a side effect roundtrips between product-owner (contractor) and product-manager (developer) are more frequent, leading to increased communication and collaborative work. Typically a software design cycle in agile is divided into sprints, where the product-owner meet the product-manager (developer) and validates recent features, raises issues and express new needs. Problem Solving sequences happen on weekly (sometime daily) basis and implies all the actors of the project, not only the development team. With the new means of communication and project management methods like Kanban (Ladas, 2009) this occurs frequently through computer mediated exchanges.

2.3 Project Memory

A Project Memory is similar to a corporate memory of an organization at the scale of a project. As such, it contains knowledge used and produced during the project realization. The following elements and relations between them constitute the Project Memory:

- Organization: teams, members, tasks, roles, competencies, etc.;
- Resources and constraints: rules, methods, directives, time, budget, etc.
- Project realization: problem solving (problem definition, suggestions, and decision), solution evaluation (arguments, criteria), etc.;
- Project goals and objectives

The problem solving is part of the project design rationale memory (Matta et al, 2000). Some technics from knowledge engineering like REX (Malvache et al, 1993) or MASK (Matta et al, 2002), aims at knowledge capitalization, but others are more oriented on the traceability of the design rationale. A clear representation of the context and design rationale can be found in (Bekhti et al, 2003) and is presented in Figure 1

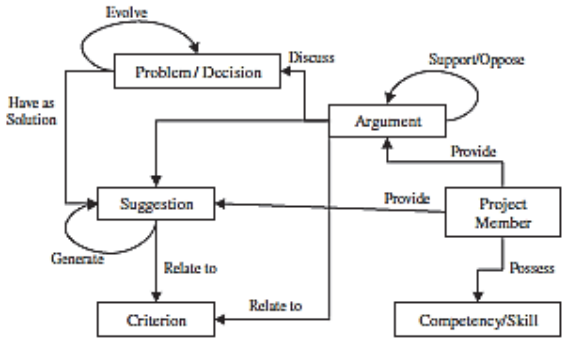


Figure 1: Problem solving and design rationale in context.

3 RELATED WORKS

Email has been analyzed by linguists like Baron (Baron, 1998) because of its hybrid nature. It

borrows the form of written language, but allows some kind of dialog due to its rapidity and the structure of the exchanges. Differences lays in the facts that it's asynchronous and that the emitter and receiver, not seeing each other are missing nonverbal elements of communication. Herring (Herring, 2004) shows that email is close enough to dialog in order to apply computer mediated discourse analysis methods to it.

Statistical Textmining methods are not well adapted to emails analysis due to its short content. Pragmatic analysis is a relevant approach for that but the choice of the speech acts coding scheme is crucial.

This justify the work of Carvalho (Carvalho et al, 2006) that describes an NLP technic based on N-grams to classify "email speech acts", such as "propose a meeting" or "commit to a task". This custom coding scheme is based on the idea of Searle's speech acts theory but created from the analysis of an academic emails corpora (CSpace).Similarly we can find in De Felice works on email (De Felice et al, 2012) another grid of analysis coming from linguistic pragmatics and psychology, the Verbal Response Mode (VRM) from Stiles (Stiles, 1992). VRM is attractive because of double coding form/intent of speech acts. But it's more suited for spoken discourse analysis and De Felice switch later to a custom scheme when analyzing Help Desk emails.

Another approach was taken by Kalia (Kalia et al, 2013) working on Enron corpus, he used two types of speech acts from Searle (Searle, 1969): directive and promissive in order to identify tasks and commitments.

Finally in our study we will use pragmatic analysis on emails with a custom speech act grid using context knowledge in a similar way as in (Matta et al, 2010) to identify problem solving interactions. In this latter work, pragmatics was used on discussion forums to identify criteria that help analyzing messages of coordination in design project.

4 COLLECTING PROBLEM SOLVING KNOWLEDGE

First we assemble and sanitize data, then in consecutive steps we apply treatments on data to identify relevant threads and finally analyze messages.

4.1 Project Data

We consider in our study that the initial data would be:

- Email corpus;
- Actors role, skills and competences;
- project phasing and specification documents;

These elements are likely to be found in numerous software design projects. They belong to the project context, which is important to represent in a project memory.

