

MASTER THESIS

**Semantic Enterprise
Collaboration Support (SECS)
with instant Awareness Stimulation
for Knowledge Sharing**

Academic Institute	School of Business University of Applied Science of Northwestern Switzerland
Student	Brayan Zimmerli, MSc Business Information Systems {firstname.lastname}@students.fhnw.ch
Advisory Committee	Prof. Dr. Knut Hinkelmann, Dean of MSc BIS Programme Prof. Dr. Holger Wache, Lecturer of MSc BIS Programme {firstname.lastname}@fhnw.ch
Thesis Submission	30/07/2010

Declaration

I declare that this thesis presents work carried out by myself and does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and to the best of my knowledge it does not contain any materials previously published or written by another person except where due reference is made in the text; and all substantive contributions by others to the work presented, including jointly authored publications, is clearly acknowledged.

MSc Candidate Signature

Brayan Zimmerli

Student ID 06-487-474

July 30, 2010

Acknowledgement

My sincere appreciations go to my open-minded and encouraging supervisors, in particular to Prof. Dr. Knut Hinkelmann, Prof. Dr. Holger Wache and Prof. Dr. Rainer Telesko for their vital dedication lasting for the last five years of Bachelor and Master Studies. Moreover, I am grateful for this excellent academic opportunity that I was allowed to attend at the University of Applied Science of Northwestern Switzerland, supported by the state and its citizens.

A cordial thank-you I send to Muriel Weber and Janine Lüscher for their inspiration and backup.

Last and most important; a warm thank-you to my father for his unstrained encouraging attitude towards education; for balancing my bank account, allowing room to intensify my studies and research. Without you, this had been barely achievable. Thank you very much indeed.

Abstract

Knowledge-intensive work consists of the reception, interpretation, and structuring of information; the explicit articulation of mostly unstructured data in suitable representations. Further it comprises the sharing, exchange, and discussion of such resources with other people. The way people proceed is significantly influenced by the tools used – the personal workspace support – and reflect the individual ways of information processing and knowledge sharing across organizational networks. This research paper illustrates an approach to foster collaboration and knowledge transfer by applying semantic technology. The main focus is about raising awareness for existing knowledge, or information that could lead to knowledge when addressing the right context and the right information worker. An exploratory prototype shall offer joint capability to improve his/her work performance by *raising relevant information* and – if applicable – *at the right time*.

Keywords: Semantic, Collaboration, Awareness, Ontology, SECS

Table of Contents

DECLARATION	I
ACKNOWLEDGEMENT	III
ABSTRACT	V
I. INTRODUCTION	1
A. Background Information on Motivation and Drivers	1
1. The Information Bulk and its Value	1
2. Requirements for Competitiveness.....	4
3. Collaboration – Adding Collective Intelligence	7
4. The Effect on the Enterprise Environment.....	9
5. Stimulation of Information/Knowledge Sharing.....	10
6. Short Summary on the Background Information	11
B. Problem Statement	11
1. Problems in Demand Identification and Formalization	11
2. Problems in Knowledge Assembling	12
3. Problems in Establishing Trust & Accuracy.....	13
4. Problems in Knowledge Delivery	13
C. Research Objective & Thesis Statement	13
D. Delineation, Limitations, Assumptions	14
1. Delineations.....	14
2. Underlying Assumptions	15
E. Summary	15
F. Overview of Sections and Chapters	15
II. LITERATURE REVIEW & RELATED WORK	18
A. Context & Demand Formalization	18
1. Context and Context Modelling.....	18
2. Context-Awareness	26
3. Context Sensors - Making Sense of Context	29
4. User Demand Sensing and Prediction.....	34
5. Fundamental Components – Ontologies and System Architectures.....	35
B. Knowledge Assembling	37
1. Knowledge Creation and Process-Integrated Deficiency Management.....	37
2. Sources for Knowledge Creation	40
C. Establishing Trust & Accuracy	40
D. Information and Knowledge Delivery	41
E. Summary: The Research-GAP & the Significance of the Study	42
III. RESEARCH DESIGN	43
A. Research Method Selection and Justification	43
B. Research Strategy – The Case Study Applied	44
C. Methodological Limitation	45
D. Summary	45
IV. SEMANTIC INFORMATION AWARENESS STIMULATION	46
A. Case Analysis	46
1. Situation Analysis.....	46
2. Tentative Hypothesis.....	49
3. Requirements Engineering.....	50
B. Architecture and Implementation	51
1. System Landscape and Deployment Diagram	51
2. Context & Demand Formalization	54
3. Knowledge Assembling and Awareness Support	58

4.	Trust & Accuracy-----	62 -
C.	Assessment-----	63 -
D.	Summary-----	65 -
V.	FINDINGS, CONCLUSION & OUTLOOK-----	66 -
APPENDIX A – EXAMPLES OF THEORY AND IMPLEMENTATION DOCUMENTATION 1		
A.	Examples of Context Description-----	1
1.	UAProf Example-----	1
B.	Use Cases Submitted by Alpiq AG-----	3
C.	Assessment Documentation-----	4
D.	SECS Desktop Documentation and Resources-----	4
1.	Installation Hints-----	4
2.	Resources-----	4
APPENDIX B – REFERENCES----- 5		
A.	Literature References-----	5
B.	Figure References-----	13
C.	Table References-----	14
D.	Glossary-----	14
APPENDIX C – NOTICE----- 16		
A.	Legal Notice-----	16
1.	Liability-----	16
2.	Copyright-----	16
3.	Law for trade market protection-----	16

I. Introduction

A. Background Information on Motivation and Drivers

Throughout the history of man-kind, information was at all times crucial to the positive development of organizations – and so to every individual worker, who claims the processing of information his/her main tasks.

Before the end of the millennium systems primarily focused on the **dispersion of information** per se, and the World Wide Web still operates in a broadcast-mode. It is not selective on the information consumer's identity and demands. However, this is (and probably continues to be) its intention.

For other environments – predominantly in business – rather a multicast-mode is applicable. Meaning that, each information consumer is supplied with documents from a central source as for example from a Corporate Content Management and File Sharing Systems. Going one step further, users might have a one-to-one information sharing relationships. In this case one speaks about a unicast-mode of information transaction. Thus, these days' systems are required to return filtered results. This for *two obvious reasons* [1]: the heap of available information and the number of contributors turns to an astronomic size [2-3]. Secondly, there is an amplified challenge to act effective and efficient [3], which is driven by global competition. Briefly speaking, in knowledge intensive labour the first reason asks for *relevance*, the second for *competitiveness* [4-5].

1. The Information Bulk and its Value

a) *The Bulk of Information and Artefacts*

The former reason of the untamed bulk of dispersed information artefacts is mainly a characteristic of the world-wide-web. As Tim Berner-Lee stated [6], information is to deliberate from organizational silos and hierarchy – and it happened. This non-reversible trend of information-sharing added new attributes [7] to data artefacts, which in integrally closed areas were not of concern up to the start of the internet revolution. Why do we need to consider new attributes, like the Dublin Core [7], to spread information? It is for the reason that, in the first instance, the opening of foreign sources is about *collecting raw data* of heterogeneously structured sources. Data which must be described in order to transform it into information. More eloquently explained, having data pieces identified with appropriate terms, simplifies this transformation process, because applying a well known syntax and a common understanding, decreases the interpretation range of the data acquired, as researchers from the University of Heidelberg, among many others, discussed in a blog [8]. There is little to no doubt in research that this first step must be conducted to consume (parts of) large-scale unstructured data sources like the World Wide Web. In this described *data-state*, a user or a system may start to categorize, contextualize, condense or aggregate the collected data to transform it to an *information-state*.

The Semantic Business Vocabulary and Business Rules (SBVR) [9] and the XML Meta Data Interchange (XMI) [10] were upraised as first international standards de jure [11], to give an example, which is used to

define and exchange concepts and instances (of concepts) between organizations and their systems. This syntactic and self describing standard, speaking of XMI, enabled data providers and consumers to exchange machine-readable information with the benefits of reducing the risk of misinterpretation of information.

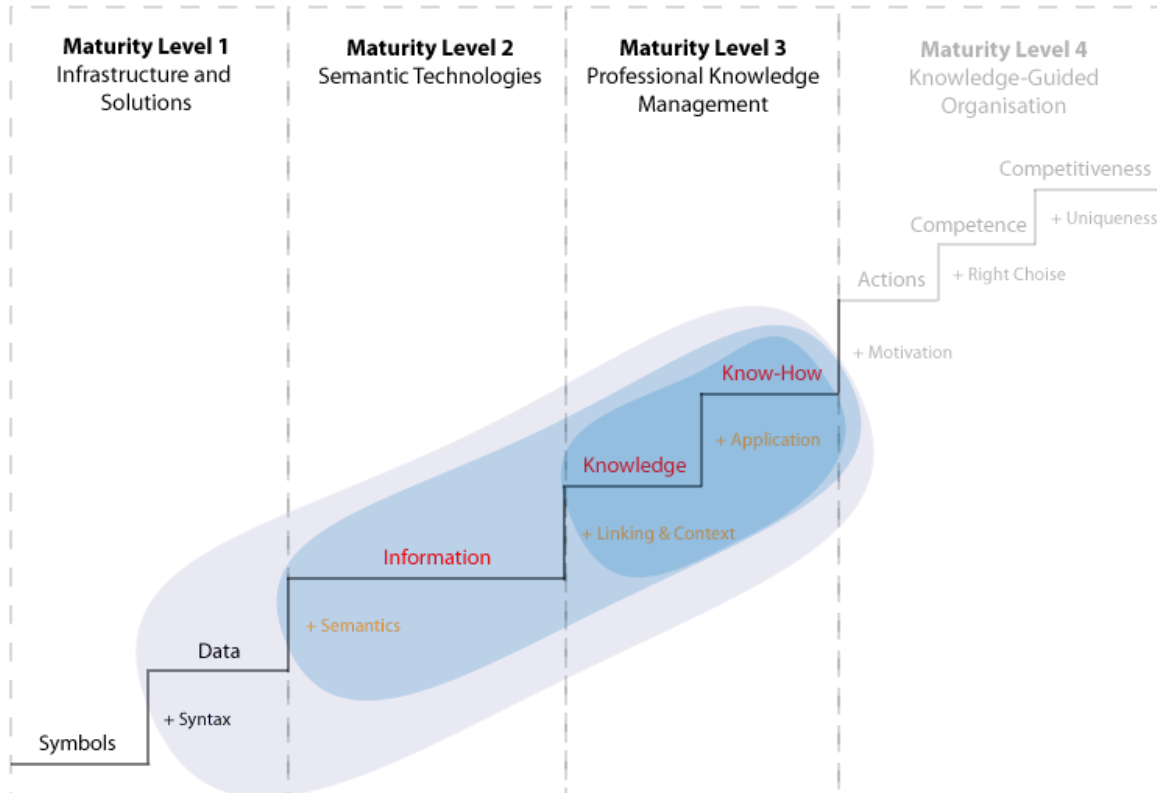


Figure 1- The Maturity Evolution within Knowledge-Driven Organizations and the Focus of this Project, © Klaus North, Zimmerli

All the way through this document we will learn that other information management concepts alike SBVR offer higher potentials in the process of information interpretation. It is the first step in indentifying relevant artefacts by adding meaning to data. “The study of science of meaning in language” [12] is what one calls *Semantics*. In other words, we speak about semantics when “meaning” is added to syntax or merged in any possible way that leads data to information as shown in the knowledge maturity development of Figure 1 by Klaus North [13]. This master thesis will propose a solution that “fertilizes” the Maturity Levels 1 to 3. Level 4 is out of scope because it concentrates topics of human resource, organizational management, business analytics and business strategy.

b) The Bulk of (Re)Sources

The bulk of information is produced by users and systems referred to as resources. One can also summarize both under the term producers. A producer researches and creates data as shown in Figure 2. At this point, the following questions appear in a conventional information management process:

- 1) What consumer requires what information,
- 2) what data do we produce to satisfy the information “appetite“ of the consumer,
- 3) how do we format (describe) it for the consumers and
- 4) how do we inform the consumer?

Introduction

Answering these questions will likely lead to an transformation from data to information as described in the previous chapter, because the producers tends to routinely customize the data and put it in context. The context is considered the “field of application” of the consumer. This implies that the producers believe to know his/her consumers and their demands for data and the composed artefacts needed. This sometimes hasty act of information preparation is appropriate within a scenario where producers know their consumers and where numbers of consumers are finite. But this is not pertinent against the background of a bulk of producers and consumers where the number is unlimited or at least arbitrary high.