4.2 Corpus Preparation

Business emails collected from a project in their raw form are very redundant. In case of multiple replies or forward, several parts of the messages are repeated (e.g. quoted reply content). This occurs typically in long threads, mediated equivalents to spoken conversations, which are especially interesting for our study. Some preprocessing steps have to be performed in order to prepare messages and threads for analysis. We chose a deliberately simple method similar to Carvalho (Carvalho et al, 2004). The steps involved are:

- Remove all previous message text from reply;
- Keep previous message in case of first reply of a thread or forwarded email cause it carries context information;
- Remove signatures and disclaimers when possible (identity of sender and receivers are kept in email metadata);

This leaves us with a corpus of messages and threads without too much duplicated or useless information. For some treatments, the granularity at message level is not sufficient, and it's relevant to split the messages into sentences. Here again, we use a standard approach and split according to punctuation and paragraph signs.

4.3 Messages Analysis

We proceed in 3 steps to narrow our research for problem solving elements

4.3.1 Frequency

The first type of analysis is frequency based. We just count the total number of message per day, regardless of the quality of receivers (i.e. not making a difference between to, cc and bcc). This overall analysis will allow us to detect if some spikes of

activity do occur. More frequent or longer threads of conversation are a good indicators of actors need to collaborate, very often due to an issue. We analyze only messages that have more than 4 answers; we believe that knowledge can be extracted based on interaction.

We align this frequency analysis to the phases of the project, in order to detect boundaries and zones that are meaningful for choices, delivery problem or new needs.

4.3.2 Topics

We decide to make a very straightforward and knowledge oriented classification of messages and sentences. This steps is necessary in order to assert to deal with messages concerning pure software functionality knowledge and to filter project coordination emails.

Our approach is to create a keywords dictionary for the main topics of the project. This dictionary can be built from the following sources:

- Project phasing and specifications documents ;
- an expert;
- domain ontology if available;

As in project memory context, we choose not to rely on statistical NLP clustering like in (Cselle et al, 2007) but to use existing context knowledge. This dictionary is voluntarily kept simple and have the form:

Topic1: *keywords1*, *keywords2*... *keywordsn*.

Using this dictionary we classify messages into weighted topics vector (same technic is applied to sentences for a fine granularity analysis). In order to do that we use a cosine similarity based algorithm. We compute a Lucene (Gospodnetic, 2004) ranking between our message and each topics. This give us a topics matrix T where T_{ij} represents weight of topic j in message i.

As a side remark, keywords chosen in topics shall not overlap too much to keep the results significant.

4.3.3 Problem Solving and Pragmatics

The filtering technics used in Section 4.3.1 and 4.3.2 allow us to detect some threads of interest were problem solving is likely to occur. At this point, we switch to a sentence level analysis within one thread.

We have information metadata about sender/receivers, their roles and skills, timestamps of messages, and the project topics they are dealing with. We then apply pragmatics technics in order to find problem solving sentences.

As first work, we focus our speech act analysis on a part of problem solving by identifying request. In doing so we are restricting our search on finding the given and goal parts of problem-solving theory as seen in Section 2.1.

In further work, we will use subsequent messages in same thread to discover informations about operation and eventual solutions. Our analysis is based first on pragmatics in order to characterize request speech act, and that by identifying request verbs and forms.

Traditionally there is a debate between linguists and pragmatics points of view about question and request. A question is more a request for information (ask for a saying in (Benveniste, 1966)) and a pragmatic request according to Searle is more a request for action. In this study we consider a request as a directive speech act whose purpose is to get the hearer to do something in circumstances in which it is not obvious that he/she will perform the action in the normal course of events. This speech act can be performed directly or indirectly, in a more threatening or softening way (which is directly related to politeness theory (Goffman, 1956)). The request can be emphasized, either projecting toward the speaker (Can I do X?) or the hearer (Can you do X?). A direct request may use an imperative, a performative, obligations and want or need

Table 1: Request Speech Act custom coding scheme.

Request Form	Linguistic form	Examples
Direct request	Imperative	Do x
	Performative	I am asking you to do x.
	Want or Need statements.	I need/want you to do x
	Obligation statements	You have to do x
Indirect request	Query questions about ability of the hearer to do X	Can you do x? Could you do x?
	Query questions about Willingness of the Hearer to do X	Would you like to do x?
	Statements about the willingness (desire) of the speaker	I would like if you can do X I would appreciate if you can do X

statements. The more socially brutal form being an order. An Indirect request may use query questions about ability, willingness, and capacity etc. of the hearer to do the action or use statements about the willingness (desire) of the speaker to see the hearer doing x. In our coding scheme, a grammatical utterance corresponds to only one speech act as in Table 1.