Service Oriented Architecture (SOA) and Peer-to-Peer (P2P) represent fairly well established concepts which tackled this downside - referring the prerequisite of knowing the number of consumers and their information demands. Both concepts foster the common principle of sharing atomic data or information pieces among undefined growing numbers and heterogeneous types of consumers – and most important – they describe their data without putting them in a constricted context. Again - these concepts pay tribute to the fact that the number of producers and consumer are quickly growing. They both help to avoid the tedious negotiation between producer and consumer in the process of information supply. The discontinuation of mandatory negotiation keeps a producer and consumer highly agile.

However, this principle entails a new condition. The producer and consumer groups must build on well known standards like HTTP, XML, XMI, WSDL, etc., in order to exchange information and avoid the negotiation. With XML one may describe data and add semantics to it. Web Service Description Language (WSDL), for example, is based on XML and assists in describing atomic data in exchange activities of producers and consumer. It clearly fosters the self-service attitude.

But why do we speak so eloquently about these new concepts in the introduction? Because it illuminates the fact that industry tends to intensify the description of data in order to handle the growth of producers and consumers. One could also say, *we describe data for the single advantage of not losing its semantics* in whatever system, stage or scenario one might process it.

Nevertheless, the preparation of data did not end in a described information-state. It (often) also transferred from information-state to relational-state. A *relational-state* [14] is considered linked information in any imaginable structure.

This cross-linking of information is conducted on the producer’s side within the isolated approach. In particular the concept of SOA disposes this approach and tends to leave it to the consumer to assemble the information to a more valuable state. As illustrated with maturity level 2 in Figure 2, this *relational-state* fundamentally assists us to extract knowledge from the untamed bulk of resources.

“The problem is not just too much information; it’s too much bad information. The information is delivered unpredictably. It comes from every direction in unimagined forms,”

Whit Andrews
Research Vice President Gartner,
2007

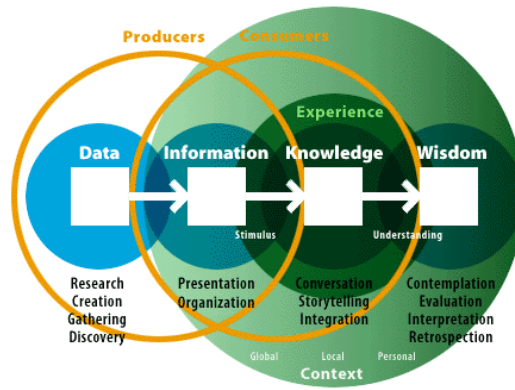


Figure 2 - The Transformation of Data to Wisdom, © Scott A. Carpenter

2. Requirements for Competitiveness

Already in 2002 [15] organization struggled with the sheer bulk of information. It goes without saying, that in 2007 [16] and later on the information flood didn't deplete and the issues from the previous chapter remain critical. As in 2002 where the majority of 90% companies believed that their competitiveness is negatively impacted as a result of information overload, this remains a hot topic for executive officers. The risk of hampering one's competitiveness aggravates since information reaches the organization, not just in high quantity, but also from all kind of unverified sources, as Whit Andrews from Gartner states [16].

a) Effectiveness, the Provenance of Source and Context-Awareness

No organization that is subject to this information excrecence could afford not to react and apply solutions with maximum grip; grip in terms of an effective way to manage the information flood, respectively an effective way to deliver information to users. This leads information workers to questions the *trust* of sources – and not just its producers' (trust) but also the quality (proof) of the information. Because effectiveness in knowledge delivery I.A.2.c) cannot be achieved with questionable information. Therefore, the *provenance of sources* [17] is essential in the data and information selection process and significantly improves the competitiveness in long terms. On the other hand, decision making upon untrustworthy and unverified information, may results in doing the wrong things. As a consequence the superior organizational or individual objectives cannot be met.

b) Efficiency and the Inference of information

While one makes decisions, he/she is thoughtful to do it the right way, because “the resources expended in relation to the accuracy and completeness of goals achieved” [18-19] determines if his/her efforts are efficient and thus value-adding to the organization's mission.

Both transformation steps, from the data-state to the information-state and further to the relational-state respectively knowledge-state, are aimed to be supported by appropriate technology application. This is done to accelerate the **knowledge engineering** by relieving the user from manual and time-consuming information and **knowledge-assembling tasks** that face intractable data volumes. These supportive systems let users focus [20] their cognitive energy on upper level tasks of the knowledge pyramid as illustrated in Figure 3. In a cognitive layer a knowledge worker is engaged with demotic statements that are non-representational statements.

Introduction

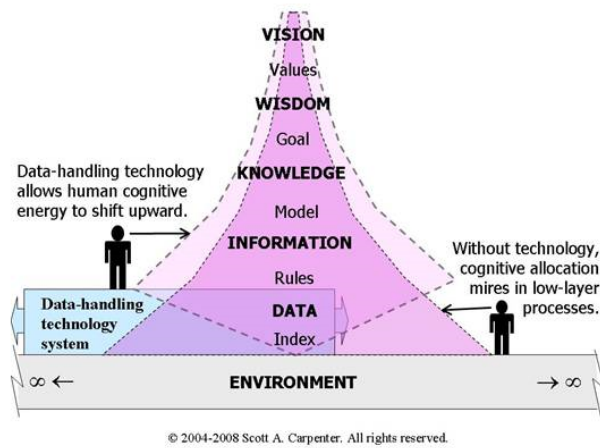


Figure 3 - Knowledge Pyramid and Cognitive Energy Application, © Scott A. Carpenter

While data-processing, e.g. indexing, represent approaches up to the information-state, semantic technology pursues to relieve the user task that lead to the knowledge-state. Scott A. Carpenter represented the allocated cognitive concentration in a quasi-diamond in Figure 4. Several researchers [21-22] argue that computing technology amplifies human cognition of valuable knowledge. It leaves cognitive resources to up-shift in the knowledge pyramid. One could guess that bringing semantics into place would make knowledge as mundane as data and information are today. This implies that users (and organizations) could allocate their energy towards their very goals and the creation of true unvarnished value [20].

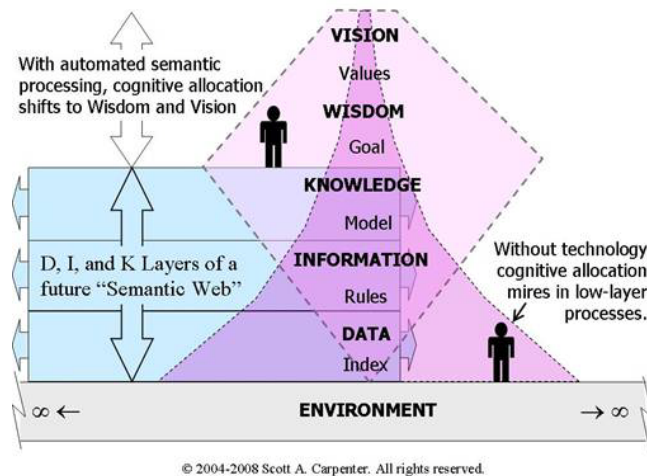


Figure 4 - Knowledge Pyramid and the Up-Shifted Energy of Cognition, © Scott A. Carpenter

While speaking about knowledge, let's set a definition we use to work with throughout this paper from the Oxford Dictionary: "(ii) what is known in a particular field or in total; facts and information" [12]. Facts refer to a true proposition one either knows it is true without any doubt or it is true under certain conditions within a "particular field" and/or scenario. We will use the term **domain** as a synonym of "particular field" and the term **context** - which is depicted in length later in this chapter - instead of scenario. In a domain we describe the content of information sources, as mentioned in previous chapters, and not the (re)source itself. For instance, "parties to a contract" or "articles of agreements in a contract". Since not any description is suitable for machine-readable information processing, one needs to decide preferably on

a single formal representation. Thinking of a system that supplies relevant information to a user, one can imagine that it must be capable to formally represent not only the knowledge available but also the knowledge demanded by the user [23]. Knowledge Engineering comprises the deduction from information elements to knowledge, thus facts.

Efficient knowledge engineering embodies all these capabilities: transforming data to information, a common and compliant representation of information, and a sophisticated knowledge base that sprouts form knowledge gathering and inference.

c) Contextual Application of Knowledge in aid of Competitive Advantage

The success promising attribute “competitiveness” comes with a prerequisite: One needs to apply knowledge in favour of business needs [2, 4]. Giving way to sophisticated systems will privilege an individual or group to leverage maturity level 3 of North Klaus [13], apply knowledge and excel their know-how. In a survey of The Economist Intelligence Unit in 2005 [2] in Western Europe in respect to managing knowledge for competitive advantage, 56% of the 122 senior executives judged the experiential knowledge of their managers and employees as ineffective to moderate. Asking the question about the impediments of the deployment of knowledge-fostering tools, “the cost of deployment” (1a) ranks top by 51% of consent, followed by 49% of lack of awareness of what is possible (1b). Moreover, only 16% of the executives do not classify knowledge management (business intelligence) as a priority. Why do they not intensify knowledge-sharing? Because information is not adequately prioritised (2a) say 55%, and 43% confirm that the information is not always accurate (2b) or reliable (2c). Further, 39% complain about having too much information available (2d) and 23% even complain about slow information processing (2e).

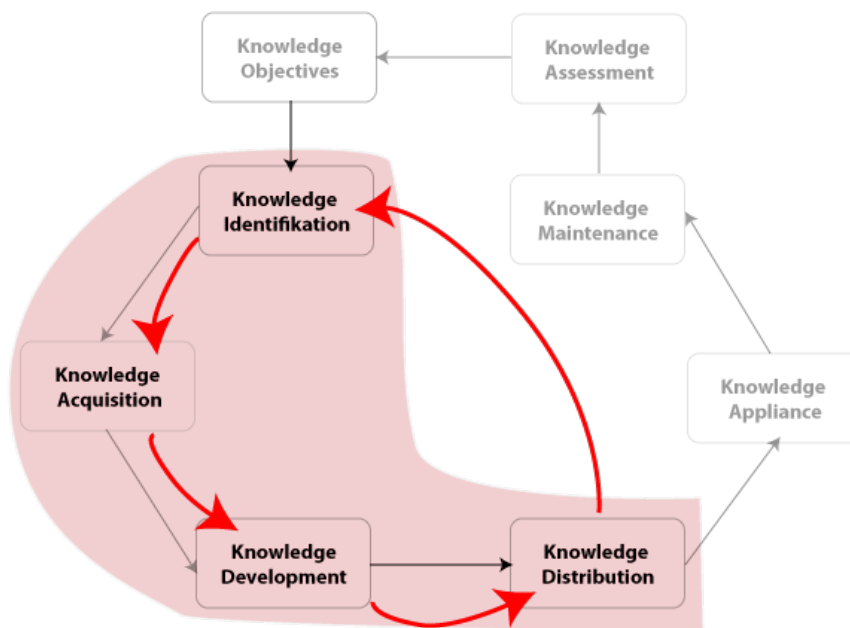


Figure 5 - The Compact Chain of Knowledge Mgmt (CCKM) derived from the Blocks of Knowledge Management, © Probst et al., Zimmerli

From Probst et al. [24] in Figure 5 we learned about the tasks from knowledge management. Four of them are at focus in this paper.

From the references in this introduction, we derive a compact *chain of major business-supportive technical tasks in knowledge management* to emphasize on, in order to pay tribute to the figures from The Economist [2] and the statement from Gartner [16] (3a, 3b). With these facts in mind, we argue and integrate the summarized elements of ensuring provenance, trust and accuracy in a new chain of knowledge management – as some other researchers [25] did not. For the sake of remembering this throughout the document we name it Compact Chain of Knowledge Management (CCKM). This compact chain also highlights the focus of this thesis.

1. Identify (1b) the knowledge required.
2. Acquire, extract (2d, 3a) relevant information and engineer knowledge from the bulk or resources.
3. Ensure trustworthiness (2c) and accuracy (2a, 3b) of information.
4. Deliver (1a) the relevant knowledge on time (2e) when needed.

With the term and phrase “relevant“ (2d, 3a) and “on time when needed“ (2e), we address context. The concept of *context* is absolutely central to this thesis, because the term covers “where you are, who you are with, and what resources are nearby” as Schilit et al. states[26].

In this given context another necessity obtrudes. How do we make stakeholders aware of their needed assistance, if they don't know about the other activities and states of work? This is a matter of (context-)awareness support or, as Dourish and Belotti say, “awareness is an understanding of the activities of others, which provides a context for your own activity” [27]. The purpose of *awareness support* is to reduce mutual uncertainty and increase knowledge among distributed co-workers [28] and is, however, not limited to co-worker-awareness.

In the next chapter we clarify the role and magnitude of collaboration within the chain of business-supportive technical tasks, context and context awareness support.

3. Collaboration – Adding Collective Intelligence

By its etymological semantics, collaboration means (the process of) working together [12]; lat. co = with, laborare = work. Collaboration is often understood as an explicit and well defined mode of working together. Ellis et al. [29] disagree and divide Computer Supported Cooperative Work (CSCW) into Workflow Management (WfM) and Workgroup Computing (WgC). While the former describes a strictly process-guided work the latter curtails the scope for very individual and unstructured activities loosely-coupled to a process [23, 30]. No doubt, semantic technology can be applied in both areas, but WgC and semantics offers higher potential due to their matching characteristics of demand and support: individuality, diverse and variable structures, and ad-hoc behaviour due to non-defined activities. As the title of this thesis indicates, we intend to *quicken knowledge sharing by stimulating awareness* and – more precisely – we want to do so for *WgC platforms*.

Collaboration, however, can be aspectised in four social interaction classes [31] one needs to understand to follow the path to awareness stimulation for knowledge: Co-Existence, Communication, Coordination and Cooperation.

a) Co-existence and Communication – the Fundamentals and the Glue

Interaction of Co-Existence basically denotes the fundamental purpose of awareness about elements in a collaborative network. Elements are users, systems or artefacts that are of interest. Once identified, co-existence also covers the term co-presence, since an element's availability might change.

The identified chain of tasks from the previous chapter could well be applied to formal processes. But as Kraut et al. [32] prescinds “Without informal communication, many collaborations would undoubtedly not occur and others would break up before becoming successful.” in regards to *communication*, collaboration often comes with this *very informal touch*. From this statement we can conscientiously derive, that if collaboration opportunities are created by accident, then there must be collaboration opportunities missed. In the words of this thesis, we neglect the facts that other interesting sources co-exist. If we think of the collaboration matrix in Figure 6, these opportunities are unconsciously ignored in a working environment with either remote places or asynchronous working hours - or in worst scenario, with both cases present. The Web 2.0 offers many techniques like blogs and wikis to overcome the spatial and temporal constraints of ignoring collaboration opportunities as in Figure 6.

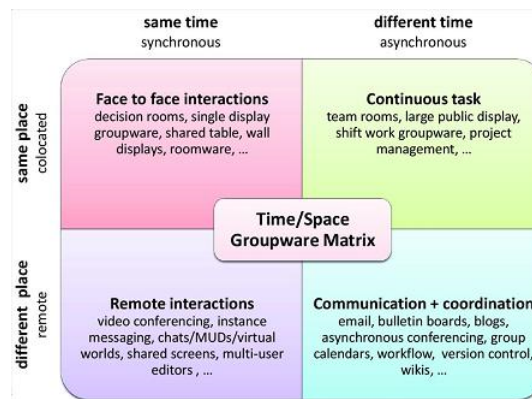


Figure 6 - The Collaboration Matrix Divided by Spatial and Temporal Attributes

In a nutshell, the awareness of co-existence is the prerequisite [28] and communication is the glue of any form of social interaction in a successful collaboration.

b) Coordination, Consensus-building and Cooperation for value-adding

“Coordination is the activity of directing individuals’ effort towards achieving common and explicitly recognized goals.” states Blau & Scott already in 1962. I. e. an organization’s elements communicate, coordinate, find consensus and cooperate in favour of a collective mission.

The consensus-building is vital of a reason we heard before. Individuals are opinionated and add different weighting, severity, semantics to elements e.g. working tasks, and groom different perceptions in general. The same is correct for proprietary systems. One understands that this causes ambiguity with any network – human or machine. Collaboration, coordination and consensus-building help to partially disambiguate individual perceptions and *assists to realize the alignment of knowledge-management with business-strategy* [4, 33]. Alignment means an organization sets a knowledge-management strategy to overcome the business challenges it faces. The observation of an organization’s characteristics in the four aspects of collaboration is conducted for the benefit of a solid knowledge management chain (CCKM) depicted in chapter I.A.2.c).

In what sense and magnitude do these four CSCW aspects matter in the CCKM? We argue, that the chain is generalised and therefore universally true to any knowledge management strategy [34] applied. Some KM strategies rely on *autonomous personification*, some on *synergetic codification* and others combine the two approaches. Thus, and looking at Figure 7, enriching the CCKM with the four CSCW aspects, let's us imagine a *collaborative KM* [35-36] is not just system-oriented but also can incorporate human interactions. It is no longer a simple central knowledge base - that is a container of knowledge - it is a network of personal and organizational knowledge bases. This carries on the advantages of personification, personal knowledge and contribution that for some reason is or cannot be systematically "captured" and externalized. On the other hand, it also leverages existing, explicit knowledge.

As a summary we can say, the combination of machine-gearred information acquisition and community-based knowledge organization leads to a situation where we claim to have the prerequisite to collect sufficient knowledge or sufficient information to derive knowledge.

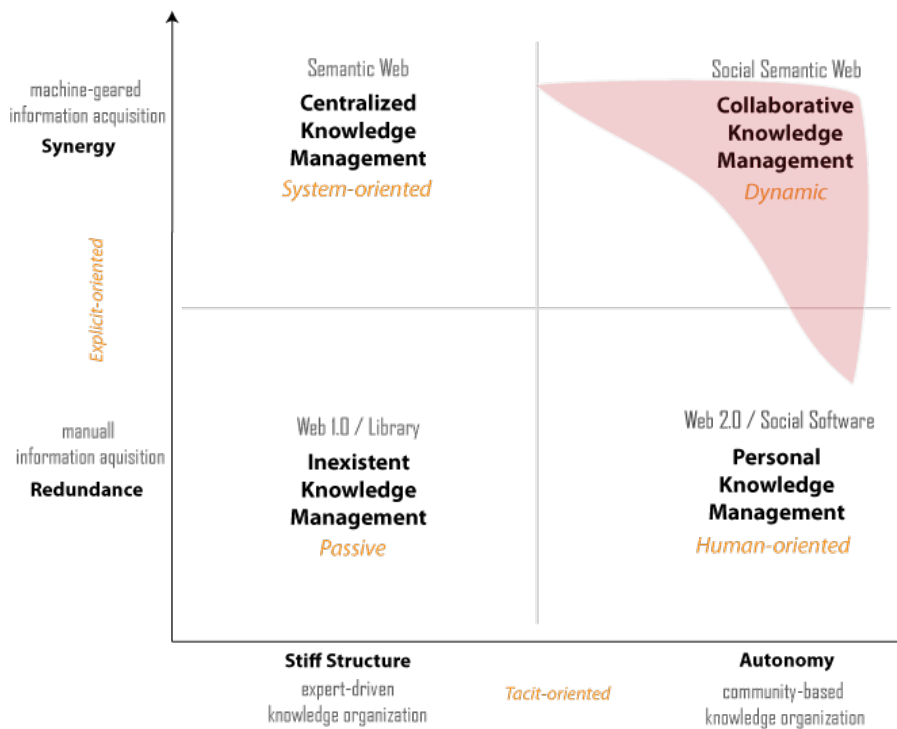


Figure 7 - On the way to Enterprise Collaborative Knowledge Management, © Schmit et al., Choi et al., Zimmerli

This paper's focus, however, resides in the red splash in Figure 7. The paper's body and conclusion will propose how to avoid information redundancy, and stiff structured elements within a collaborative enterprise. These entrepreneurial characteristics are explained in the next chapter.

4. The Effect on the Enterprise Environment

An enterprise organization delimited and extended itself in various aspects to community-based knowledge organizations in the past. But as many authors [37-39] of the IT industry and institutes mention, it will shift towards human-oriented and community-driven, and bring down silos and boundaries that hampered knowledge exchange among co-workers acting in the same context. Community-driven means to share information and knowledge among people with the same interest,

regardless of their departmental, hierarchical or functional position. A distinctive attribute of this new approach used, is the *autonomous action* of its users; autonomous in the sense of acting [39] informal, spontaneous and non-structured, without being patronized by any central instance of an enterprise e.g. business process manager, workflow-management server and business rules server. In contrast, vintage enterprises stick to hierarchy that is inflexible in acquiring information sources within and outside the organization, missed knowledge sharing opportunities and, therefore, lack of competitive advantage [2, 25] in fast moving markets.

Senior Researcher Andrew McAfee sees the Social Semantic Web as a train to deliver dynamic Collaborative Knowledge Management to the enterprise [39]. On the other hand, a 2009 survey of the Enterprise 2.0 Conference organization pinpoints the obstacles: 52.3% senior managers see the risk of employees' resistance, 42.1% find it difficult to measure ROI and the next runner up is the *concern about system integration*, 40.6%. While researchers promote this shift to high transparency, system engineers are confronted to present solutions to process all the open information so that all the achievements in *machine-gearred information retrieval systems* can prevail – without violating user privacy and enterprise security regulations. Approximately one third of surveyed population see security as an issue.

5. Stimulation of Information/Knowledge Sharing

As previously written, awareness-support in collaborative enterprises is a must-have. McAfee provided a list of six functionalities [39] required for Enterprise 2.0: Search, Links, Authoring, Tagging, Extensions and Signals. They are summarized under the term SLATES and can be used as checklist for enterprise 2.0 solution integration. Dion Hinchcliffe makes some solid remarks on SLATES and adds four more points [40]. The Emergence, one of the four points, emphasizes the critical aspects of Signals from SLATES. While McAfee mentions how users can be notified by Email or RSS (Signals), he gives a back seat to the problem that emergent content volume raises: The sheer volume of the social web hype can impede an information worker's performance, argues Hinchcliffe. So, his question combined with our topics is: How can we appropriately stimulate awareness-support to overcome this risk?

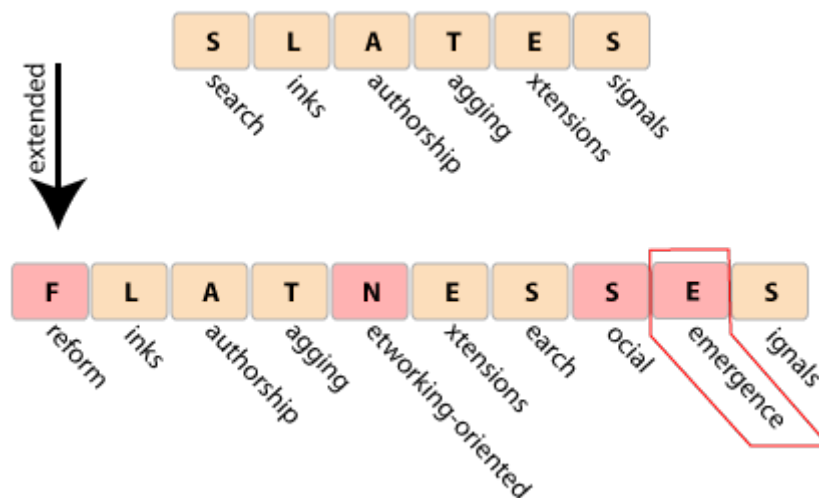


Figure 8 - From SLATES to fLATnESseS, © McAfee, © Dion Hinchcliffe

6. Short Summary on the Background Information

All elements elicited from the scenario of information workers (IW) require a representation to make it processable (machine-gearable) by computers. Otherwise computer systems and computer networks may not apply their filters in an effective way in order to return more precise results from the bulk of information resources. Furthermore, IWs collaborate intensely with other IWs. Some are more pertinent, some less. In this Master Thesis, semantic computer supported collaboration is explained to securely and actively push and pull valuable information to the surface and – also – at the right time. A short latency shall guarantee awareness support within dyads and larger groups within the enterprise.

In collaboration research one finds little about instant awareness support for information which is relevant to a specific information worker or group. One reason might be the various challenges and problems which are aggregated with the terms semantics, enterprise, collaboration and awareness. Nevertheless, at this point of the paper we have a clear understanding of what elements are key. In the next chapter we will locate the problem areas and its interdependencies.