5 EVALUATION AND RESULTS

For the purpose of this study, we use a real world project in software development and apply our method on data. Data were collected after the end of the project that last nearly 2 years.

5.1 Project Description

A publishing group editing law-related codes (like insurance code, labor code) hired a software development company to create a workflow tool for their journalists. Due to geographical constraints, nearly all the communications during specification, implementation, tests and delivery were done through email. The development method was mixed agile, with weekly deliveries after initial analysis.

5.1.1 Project Team

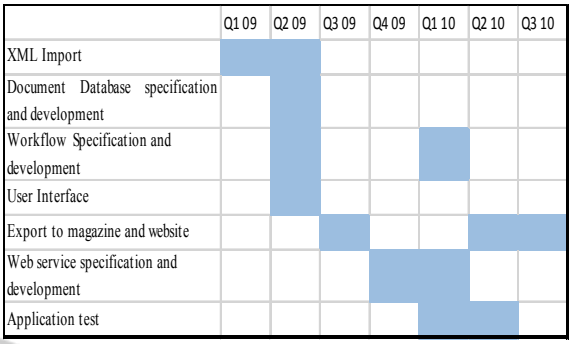
The team was split between the contractors and the development team. Among these, various roles and skills were present, but the main actors were:

- SRA: Chief editing manager (skill: law and management, Role: Contractor);
- JBJ: Information System Manager (Skill: Information system, Role: Contractor);
- FX: Information System Developer (Skill: Software Engineering, Role: Development manager);
- RT: Information System Developer (Skill: Software Engineering, Role: Sub-contractor

5.1.2 Project Schedule and Topics

The project start in early 2009 and ended in late 2010. The different phases are summed up in Table 2.

Table 2: Project main phases.



Based on project specifications and code deliveries, a small topics dictionary was built according to Section 4.3.2 and the message topic matrix was computed accordingly.

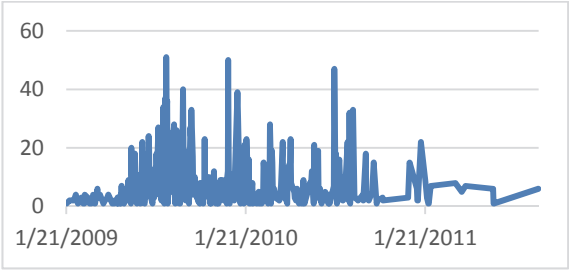
Table 3: Excerpt from Project Topic dictionary.

Topic	Keywords
XML	structuration, tag, tree, xsd, dtd, schema, markup, xml
BDD	Database, table, fields, editorial part, code part
Workflow	workflow, validation, collaboration
Code	Law, legifrance, insurance, chapter, article, annexes, labor code
User Interface	UI, login, user management, front end, search

5.2 Frequency

Our corpus represent 3000 messages in 800 threads between 30 projects actors. Size of message

Table 4: Daily message frequency.



We identify 10 main actors during this project that account for more than 80% of the messages. Also the daily frequency in Table 4 show 3 relevant spikes of activity matching critical time of the project: the first delivery and second delivery and a new features addition. We will reduce our investigation to the first spike between 06/2009 and

09/2009 where a lot of exchanges occurs and focus on long threads showing the presence of a dialog

As an additional information, it is to be noted that the Development Manager was the one receiving in TO (direct receiver) the higher number of messages, the Chief Editing Manager was the one sending message the most and the Information System Manager was the one receiving the most message in CC. This global numbers are matching their roles in the project, respectively executing, requesting and supervising.

5.3 Threads and Messages

At this point, we have narrowed our research for problem solving on some specific threads of interest, where the topics are known, the date and role of actors and discourse turn are fully qualified. We visualize interaction turn and highlight differences between to and cc receivers. An example of such thread can be viewed in Table 5.

Table 5: Thread context.