B. Problem Statement

We want to use our CCKM as a thread to make sure we do not neglect any problem on the way to a semantic enterprise collaborative system with awareness stimulation; because we argue that the real challenges remain in the elements of the chain.

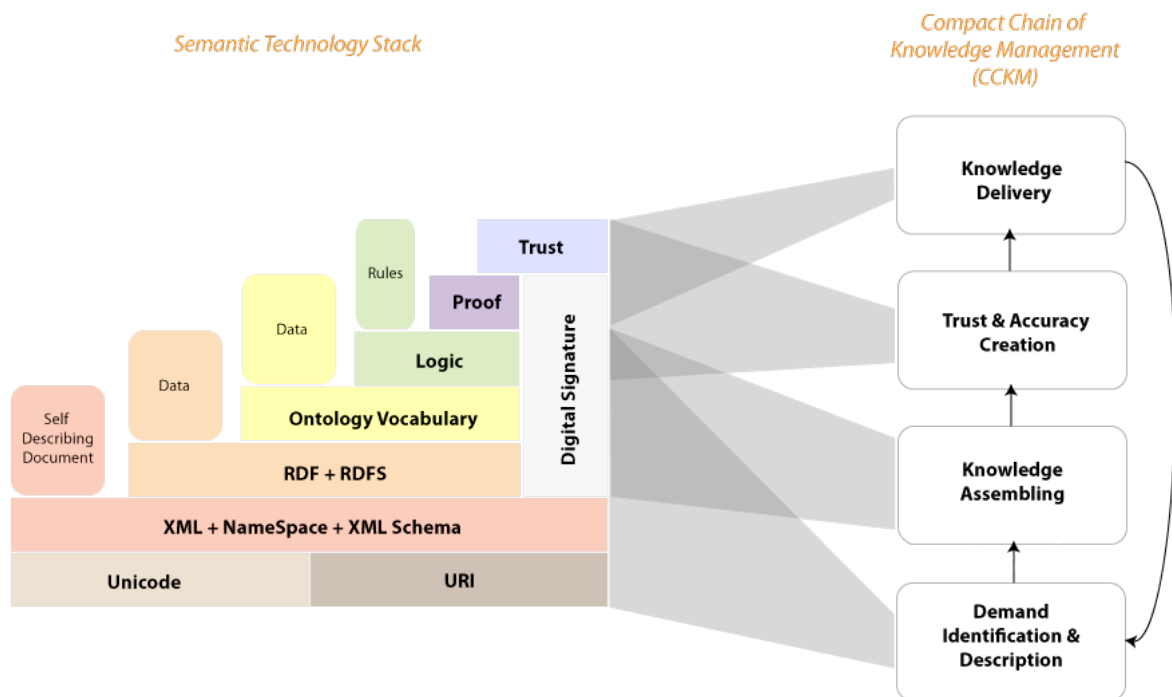


Figure 9 - Locating Area of Appliance of the CCKM, © W3C, © Zimmerli

1. Problems in Demand Identification and Formalization

We need to bear up the machine-gearable approach when we want to manage knowledge engineering efficiently – as we infer from reading about efficiency and competitiveness in chapter I.A. From this we derive the requirement to *formalize the knowledge* demand of our user. This research is not on process-

guided semantic collaboration. It is important to distinguish, because *strictly structured processes* with workflow definitions have dissimilar properties in regards to information allocation, compared to autonomous and loosely-coupled processes.

Hence, we need to accept that we have no straight-forward approach to describe the knowledge demand, nor to identify the context the user is engaged. That leads to the first questions within the first element of the CCKM.

- How do we *describe context*?
 - o What is context?
 - o How do we formalize the demand to be computable? Respectively, what does the *context and demand-representation* look like?
- How do we *create context*?
 - o How do we deduce context from satisfactory and consistent signals?
 - o How do we deduce context from unsatisfactory and inconsistent signals?
- How do we determine *personal knowledge demand*?
 - o What signals do we listen to, to identify the work context of the user and identify the knowledge demand?

2. Problems in Knowledge Assembling

Knowledge Engineering mainly is about acquiring, extracting and building a knowledge base. Knowledge bases can also be distributed. This is where personal and centralized instances compete with each other. That returns us some well known conflicts of interest. However, before we need to answer questions of fundamental matter derived from Figure 10:

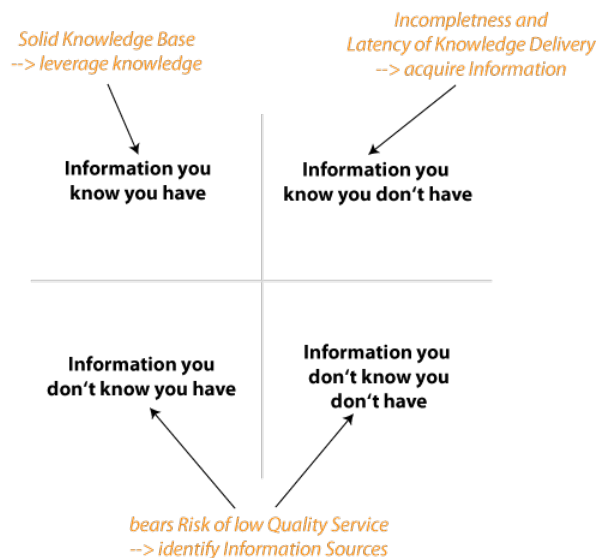


Figure 10 - Four states of information possession, © Hari Srinivas

- Which (re)sources do we select and acquire?
 - o Do we have clear directives of *what* we need supply and to *whom*?
 - o Do we have sufficient resources to build and derive knowledge? Do we discover or derive knowledge, or both? Do we *interpolate* knowledge?
- How do we *create* knowledge?
 - o *Inaccuracy*: How do we deal with falsely derived facts, resp. knowledge inaccuracy or inconsistency [41-42]? E.g. Contract A is archived, vs., Contract A is abandoned?
 - o *Vastness*: How can we avoid not be drag down by ~~maintenance~~ [43] (blocking cognitive energy of the co-workers)

- *Latency*: How can we avoid high latency while building knowledge [43]?
- *Uncertainty*: How do we deal with uncertainty resp. incomplete knowledge?
- How do we *combine* difference information sources and knowledge bases?
 - *Ambiguity*: How do we deal with conflicting assertions? How do we disambiguate [42]? E.g. "Contract party is in a solid financial situation." Vs. "Contract party is in bad financial situation."

3. Problems in Establishing Trust & Accuracy

Trust is a quality criterion that is more important to organizations in business, because they are more exposed to white-collar crime than non-profit oriented organizations and often face large amount of data to processes without manual controlling. The latter is also true to the accuracy of information.

- Is the source believed to be *sincere*?
 - With what method do we assess trust?
 - With what method do we promote trust?
 - To what degree of trust shall we accept?
- What knowledge do we share taking into account enterprise and user *policy*?
- Is the information accurate in terms of precision and quality?
 - With what method do we grade accuracy
 - To what degree of accuracy shall we accept?

4. Problems in Knowledge Delivery

Last but not least we face the problem of conveying the right information to the user at the right time, respectively when he/she needs it. As Hinchcliffe mentioned, the emergent of the social web (Web 2.0 + Social characteristics) can overload a users and bind his cognitive energy to manually filter information alerts. Anyway, we are interested in supply the collaborative user with knowledge and not simply with information. Creating knowledge alerts instead of information alerts prerequisites to "predict" the current demand, assemble the knowledge, ensure trust and deliver it on time. This is, in the end, why demand modelling is so essential to all subsequent elements in the CCKM.

- To what *endpoints* and how do we deliver knowledge?
 - Do we deliver to *Centrals or Peers*?
 - Do we deliver as request for knowledge-base-change, promotion or what is the *delivery's purpose*?
 - Can information consumers deposit a knowledge question in our system, which our SECS-system can reply to as soon as it is available?
 - Do we support *proof, trust and accuracy* assessment of the recipient?
 - Do we deliver partial and incomplete knowledge?
- Is it the *right time to deliver*?
 - How do we manage the sensor-context-matching process which leads to *knowledge alerts*?
 - Do we *postpone knowledge delivery* and queue until right in time?
 - How do we treat space and temporal matrix states as seen in Figure 6?

C. Research Objective & Thesis Statement

The thesis title's buzz words are eloquently described. We also had been given an initial idea of how we can combine these elements to support knowledge-driven organizations to overcome the low-value cognitive energy focus described by Scott Carpenter [20] in chapter I.A.2.c). Thus, we make the thesis statement that represents this meta-goal on our path. Simultaneously it stands – together with the CCKM – as a side rail, preventing us to zone out.

«Semantic sensors stimulate information awareness-activities, to provide knowledge to enterprise co-workers at the right time.»

At this early point *we only skin-deep claim to build an exploratory prototype* that leverages the collaboration factor without disturbing an enterprise information worker. The client shall observe personal behaviour, describe the user's knowledge demand and prepare knowledge from local and remote sources. At the end the prototype shall raise above the major challenges from the problem statement, respectively major aspects of the CCKM. As a general rule we can say, the user confirms a benefit if the cognitive energy focus is up-shifted in the knowledge pyramid. It is the research's obligation to determine appropriate metrics to measure that up-shift.

D. Delineation, Limitations, Assumptions

1. Delineations

Workflow-guided activities are not part of this paper. The fact that the technology industry has a common sense on describing process elements with XML (see PBEL, XPDL), alleviates the task of unmistakably associating information artefacts with those elements of a strictly defined process. E.g. an activity may be associated with workers, input and output results and exact task description, naming and connecting all information artefacts required to accomplish the task. This atomic view of an activity in a chain of activities can be modelled [44]. Clearly, this comes with a true benefit; from a modelled concept, the process elements in this case, we can infer context; context which allows deriving knowledge by assembling information artefacts in at least one context. With strictly guided processes (workflows) the context is given by the process itself, e.g. "contract improvement" in the field of contract management. In this matter we can think of this approach as a closed-world assumption. No activities and artefacts can be connected, if they are not defined in the workflow. The user is urged to think of any new connection as invalid, as long it has proven universally right.

This was subject to research already in the late 90' [23] and excelled till today in theory [45-51] and implementations [52]. We will investigate on some of their concepts, which we can either extend or reject in the area of context elicitation and awareness support.

Security provisioning [53-54] represent a minor but increasing research focus in open collaborative and semantic systems. This is mainly due to the tightening of governance, specifically IT-governance, by policies. In favour of a profound research report about collaboration and awareness support, we fully neglect the security topic.

Although overlapping concepts and technology, this paper does not focus on *semantic web services*, short SWS. It is not about discovering services that provide knowledge. Alternatively, it shares the common idea of predicting a source's applicability; a web service and analogous to our topics a context-applicable piece of knowledge.

However, we will reserve the opportunity to incorporate the probabilistic, heuristic and meta-heuristic approaches [55-57] - of the field of SWS - in prediction of context and knowledge elicitation.

2. Underlying Assumptions

Throughout this paper we will always refer to an enterprise environment unless otherwise indicated. As stated in chapter I.A.1 and I.A.2 there are particular characteristics like the magnitude of information and the crucial delivery time where CIO's turn their attention to [2, 16]. Other organizations are not coercively subordinate to these very conditions.

In this socio-technical field research, in particular when it comes to statistics and probabilities, we assume normal distributions. However, we are aware, that many socio-technical observations approximate the Zipf distribution, but cannot incorporate it in this thesis because it would require pre-statistical analysis of the data set respectively the distribution. Zipf's law states that a given frequency of a property is inversely proportional to its property in a logarithmic scale. E.g. the frequency of a specific behaviour in a specific contextual activity will occur approximately x-times as much as the next frequent activity. Or, even more practically explained, a user's most activity is opening a contract document. The second most activity, that could be saving a document, is only executed half as much. The next activity in the row is four times less than the first one. So the abscissa represents the activity's rank whereas the ordinate displays the frequency measured in the cumulative distribution function.

E. Summary

After precisely expressing our intentions in previous chapters of this initial section, we summarize that our activities take place in the research field of collaborative computing, in the area of semantics with the focus on instant awareness stimulation.

F. Overview of Sections and Chapters

Chapter II, *Literature Review*, identifies related work done in the four chain elements of the CCKM. The chain elements are disproportional containers which already assisted in identifying problems in the field as described in the Introduction, chapter I. It goes without saying that the fields of enterprise collaboration and semantics are wide. Nevertheless, we verbosely enlighten parts of related theses and projects that share the same characteristics with the chain elements, avoiding being trapped on the way to our ambitious thesis statement. The research gap analysis mentions word by word the missing efforts done to overcome the problems mentioned in the Introduction.

The *Research Design*, chapter III, provides an insight of the type of research we are going to conduct in order to meet our objective. It also lists the limitations of the methodology and comes forward with a proposal to mitigate the shortcoming.

Comprised of three major sub-chapters, the *Body* in section IV is our core contribution. The Analysis extracts the requirements from a genuine and applied field, in this case contract management at Alpiq Inc. [58], a major player within the Swiss electricity industry. A tentative Hypothesis is made. The Architecture sub-chapter is heavily guided by the CCKM and tries to satisfy the requirements extracted from the Analysis. It also illuminates the architectural decision made in order to best-possibly support the user. The succeeding sub-chapter describes the Implementation, but more important, it contains a sound

Introduction

Assessment of the appliance within this study. That is to say, the application's results are opposed to the requirements, and then qualitative verified for benefits or failure.

The last chapter *summarizes the research* in terms of thesis history and success. It also offers incentives for subsequent and promising research activities.

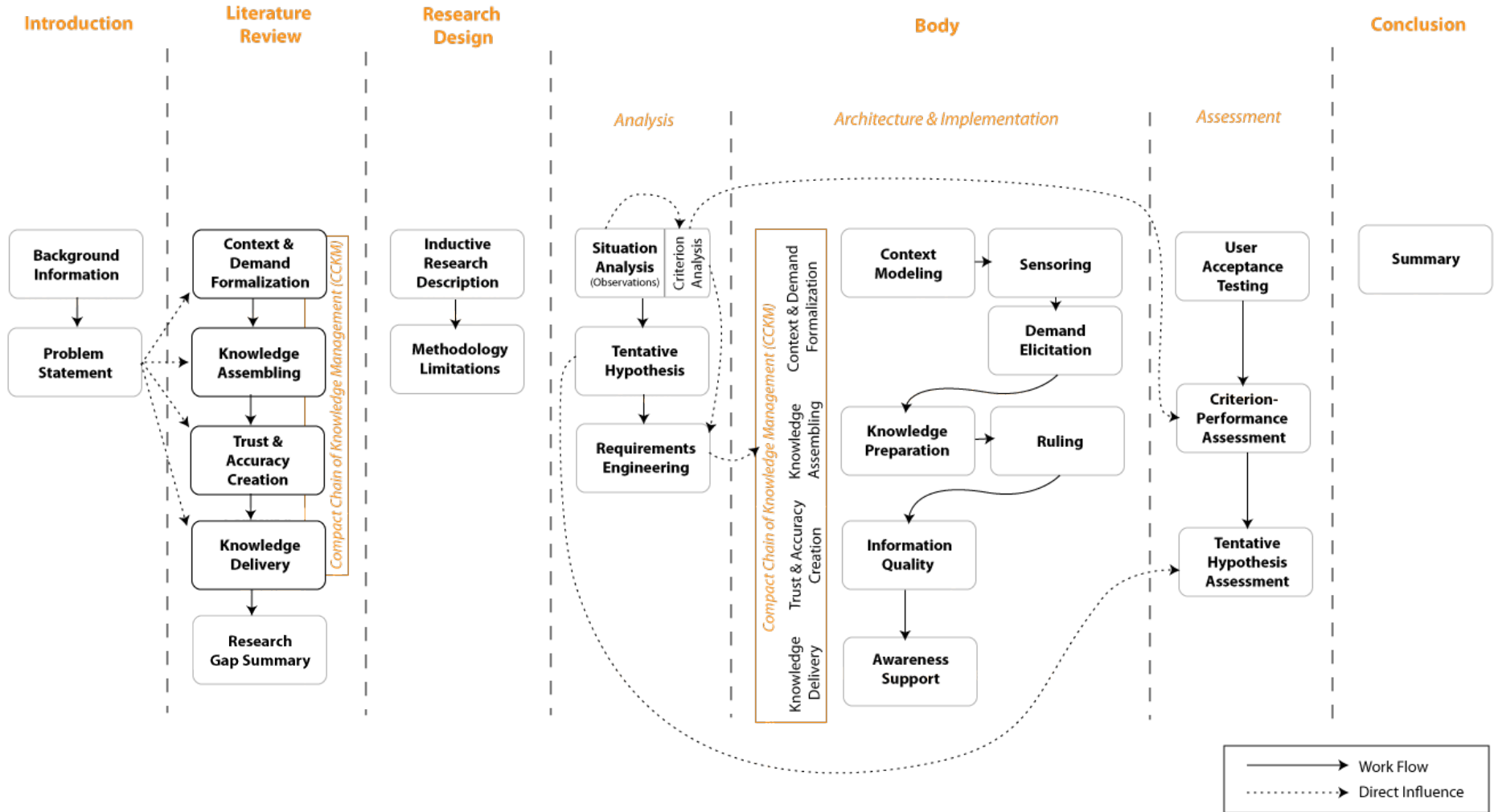


Figure 11 - Thesis Road Map with an Emphasis on Research Stringency

II. Literature Review & Related Work

This chapter is mainly guided on our compact chain of knowledge management (CCKM). It is right to say that this chain is limited due to the fact that only primary [5] and not secondary knowledge management activities [25] are included. Since it is this paper's ultimate objective to support collaborative computing with the focus on awareness stimulation, our CCKM is sufficient. The CCKM is our thread in this chapter, which covers related research so far conducted. We explain research and its problems, applied methodologies and the results for the purpose of the identification of the *research gap*.

The intention to up-shift cognitive energy [20] demands an alignment for business and knowledge strategy [4]. In 1998, at the American productivity & Quality Center (APOC), O'Dell and Grayson identified six strategies [34] how leading organizations handle knowledge. One might challenge this strict division of those strategies, because from recent studies we have learned of tendencies that favour certain strategy combinations. Information workers are personally responsible for gathering their knowledge and that is increasingly influenced by social networking activities, especially when it is difficult to externalize knowledge and retrieve it from a central source. Secondly, the strategy of "Transfer of Knowledge and Best Practices" is advised to be aligned with the former, because it allows knowledge to flow consistently without any intra-organizational obstacles.

The merger of these two strategies suits us best and brings us directly to the first sub-chapter: Context & Demand Formalization . This is – and we repeat – because we claim that today in enterprises there is a personal duty to gather and share knowledge, and we want to keep it machine-gearable as reasoned earlier in Figure 7. We are not among the firsts trying to identify personal knowledge demand in conjunction with formalized models, as we can see in the next chapter.

A. Context & Demand Formalization

1. Context and Context Modelling

As previously mentioned in chapter I.A.2.c), we need to process only *relevant information* to be efficient and in consequence competitive. Relevant is considered information that is valuable in a certain situation. We call this context. Since context can be interpreted so widely Dey and Abowd [59] backed up and abstracted the term's meaning:

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

Characterizing a situation with information is what *context-building* is about. Although context is difficult to define, it is for a good reason to categorize it. It helps to grasp the slippery elements and make sure it can be projected to a system, e.g. to model information demands. Dey et al. extracted from their statement four aspects. Post-definitions of context [26, 60] used to be limited to a few of the aspects or simply summarized them under fewer aspects.

a) Proposals for Contextual Aspects

We want to verify the primary context aspects from Dey et al., an early and well known context model, if it is suitable for our purpose:

- Location of user and (re)sources
- Time of activity, location or identity
- Activity of user or others
- Identity of user or others

Beware, that an entity of an aspect can be included or excluded from the relevancy, and thus from context, by changing a single information piece of any aspect. E.g. in contract management co-worker A, who resides in the same location, provides business parameters to our information demand of our activity of business requirements engineering. However, for technical requirements engineering activities co-worker B might be relevant (who knows of background information that significantly would change the contracts content or form) – irregardless if he/she resides in the same location or not.

It is certainly perky to challenge these four well argued and even often applied aspects [61], as Kaenampornpan et al. evidently list. With all respect, we impute previous researches in this field [26, 60, 62] that they concluded a narrow-minded argumentation about the appliance of context. While being reasonable that the four aspects influence a user's or a system's behaviour, it is missed to capture the "why" that the user is on. In other words, *why is a user doing what she does?* We can guess about this omission. Researches understand the combination of the four aspects as acceptable evidence – or at least as belief, depending whether they are Frequentists or Bayesians – to infer a context by a formal measure of (un)certainty, that is probability. For this reason, they might see the "why" as an output of their inference, respectively as an answer in the pursuit of context clarity. Taking up this stance bears risk, because at an early stage we eliminate the opportunity to appoint a context with the mission of the user. One might say it is impossible to ever "compute" the mission or objective of a user from the beginning. Yes. That is true. But why do we exclude elements in the progress of context inference, just because they are not at our disposal from the beginning? Besides, there are situations where you do have insufficient information even about a primary context aspect and therefore can also not use it right away.

We describe a handy scenario from contract management to explain "mission" as a plausible context aspect in conjunction with Dey and Abowd's context model.

Scenario	At Alpiq AG, John Doe, a contract manager, prepares a new contract version that is an obligation to them and to the local Zurich Power Company (ZPC) coming into effect next year.
Context Elicitation from Primary Context Aspects	Location Update Location: Olten Affecting Supplier Area: Zurich Affecting Sales Areas: Zurich
	Time Update: Now; Contract Start: 20XX;
	Activity Revise Contract, Draft a Contract
	Identity Alpiq AG, power marketer John Doe, Contract Manager; Zurich Power Company ZPC, local power supplier

Table 1 - Scenario to Explain the Importance of the Context Aspect "Mission"

Now, from being provided with this knowledge extracted in Table 1 for the first time, one might not propose any helpful action. For example: It could be fatal to automatically inform the supplier, ZPC, if that document is populated with biased service delivery improvements and no compensation, in a first draft. Nevertheless, we could combine these information pieces and come to the conclusion that John Doe is on the repetitive and annual mission of contract improvement. That sounds like an elicited context. Indeed, context and mission can comprise the four primary aspects, but the difference is the fact that a mission has a defined direction, a target. The Oxford Dictionary [12] defines mission as “an important assignment” and – as we know – assignments have objectives, whereas a context doesn’t necessarily. Therefore a context can be viewed as generalization of a mission. After all, we agree that mission is not necessarily suitable to represent a primary context aspect, also for the fact that mission is a container to a collection of primary aspects – btw, he calls them categories. Short, it has not a 1-to1- relationship as the other aspects possess. Still though, we take the stance with other researches [61, 63] that mission ought to take part in an extended aspect model of Dey and Abowd to enrich the inference process with a direction. From now on, we consider “mission” as a meta aspect, which contains primary aspects.

In the first instance the *Location aspect* is well thought-out [59, 63-65] the user’s current physical position. It is considered the predominant aspect [59, 63, 66] in adaptive systems and used to imply what users are doing either sole from their position or coupling the position with other contextual information. In principal, developers limit the appliance of the Location context aspect to a single subject that is the location of user who operates a system or the location of the system itself [67]. However, Location aspect is should not be seen as limited to the position of the user, but also of the areas that she influences or he is affected – as noted in our example. Still though, one could think about other entities in the context as marked in italic in Table 1. While the contract revision is edited in Olten, the affected areas are the sales and supplier region of Zurich. From the context model’s creator we miss a hint on how broad these aspects reach. We will try to give this categorization a second guess. For example, a location could also refer to an IP address, a system’s place where we retrieve data from, the characteristics of a place like the population or density of users, or even a path of locations we could use to derive knowledge. *Time* itself means nothing to us unless it is at least coupled to one of the primary context aspects. Space and time in particular share a powerful combination, because the majority of events are attributed with a place and certainly a time span from what we can imply many possibilities. In literature [63, 68-71] we learn about other applied temporal structures like the interval, which describes the repetition of which an event occurs. That could be represented by the first day of the week, every Wednesday, every leap year, etc. In addition, in 2007 Bolchini et al. [67] introduced that Location and Time are modelled either absolute or

relative. Absolute is, for temporal example, the 1st March 2010 GMT, whereas the relative is “2 hours from now” and now takes respect to the time zone and current date. The frequency itself, which describes the number of times an event occurs in a time span. With all the xillion tasks of enterprise users, a system faces a hard time in deriving a context having only time and location information available. Therefore, Dey and Abowd added the *Activity* context aspect which answers the “what” question. It is doubtlessly easier to imply a user’s intention by knowing about his activities. The sequence of activities, for example, shed light on what the user is up to. If a “secondary pieces of context” [59] like an email address can be bound to an activity, it is nearby to think of a communication activity between the user and another individual or group. This leads to the last aspect, the *Identity*. Same as with Location, to know the user’s identity is often a prerequisite, but to know other entities and having them connected with a sound activity description, is also of great value to context inference. Looking at the extracted entities in Table 1 at a glance, one could say with a solid belief she knows of the context and scenario the users operates in. We can also say that we assess Dey and Abowd’s context aspects – with the exception of the additional *Mission* meta context aspect – as suitable for our purpose. Since we have contrasted the field of context and context aspects, we have a deeper and more schematic look at what other researchers say, which we want to share for the purpose of critical literature review.