DATE	THREAD		SKILLS	Topics
			Law	
15/09/2009 17:25		CC	IS	book,xml
		TO	IS	
15/09/2009 17:34		TO	Law	xml
		CC	IS	
15/09/2009 17:37		TO	Law	
		TO	IS	
15/09/2009 17:52		TO	Law	code
		CC	IS	
15/09/2009 17:54		TO	Law	
		TO	IS	
16/09/2009 09:35		CC	IS	xml, layout export
		TO	IS	
A=Contractor Editor				
B=Contractor Information Sytem				
C=Developper				

This give us rich context to perform pragmatic analysis. To analyze message text and find problem solving elements, we used our pragmatics custom grid to identify request speech acts. Then, following the message thread, we look for answers related to same topic which emitter possess the appropriate role and skills.

For instance, in September 2009 in a thread related to topics XML and code, we found that a contractor editing role (SRA) is asking for a new xml markup in an indirect manner, one possible answer is given in the next message by the developer manager (FX) based on his role, competence and the topic. We also detect a false positive request by the contractor Information system manager (JBJ), but it will be discarded because not related to any topics from dictionary. Table 6 shows this example

Table 6: Messages analysis example.

From			Sentences	Topics	Problem Solving
SRA			If i could put a <juri> markup with all the needed content inside it in the books, i would be happier.	book,xml	indirect request
	to:	FX	Thanks,		
	cc:	JBJ	S.		
FX			Here you are, i added <link> markup also	xml	(answer?)
	to:	SRA	Sorry again for the bankers		
	cc:	JBJ	F.		
JBJ	to:	FX	some free development days would be welcome ;=)		indirect request
		SRA			
FX	to:	JBJ	It's planned :=) Don't worry 1 days/annexe = 2 extra free days on code	code	(answer?)
	cc:	SRA	customer satisfaction is .. our motto ;=) F.		
JBJ	to:	FX	What!!! I'm going to call all my other subcontractors! They owe us 400 days!		
		SRA			
SRA	to:	FX	thanks.	xml	
	cc:	JBJ	Last night, i thought about another markup that could be useful to me : a markup	xml	indirect request

6 CONCLUSIONS

The objective of our research is to discover problem-solving knowledge from computer mediated communication within a professional project. In this paper, we present a method using business emails for this aim. Email was considered as a specific asynchronous and not face to face kind of discourse and we applied some pragmatics analysis to it. Speech act analysis alone is a difficult problem, so we consider it inside the project context: human organization and skills, and project specifications. Our questions were: Does emails contain any useful trace of project related knowledge? And could we identify an appropriate method to extract and structure it?

The principle of our approach is to take into account all the aspects of the mediated exchanges during a project. Thus we have presented a general guideline to detect interesting (related to problem solving in software design) threads among a vast amount of messages. We then have proposed a custom coding scheme to detect request and clarify ambiguity in sentence analysis. To illustrate this, we have analyzed a professional project corpus and detect some parts of problems and possible answers. On the whole these results confirm the applicability of the technique, and we will further investigate on different types of project and other grid of analysis like: decision making, negotiations, and design rationale (Matta et al, 2011). It comfort us that when emails is not only seen as way to broadcast information, but as a media allowing online exchanges, like discourses, it could turn into a source of knowledge.

Our current work objective is to explore various techniques from machine learning to implement support algorithm for the projection of our features vectors (topics, pragmatics and context) to problem-solving knowledge model. Although related to the works of Cleland-Huang (Cleland-Huang et al, 2006) on Requirement traceability in software design, we will focused more on functional testing and design detection. Finally, this study is a part of our work on project memory: traceability and structuration of knowledge in daily work realization of project.

REFERENCES

Baron, N. S., 1998. Letters by phone or speech by other means: The linguistics of email. In *Language & Communication*, 18(2), p133-170.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Thomas, D., 2001. *Manifesto for agile software development*.

Bekhti, S., Matta, N., 2003. Project memory: an approach of modelling and reusing the context and the de design rationale.In *Proceedings of IJCAI'03 (International Joint of Conferences of Artificial Intelligence), Workshop on Knowledge Management and Organisational Memory*, Acapulco.

Benveniste E., 1966. *Problèmes de linguistique générale*, t. I, Paris, Gallimard.

Carvalho, V., Cohen, W., 2004. Learning to extract signature and reply lines from email. In *Proceedings of the Conference on Email and Anti-Spam, Palo Alto, CA*.

Carvalho, V.R., Cohen, W., 2006. Improving "email speech acts" analysis via n-gram selection. In *Proceedings of the HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech (ACTS '09). Association for Computational Linguistics,* Stroudsburg, PA, USA, p35-41.

Cleland-Huang, J.; Settimi, R.; Xuchang Zou; Solc, P., 2006. The Detection and Classification of Non-Functional Requirements with Application to Early Aspects. In *Requirements Engineering, 14th IEEE International Conference,* p39-48.

Cselle, G., Albrecht, K., Wattenhofer, R., 2007. BuzzTrack: topic detection and tracking in email. In *Proceedings of the 12th international conference on Intelligent user interfaces*, p190-197.

De Felice, R., Deane, P., 2012. Identifying speech acts in emails: Toward automated scoring of the TOEIC® email task. In *ETS Research Report No. RR-12-16,* Princeton.

Diesner, J., 2005. Exploration of Communication Networks from the Enron Email Corpus. In *Proceedings of Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining 2005*, p3-14.

Gray, P. H., 2001. A problem-solving perspective on knowledge management practice. *In Decision Support System 31, 1 (May 2001)*, p87-102.

Goffman, E., 1956. *The Presentation of Self in Everyday Life*, University of Edinburgh Social Sciences Research Centre Monographs, no2.

Gospodnetic, O., Hatcher, E., 2004. *Lucene in Action*. Manning Publications.

Hardin, L. E. 2002. Problem Solving Concepts and Theories. In *Journal of Veterinary Medical Education*, 30(3), p227-230.

Herring, S. C., 2004. An Approach to Researching Online Behavior. Designing for virtual communities in the service of learning. In *S. A. Barab, R. Kling, J. H. Gray (Eds.), designing for Virtual Communities in the*

Service of Learning (p338-376). New York: Cambridge University Press.

Jindal, N., Liu, B., 2007. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web*, p1189-1190.

Kalia, A., Motahari Nezhad, H. R., Bartolini, C., Singh, M., 2013. Identifying Business Tasks and Commitments from Email and Chat Conversations. In *HP Labs Technical Report*.

Ladas, C., 2009. *Scrumban-essays on kanban systems for lean software development*. Lulu.Com

Malvache, P., Prieur, P., 1993. Mastering Corporate Experience with the REX Method. In *Proceedings of ISMICK'93, International Synopsium on Management of industrial and corporate knowledge*, Compiegne.

Matta, N., Ribiere, M., Corby, O., Lewkowicz, M., Zacklad, M., 2000. Project memory in design. In *Rajkumar Roy, (Ed.), Industrial Knowledge Management – A Micro Level Approach*, Springer-Verlag, London.

Matta N., Ermine, J-L., Aubertin, G., Trivin, J., 2002. Knowledge Capitalization with a knowledge engineering approach: the MASK method. In *Knowledge Management and Organizational Memories*, *Dieng-Kuntz R., Matta N. (Eds.),* Kluwer Academic Publishers.

Matta, N., Atifi, H., Sediri, M., Sagdal, M., 2010. Analysis of Interactions on Coordination for Design Projects. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2010 Sixth International Conference* , p344,347.

Matta, N., Ducellier, G, Charlot, Y., Beldjoudi, M.R., Tribouillois, F., Hibon, E., 2011. Traceability of design project knowledge using PLM. *In IEEE proceedings of International Conference on Cooperation Technologies and sciences*, Philadelphia, May 2011.

Newell, A., Simon, H. A., 1972. *Human Problem Solving*. New Jersey: Prentice-Hall, Inc.

Searle, J. R., 1969. *Speech acts: An essay in the philosophy of language* (Vol. 626). Cambridge university press.

Stiles, W. B., 1992. *Describing Talk: a taxonomy of verbal response modes*. SAGE Series in Interpersonal Communication. SAGE Publications. ISBN: 0803944659.

Tourtier, P.A., 1995. Analyse préliminaire des métiers et de leurs interactions. In *Rapport intermédiaire, projet GENIE, INRIA-Dassault-Aviation*.

Wasiak, J., Hicks, B., Newnes, L., Loftus, C., Dong, H., Burrow, L., 2011. *Managing by E-Mail: What E-mail Can Do for Engineering Project Carley Management*. In IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 58, NO. 3.