Chen and Kotz [72], later in time, list four contextual aspects. *Computational* context affects states of devices and the network used to work with. The *Physical* context type is an umbrella for true physical, natural states e.g. temperature, acoustics, luminosity, etcetera. Most popular context type is the *User* context that is comprised of persons, places and objects. *Temporal* type is the only missing and leaves no doubt about the purpose. This context type model is adapted against the background of mobile computing, speaking of entities that frequently change their location and thus their environment. Kaenampornpan and O’Neil share the indisputable need about the entities of these aspects but summarize computer systems under the first level context aspect *Physical*, rather thinking of *Physical* as existence than classical elements of physics like temperature and acoustics. Stringently they create the aspect *Virtual*, which holds entities of networks as IP address, security roles and others. Schmidt et al. limited their context categorization to *Physical Environment*, *Human Factors* and *Time* [73]. The first aspect they pack with the location of the system – which they assume to be the same as the user. In addition they see classical physical elements as working conditions which is indeed where the technology infrastructure has to operate in. The Human Factors aspect doesn’t simply pay respect to the user, yet also to her social environment. In what emotional state is the user? What habits exhibits the user? With whom does the user communicate? What group dynamics is he bound to? But also, what task does he have to fulfil. In this sub aspect *Task* Schmidt et al. also ask for the “general goals”, which we call the mission. Of course they list the commonly accepted entities like the task one is engaged in, spontaneous tasks and tasks one is up to perform. Abecker et al. [23] strictly simplified and did put context entities in one aspect called *Enterprise*. In this container they structure employee as part of a department and the department as part of an organization; and the same for activity as part of a process. This seems too little for a serious context-aware application but they heavily rely on process-information in order to solve the relevancy dilemma; nevertheless, they put a new argumentation forward that probably has inspired many other researchers: the context a document was created in, must be exploited for an estimation of its relevance in

an new, possibly similar, application context. Other researchers [74-75] termed this the context history, thus either another aspect to elicit context from historical data of other aspects [74], or from passed elicited context [23, 67], or both.

In a nutshell, there are *several solid context categorizations*, which term similar meanings. Some context models are more granular and have different structures to model a context. While a highly granular model fosters the practical applicability of a system in a certain area e.g. contract management, it is “often inversely proportional to the generality of the model: the more expressive [...], the less practical and usable.” conclude Bolchini et al. in their context model survey [67].

b) Context Model Requirements

We recall though that we need to decide on a context type model that guides us best through the architectural task of our exploratory prototype. There looked at many comparisons [61, 67, 76-79] conducted for this very purpose – some in textual form, some even weighted in matrices. With the outmost probability we can say that these researches evolved their work from the predecessor’s valuable contribution; not just in volume but also in schematic classification.

Kaenampornpan and O’Neill profiled the *aspects of context models* in a comparative matrix [61]. They have chosen Location, [physical] Conditions, Infrastructure (computing environment), Information on User, Social, User Activity, Time and Device Characteristics. The report results in the confirmation that “there is no clear and unified understanding of context [models], which leads to ambiguity in design and implementation” and the named matrix that lists all pertinent context models known before 2004. Interestingly enough their self-proposed context model is based on psychologists Vygotsky, Rubinstein, Leont’ev’s highly renowned Activity Theory [80], which is a philosophical framework to conceptualizes human activities that Engeström amended with a triangular structure [81] to emphasize the importance of the object transformation by any subject, respectively an individual. The object is the individual’s intention. Often not tangible, but it is definitely the element that must be taken into account for a successful transformation into a valuable outcome. In Kaenampornpan et al. this object is namely the computer-supported user’s objective. In this paper the outcome refers to a completed mission of the user, whereas the success of the accomplishment can vary as the next cited researches point out.

In 2003, Korpipää et al. [82] mentioned four requirement criterions for context models that prevailed. They can also be considered as quality elements and fundament for a successful implementation.

- **Simplicity:** expressions and relations should be simple to avoid confusion in implementation and maintenance
- **Flexibility and Extensibility:** allow new context elements when needed
- **Generality:** be universally applicable, not just to a certain scope e.g. industry
- **Expressiveness:** as simple as possible, but not simpler than a context requires to be distinct

Against the background of increasing research and prototyping Krummenacher et al. [83] made greater demands towards context models, from what new criterions emerged.

- **Applicability:** similar to Generality
- **Comparability:** the provided values of accepted sensor need to be comparable among the context models

- Traceability: the provenance of information must be transparent
- History: the mutation of context and context aspects must be documented to avoid non-sense implications
- Quality: sensed data varies over time and the model must provide a concept to handle this fact
- Satisfiability: conformance ensures that data values can be processed e.g. the model allows to define applicable ranges and domains
- Inference: high-order context can be derived from low-order context e.g. a new situation or a process is recognized by a specific sequence of activities
- Incompleteness and Ambiguity: ability to disclose and model inconsistency and incomplete context

Strang and Linnhoff took a narrow view [76] for the sake of practicability, in 2004. They outlined very *specific requirement criteria* which go beyond Korpipää et al.'s definitions. They say that Distributed Computing evolved to Mobile Computing continued the evolution chain to ubiquitous and pervasive computing stipulating higher demands, which ought to support context-awareness, ad-hoc networks and smart-sensors & devices. For the context-awareness they classify the demand in six groups. 1) Capability for Distributed Compositions (dc) must live up to the coordinative features of a central instance in terms of the creation and maintenance of context description. 2) Partial Validation (pv) described as the demand for validity-checking for contextual interrelationships to avoid error-prone modelling. 3) Richness and quality of Information (qua) desires to rank the sensor-delivered information in contrast to context knowledge validation demanded by (pv). 4) Incompleteness and Ambiguity (inc) denotes the obvious risk of insufficient or inconsistent context information and requires to fill the gap or to disambiguate competing assertions and concepts. 5) Level of Formality (for) addresses the need to describe context elements with a shared understanding among all participants, and since numerous ubiquitous computing applications act without a truth-dictating nucleus, traceability is vital. 6) Applicability to Existing Environments (app) alludes closely what Bolchini et al. lifted out of the attribute granularity. Some models do not satisfy demands and constraints of any context for the simple reason of specialization. However, context models can be prepared for expandability e.g. by rejecting static formalization syntax as described in the next paragraph.

c) Formal Types of Context Models

Strang et al. identified six “modelling approaches” to which we refer as *formal types of context models*. The simplest is the *key-value pair* formalization which is established by a variable name for a context aspect and a given value. These values are “easy to manage but lack capabilities of structuring” and do not offer the power to disambiguate competing context descriptions or meanings. Also, it’s tremendously difficult to share and incorporate them among distributed autonomous systems. In fact key-value pairs can only assure the demand criterion of applicability (app).

To overcome the low-expressiveness of key-value pairs and to satisfy the need to merge different context description W3C reverted to the well established Resource Description Framework [84], based on the markup language XML. Indicated by its name, RDF could help to describe and combine the user context aspect, which resulted in a recommendation for a Composite Capabilities / Preference Profile, CC/PP [85]. There appeared other approaches [86-87] that got rid of the static structure of CC/PP and improved the context merging ability. Pervasive Profile Description Language (PPDL) [88] tackled the problem of

constraints and relation expression but strongly limited itself by omitting important context aspects. The idea of hierarchically describe context is denoted as *markup scheme models*. In you see an example of another extended CC/PP approach, the User Agent Profile (UAProf) [89]. This framework was driven by the need of identifying mobile devices during the short rise of WAP 2.0 and serves the context aspects of location, time and identity. An example of this RDF-based profile is attached to the Appendix.

The *graphical model* approach is based on UML, in particular on UML diagrams that are able to create edges or predicates to describe relationships. In most cases they are derived from the class diagram type of UML. Henricksen et al. [90] extended the Object-Role Model (ORM) [91] that concentrates on facts and fact types. They expressly pay tribute to the factor of change by creating a fact type static, indicating whether a fact persists true in any constellation and is always true regardless of time – indeed very similar to a constant. Also, in this approach they differentiate weather a fact is profiled e.g. by the user itself, stated by a sensor or even derived by existing facts of a source like a knowledge base. In other words, they do not simply use facts of content in the context like location, time, identity, activity and mission; in particular they attach information about the content. As “about” implies, this is meta data on context aspects add to information quality. After all, these graphical models do not qualify for distributed systems (dc) due to their lack of expressiveness, impeded by non-machine-gearable syntax. In consequence validation (pv) and the handle of completeness (inc) is insufficient.

Object-oriented models employ the main benefits of the object-oriented idea to address the dynamics of context in ubiquitous computing. For example, in volatile environment a developer might focus on the most needed elements like attributes and functions of a superior class that is valid to any information supplied by any information supplier, i.e. a sensor. If observations on a new pattern are made, it is inexpensive to create a sub-class that can inherit properties but still though thoroughly specify the new class for the benefit of valuable context identification. Of course, object-oriented models operate with the various relations a graphical model does, and it is only limited by the programming language used. However, OOMs compensate the deficiency of graphical models: (dc) ability is achieved with OOP language e.g. Java Persistence API that offers full support for context sharing by sending Objects over the net. OOP-languages offer high degree of formalization, thus the resolution of ambiguity (inc) can be achieved as we learned from several projects [66, 92-93].

Logical based models show a high degree of formality (for) by nature, which eases the process inference respectively the creation of new facts from existing knowledge. Inference is employed by rules as we see later in this section. Briefly speaking, rules consist of one or a series of conditional statements that end in adding facts to a knowledgebase. Since context and its aspects are slippery terms, conditions are often softened with fuzzy logic [94-97] to converge a best possible statement. The implementation of logical models is often achieved with Prolog – but gives a hard time to implement in a ubiquitous computing device (app) due to its memory intensive reasoning engine with low programming interface control. In none of the models and systems we looked at is a way to handle incompleteness and ambiguity (inc). The theory in context modelling did so far not approach optimization techniques. This is certainly a general gap on the way to effective context elicitation. Due to a stronger focus on information quality (qua) and

validation (pv) – to be competitive in the real world –, we notice a deviation from purely logical based models in numbers of research papers and bad critics.

Ontologies have its seeds in the doctrine of “existence” or “being”. This knowledge structure tries to illustrate the relationship of concepts within a domain. A domain is usually defined by the elements of one or more contexts. *Ontology based models* exploit the highly extensibility (dc) and formality (for) and use existing inference engines to check on validity (pv), completeness (inc) and can detect ambiguity (inc). With emerged ontology languages like OWL, and in particular OWL-DL, one can profit from description-logic-valid syntax and semantics (app). While a solid fundament exists, the requirement criteria (pv), (inc) and (app) call for a mathematical expertise and programming effort that is not to be sneezed at. This argumentation is supported by a rising number of applied models and projects [83, 98-99] like CONON which implements the four aspects of Dey et al. [59] inspired by Ryan et al. [60]: Location, Time, Identity and Activity.

Context Model Type	dc	pv	qua	inc	for	app
Key-Value Models	-	-	-	-	-	+
Markup Scheme Models	+	++	-	-	+	++
Graphical Models	-	-	+	-	+	+
Object Oriented Models	++	+	+	+	+	+
Logic Based Models	++	-	-	-	++	-
Ontology Based Models	++	++	+	+	++	+

Table 2 – Commonly Agreed Formal Types of Context Models, © Strang and Linnhoff-Popien

Strang et al.’s context modelling survey [76], among others, made clear that ontology based context model offer the highest potential for context elicitation. The assessment of developed practical models is not dominated by a single research group. However, we recommend William Loyola’s elaborated comparison of approaches [79] where he assigns developed models to formal types of context models and lists features and properties they offer. A bit later in time, 2007, Baldauf et al. [100] and Bolchini [101] give a sound matrix on a criteria-based assessment of developed context models. In the focus of this research paper, we put ontology-based context models on top and eliminate all other types of context models, as a result from our reasoning and Strang and Linnhoff-Popien’s conclusion in Table 2. In our semantic related projects [44, 102] we experiences additional technical benefits. Ontologies can be reused, are code independent, and they assist to pursuit the n-tier approach, which strictly divides data from business logic. Based on XML, Ontologies are also “extremely interoperable” [53]. Ontologies offer the widest representation and go far beyond the hierarchical characteristics of taxonomy and relation-limited topic maps. Ontologies may also be equipped with the popular rule languages like SWRL [103] which enables a system to imply facts that are so far not stored in the knowledge base.

System	Space	Time	Space/Time coordinates (Relative or Absolute)	Context history	Subject (User or Application)	User profile (Role or Features based)	Variable context granularity	Valid context constraints	Type of formalism: Key-value-based	Type of formalism: Mark-up based	Type of formalism: Logic-based	Type of formalism: Graph-based	Type of formalism: Ontology-based	Formality level (High or Low)	Flexibility	Context construction (Distributed or Centralized)	Context reasoning	Context quality monitoring	Ambiguity/Incompleteness mgmt.	Automatic learning features	Multi-context model
ACTIVITY	+		A	+	U	F	+							L	+	C	+				+
CASS	+	+		+	U									L		D	+				
CoBrA	+	+	A		A	F					+			H	+	D	+				
CoDaMoS	+	+	R/A		A	F							+	H	+	D	+	+			
COMANTO	+	+	R/A		A	F							+	H	+	D	+				
Context-ADDICT	+	+	R/A		A	R	+	+	+				+	H	+	C	+				
Conceptual-CM	+	+	R	+	A	R							+	L	+	C	+				
CSCP					A	F			+					L	+	C	+				+
EXPDOC		+	R	+	U	F					+			H	+	C	+				+
FAWIS				+	U	F	+		+					H	+	C	+				+
Graphical-CM	+	+	R		A	F						+		H	+	C	+	+			+
HIPS/HyperAudio	+	+	A	+	U	F			+					L	+	C	+	+			+
MAIS	+	+	A		A	F	+						+	H	+	C	+	+			+
SCOPES					A						+			H	+	D	+	+			+
SOCAM	+	+	R/A		A	F							+	H	+	D	+	+			
U-Learn	+	+	A		U	F		+					+	H	+	D	+	+			+

Table 3 - Comparative Matrix of Context Models, Aspects, Model Types and features, © Bolchini et al.

In the next sub chapter we investigate – besides others – the technical concepts in order to discover their strength, weakness, and optimal application area. Krummenacher et al. did not just provide theory with context modelling criteria but also with ontology engineering criteria [83]. These criteria we want to fulfil in our own context model.

2. Context-Awareness

Context-awareness is a term first introduced by Schilit et al in 1994 [104]. It describes a class of applications which can sense the context they run in. Most researches confirmed [27, 73, 105-106] context-awareness also implies the system to be adaptive to the user’s environment and behaviour. That is to say, the output of the system is likely to vary, influenced by context aspects entities. Change, customize, adjust are used verbal synonyms for adaptive respectively adaptive systems. This common sense is comprised of two facts: a system is aware of a context and - also - can adapt to it. Gutwin et al. [107] make the following definition for context-awareness.

“A system is context-aware if it uses context to provide information and/or services to the user, where relevancy depends on the user’s task.”

a) Types and Aspects of Context-Awareness

Sometimes “type of context-awareness” is mistaken [108] as an aspect of context, or category of context as Dey et al. [59] termed it. Let us resolve this danger. Gutwin et al. [107] were the first to denote four types

of context-awareness. It is used for eLearning, but is generic enough for our meta objective and therefore noteworthy as earlier [104] and later [109-110] definitions from Schilit and Pascoe. The *social type of context-awareness* is dedicated to the interaction with others. In particular finding out who is important to one's mission, who's available, etc. Further this type of awareness concentrates on what roles exist and are there policies to consider? *Task-awareness* not just incorporates questions on what is already known about a task, but also the awareness of others in regards to another individual task or a collection of tasks. Do we need to inform others about a certain task? Or more elementary, does the system need to inform the user? If yes, when do we make the user aware, how and are there any consequences the system or user needs to be aware of? More abstract is *concept-awareness*. It comprises the ability to understand and treat the concept and develop a meaningful notion. A meaningful notion, one learns about, could force him or her to revise other concepts or even entire topics. Concept-awareness is highly powerful, because it allows a user and system to react and to render its knowledge more precisely, interpolate and extend it – again, preconditioning concept-awareness. *Workspace* is the last awareness type and is about pure interactions and collaboration with others. It bundles the answers to the previous questions in an agile environment and ought to provide it in real-time to the user. At all time it is confronted with the question: do I inform the user or not? The system – whether that is a local agent or a central instance – bundles this information on a workspace with a GUI.

While Schilit's term definitions of context-awareness features [104] seem abstract and non-intuitive, Pascoe [110] arranged the features of his definition in a sequential temporal way. They term the ability to identify and recognize a context *contextual sensing*. In comparison to Schilit's *proximate selection*, it also considers remote and foreign contexts. For example in contract management, a new contract manager's system could ask a central context management instance for context-identification support, to understand in what process state a contract is. *Contextual adaption* is, very generally speaking, the ability to alter attributes of any entity relevant to a context and is also capable of executing a task or service in the presence of specific constellations of a context. With Schilit's words, this is categorized as *context-triggered actions* and prerequisites that a system is able to discover new resources. And so, Pascoe's *contextual resource discovery* slightly matches the feature description for Schilit's *automatic contextual reconfiguration*. Because Schilit argues that the change of a context, e.g. by adding a new resource, let you "reconfigure" the user's system or at least create new interesting information for the user. This step usually happens by adjusting the GUI and graphically highlighting prominent facts. The *contextual augmentation* depicts a process of enriching an existing or new context with available resources in order to gain new information or opportunities, that is to say activities to execute.

Dey et al. [59] shared the idea of having context-awareness features in conjunction with a supportive context model and noticed common similarities. All three predecessors list the prerequisite to identify and exploit relevant resources (1). They also expect a context-aware system to act autonomously (2). Autonomous means to embody a decisive system or a self-determined system, making decisions upon a rule set executed by a rule engine. In our opinion, the intelligence of such a system ends where it fulfills the user's need of information and can up-shift the user's cognitive energy – and no more than that. We explain the upper-level limitations in section II.C. Lastly their definition also includes "to display relevant information" (3), say Dey et al. They reprehend the alleged fact that there is no *contextual augmentation* in

Schilit's taxonomy. We could argue either way, and reject it by saying that augmentation is a general prerequisite to all awareness types in Schilit's categories. On the other hand, it is not described fine granular in their document [104] to make a clear judgement. Vice-Versa, Dey et al. claim that Pascoe neglected Schilit's idea of *contextual commands*. Those gave a plain example on parameterizing a printer selection in their description, which fits into Pascoe's *contextual adaption*. The only difference is that Schilit et al. see their contextual commands as manually executed and only as a service – not a plain information delivery. From studying these taxonomies, we can say that it leaves room for interpretation for the programmer and it is subject to personal flavour what taxonomy you prefer. However, in the consequence of the argumentation of Dey et al., they derived three categories of context-awareness were derived.

- 1) Presentation of information and services to a user
- 2) Automatic execution of a service
- 3) Tagging of context to information for later retrieval

Context discovery is a feature they incorporate to the first two features, saying that both require discovery mechanism. They also abandon the concept of Schilit et al. thinking of information or services, arguing that it is "in most cases, [...] too difficult to distinguish between a presentation of information and a presentation of services.". *Tagging of context* is conceptually an identical feature as Pascoe's *contextual augmentation*, since it embodies the process of adding meta data to a resource that might be of use at a later point in time. Dey et al. are expecting a system capable of these three features to do this on its own or explicitly supported by the user.

In fact Dey et al. untie context-aware features from the idea of dimensions like Schilit's information versus service and manual versus dynamic, or from the sequentially ordered features of Pascoe et al. At the end of the day, it's is up to developer to decide what context-awareness model and context model she chooses, but it must be based on the business requirements. For example, if a business relies on social aspects, it is doubtful to favour the context model of Lieberman et al.[111] since they pay only little tribute social context modelling [61]. Therefore both models play a vital role in the software development process by influencing the function tree with context-awareness requirements and the persistency model with the imperative properties that demand adequate expressiveness. For these reasons, and opposing to Dey et al. [59], we claim that the system's anatomy will be impacted a great deal. In return it comes with the benefit of conceptual maturity.

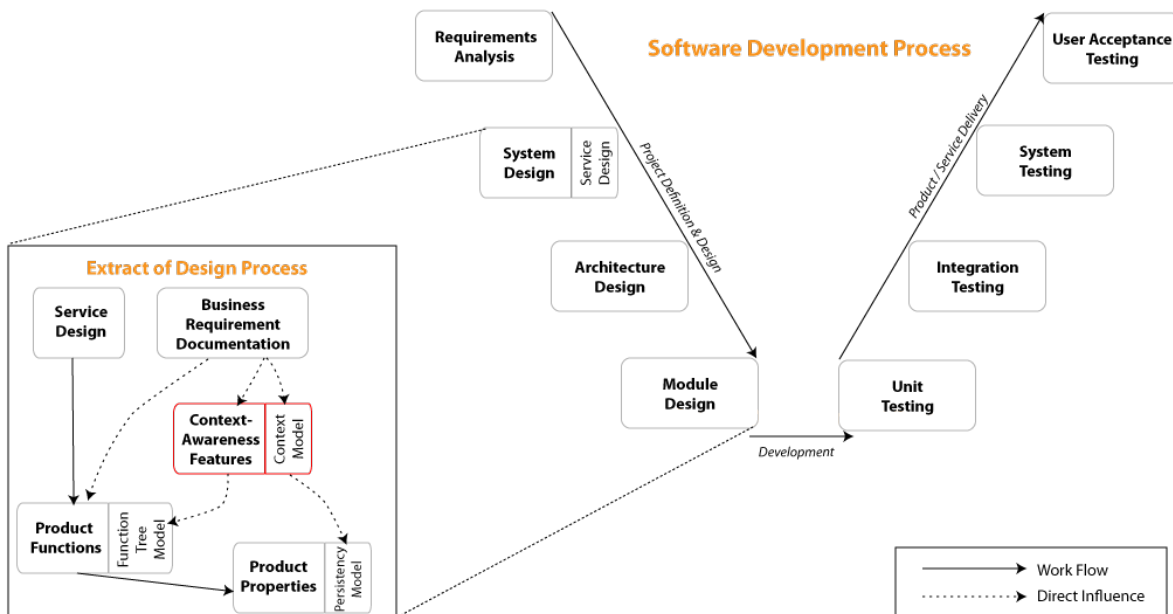


Figure 12 - Integrating Context and Context-Awareness Requirements into the Software Development Process

At this point we have a picture of what it takes to describe context, what context-aware features one might exploit and what the benefits are. As this is challenging enough, one also needs to populate a data container that relies on a context model, coming up in the next sub chapter.

3. Context Sensors - Making Sense of Context

Another explanation why the development processes changes is brought forward by Kim et al. in 2008 [112]. A context-aware system differs dramatically from a traditional service by actively delivering the content to the user in opposition to waiting for the user to crawl. This capability relies on a single major characteristic: it is fully event-driven. That is true for every application with a user interface but not if we attach the condition that the system must act autonomous. In consequence, a sensor-based system is likely less obtrusive towards the user than conventional system.

a) Sensors and Sensor Types – Data Gathering

A user's input, considered as explicit input, within an autonomous system is replaced by one or more sensors. Although sensors are widely known as context providers [113], we will stick in this paper to the plain term *sensor*. Because a sensor's obligation is plain simple, too: it must pick-up data from an environment and forward it to a registered consumer. This does not include any kind of data or information processing – except when talking about new rule-based sensors. It is other system components which provide context information, as we see later in this chapter. However, the sensors vary in terms of their delivery: data or information. We recall [13], data given a semantic, e.g. adding a unit to a bald number, turns it into information. Christopoulou et al. [99] put this in sequence by terming it *low-level context acquisition*. The state of low-level context is pure raw data and is similar in its purpose to the *contextual sensing* of Pascoe et al. [109]. It is a fairly solid concept to emphasise the role of a sensor: data provider. To create (high-level) context one must fuse and interpret data, as we have verbosely depicted in the previous chapter. The input parameters of the derived context are data fragments gathered from sensors.

Sensors can be split in two major groups, *logical and physical sensors*. Physical is about retrieving from natural conditions like temperature. The logical group is widely more interesting to this project, because it picks computer and user signals e.g. a website a user visits, a message delivery to a recipient, an opened file, etcetera. We have learned about context aspects earlier and it is a mistake to strictly assign those aspects to either logical or physical sensor groups. For example, a location aspect is a network address which is considered logical. Standing “next to a printer” is a location aspect, but is considered as a physical input, because the data comes from visual sensor, RFID [114] or any other sensor capable of capturing real-world events. But why is this distinction so important? Physical sensors regularly contemplate the application of extra hardware to sense one piece of information. That makes it expensive, not just for purchase but also for operations and maintenance. Logical sensors are often software-based and embedded into an electronic user device e.g. a personal computer in this project’s matter. Since personal computers need to be maintained anyway, these sensors come at low costs. This leads us the characterization of sensors in a system, which we listed together with applied sensors in Table 4.

Sensor, Device	Extractable Data (low-level)	Extractable Information (high-level)	Sensor Types				Sensor Characteristics						
			Anatomy of Sensed Object	Locus of Sensor	Software Source Type	Aspect	Costs of Capacity				Granularity, Quality (Potential)	Range, Accessibility	Availability
			Physical, Logical	Local, Remote	Operating System / Service, Application	Location, Identity, Activity, Time, Mission	(Initial) Creating	Acquiring	Using	Maintaining			
Email	Co-workers, Topic	Co-worker, Task, Priority	L	L	A	A, I, T	3	4	4	4	3	4	4
Calendar	Invitees, Topic, Time Slot	Co-worker, Task, Mission, Availability, Priority	L	L	A	L, A, I, T, M	3	4	2	3	4	4	4
Task-List	To-Dos, Co-Workers, Contacts	Mission, Task, Co-Worker, Priority	L	L	A	A, I, T, M	3	4	4	2	4	4	4
Address Book	Contacts	Person, Company	L	L	A	I, T	3	4	2	2	3	4	4
Network Sensor, NIC	MAC Address	Global Uniqueness of Client	L	L	OSS	L, I	1	4	1	0	1	5	5
	IP	World Location	L	L	OSS	L, T	1	4	1	0	2	5	4
Network Settings	Subnet	Nearby Devices e.g. Printers, Fax, Scanners	L	L	OSS	L, T	1	4	1	0	2	5	4
	Host (FQDN)	Host name, Organization	L	L	OSS	L, I, T	1	4	1	0	1	5	2
Application Protocol	URI	User, Current Task, Location or Resource	L	L	OSS	L, I, A, T	1	4	3	0	3	5	4
File System	Open Files	Current Task, Users, Roles	L	L	OSS	A, I, T	1	4	4	0	3	5	5
Task and Process Manager	Running Applications	Current Task, Users	L	L	OSS	A, I, T	1	4	4	0	2	5	5
Print Manager	Print Commands	Document, # of Documents	L	L	OSS	A, I, T	1	4	2	0	2	5	5
Desktop	Visual Arrangements and GUI Inputs	Application, Document and User Relationships, and Workflows	L	L	OSS	L, I, A, T, M	1	4	4	0	5	5	5

Table 4 – Examples of Sensors, Sensor Types and Their Characteristics

Sensors can be installed locally or remotely. As you can see from the reference table, remote sensors tend to be of higher costs and lower accessibility and lower availability. Acquisition is also often costly because searching and attaching a useful sensor requires capacity. Numerous implementations therefore relinquished from the *dynamic acquisition of sensors*; SAMOA [115], CAMUS [116]. On the other hand, remote sensors are often not subject of one’s maintenance responsibility. As a rule of thumb, one may assert, that application sensors are more expensive to create than operating system sensors, for the simple reasons of proprietary and the wide range of vendors – at least way less than operating system vendors. Another downside when creating a sensor for applications is that there is often no API to access

application data, of course, often for the good of security. Also, application sensors have higher maintenance costs due to the more frequent changes of applications compared to OSS. However, Hebel et al. point out [117] that new xml based microformats like hCard/iCard for contacts and hCalendar/iCalendar for calendar entries *alleviate the using of application sensor data* significantly. Another characteristic is the *locus of a sensor*. Is it local and manageable at any time, or is it from a remote site with little to no guaranty for accessibility, availability and quality?

In a competitive business environment, three main criteria prevail when it comes down to choosing a sensor for enterprise collaboration: 1) the *granularity of data*, which can be seen as the potential to elicit knowledge and context; 2) the *frequency of usage* of an application embedded sensor, i. e. it is more likely that a user works with groupware software, than he changes his IP address. If the IP address does not change, no event nor signal can be thrown to derive knowledge. Thus, application sensors are in general more precious than OSS sensors. If we recall the richness and quality requirements from Strang et al. [76], one understands that a low-granular data sensor contributes little to context elicitation. These days there are many projects [118-120] that led to application plug-ins to leverage high-granular application data. In 2005 Sauer mann et al. made clear that it takes a “pluggable architecture” and expensive adapters [120] to leverage application information. 3) The *cost of a sensor* in its proportion to acquisition effort and benefits matters.

b) Context Inference - Deriving Information from Signals

So far we thought about the characteristics of sensors. As described in chapter II.A.1.a), it is easy to imagine that an aligned combination of sensors increases progressively the value of information extraction from a context. This is for two obvious reasons: “Sensing technologies are not 100% reliable” as Abowd and Mynatt [63] state, and a combination with another sensor would help to isolate wrong data. The second reason is similar but resides on a higher level. It’s about high-level context; in other words, context elicitation. Being provided with raw data from various sources that feed a context model, empowers a system to combine the aspects and elicit context and knowledge.

Aforementioned, creating and acquiring sensors is costly [121]. Deriving information is less expensive, especially since computational capacity is purchasable at low cost [122]. Systems tend to do so using some kind of formal logic; that is to say, propositional logic in general and description logic as part of subset of first-order predicate logic. Description logic, as the others logics, has the ultimate purpose to reason on properties of entities. This process is called *inference or reasoning*. DL operates with a terminological box (TBox) and an assertion box (ABox). While the TBox describes concepts of domains, the ABox “invokes” instances of those concepts, names relations and constraints, but always complies with the prescribed concepts and relational properties. The sum of entities in T- and ABox are considered a *knowledge base*, although Walton [123] disagrees and limits the knowledge base to the ABox – rejecting the fact, that domain knowledge e.g. in an ontology, is knowledge, too.

Smart sensors or components with the capability of reasoning are often termed context synthesizer [110, 113] when one tries to elicit context from a number of signals. As we have learned, a context is a vague concept; slippery and tricky to grasp. For these difficult concepts there is often not a true or false answer applicable. Let’s give an example from “Approval Management at Alpiq”. The company’s contract

management system does not handle a machine-controlled approval mechanism. Therefore the creator of a contract must be sensitive to existing issued approval policies. Imagine the SECS system would limit its recommendations of approvers it knows from past observations of the same activity. Because of this strict historical relation two deficiencies occur: 1) recently hired employees are not considered and 2) recently leaving approvers neither. Fuzzy-logic is a more *tolerant decision approach*, which includes potential valid elements, in this case potential approvers. “It takes fuzzy-logics to cope with the uncertainty in captured context” say Harroud et al. [124]. Elie Sanchez criticizes [125] the established ignorance of *soft logic*, which is considered the way human reasons contrary to *hard logic* of machines. Moser and Zimmerli did considered fuzzy-logic as rule representation [44] in their prototype of a semantic enterprise [process] repository. One may think of rules combined with fuzzy-logic as an approximation towards knowledge – accounting for Sanchez’ critique. A *set of rules*, often denoted by a rules-base, represent an expertise on how to solve a problem of uncertainty. Rule and logic systems are confronted with knowledge of diverse completeness level. That is exact, uncertain, incomplete and vague knowledge according Hinkelmann [126]. The degree of uncertainty correlates with the tolerance of reasoning an assertion against the knowledge base. Putting this picture together, facts and rules represent a knowledge base. The former provides domain knowledge, the latter “problem solving knowledge” [126].

SWRL is a rule language proposal [127] of the W3C. It primarily builds on OWL-DL, horn-like rules and the syntax of the older Rule Markup Language (RuleML). “In particular, SWRL fails at representing vague and imprecise knowledge and information.” is what Pan et al. [128] argue in regards to this rule language. Therefore they create f-SWRL as an initial draft to overcome the deficiency of SWRL. It is basically about incorporating weights in the well-known concepts of antecedent and consequent. Find a primitive example of a single atomic *f-SWRL* rule in the context of “Approval Management at Alpiq” below:

$$\text{PreviousApprovals}(?e) \times 0.3 \wedge \text{ContractExperience}(?e) \times 0.8 \wedge \text{lawyer}(?e, ?x) \times 1.0 \rightarrow \text{Approver}(?e) \quad (1)$$

The fact that previous approvals have been done by an employee (a), is not as severe as the contract experience and the axiom “lawyer” of our company. Rule (1) entails that (e) is an approver if all conditions are fulfilled to the prescribed degree. Yes, the increase of axioms adds to the approximation to truth as long as the axioms are defined for the used instance and applied with a business sense. In consequence; one can use an axiom with 0.0 degree no negate; one makes the rule engine throw an error putting an instance of a wrong concepts into an axiom; one does not create value adding non-sense predicates to the rule. And not less important, f-SWRL sticks to the forward and backward chaining feature. Of course, there are more fuzzy-logic approaches [94, 96, 125, 129-130], while they have found similar ways to deal with the fuzzy vagueness of knowledge; they showed diversity in computing the degree, respectively the probability of uncertainty to make fuzzy-rules possible.

Forward chaining is data-driven, because existing facts determine the elements of the antecedent selection list, in contrast to backward chaining, which is a goal-driven approach and chains the consequents to draw a conclusion.”

- Excursion on rule chaining -

Damáσιο et al. [131] named two major uncertainty-resolving approaches. Implication-based approaches we just looked at and probabilistic-based approaches. These initiatives [131-136] were inspired by Judea Pearl's early thoughts [137] on *probabilistic reasoning* in 1988. They handle defective models, domain mergers and non-deterministic domains and often rely on Bayes' theorem of probability. Bayes' rule in *Bayesian Networks* provides a continuous way of updating a joint probability distribution, exactly what is required in a lively collaborative environment. Think of variable b as the cause and a as the effect. Assuming there is conditional dependence; one can claim a degree of causal probability e.g. belief.

$$P(a|b) = \frac{P(a|b) \times P(a)}{P(b)} \quad (2)$$

A Bayesian network incorporates a directed acyclic graph (DAG) which represents causal relationships under uncertainty. It's is fundamental to understand, that the causal relationships are used as rules, even though they are regular conditional probabilities.

$$P(A|B_n); n \geq 0 \quad (3)$$

The conditional probability distribution is attached to every variable. In graphical notation one adds a conditional probability table (CPT) to store all causal probabilities influencing a variable. Other persistence options are decision trees and rules. New evidence of belief is updated with every data event and pushed through the Bayesian network to any variable that is influenced. In theory we refer to this idea as *belief propagation*. At least three of the approaches [133, 136, 138] suggest to extend RDF or OWL with a probabilistically annotation and transform it into Bayesian network. This choice is figuratively a backbone, because OWL supports description logic, is decidable and inference engines are available. An advantage of BNs are certainly fast response-times and the simplicity [139] which makes it fairly transparent and for human eyes. On the other hand, collecting and structuring expert knowledge is barely possible, because the networks relies on its own computed numbers and could collapse when interfering. It is a common mistake that BN's cannot handle continuous variables, but it is difficult to set up if we recall that conditional statements are involved. A simple CPT won't do it.

Less unexploited, compared to fuzzy-logic and BN, but increasingly popular are *heuristics* or meta-heuristics. The latter depicts an algorithm that is allegedly and universally applicable to every optimization problem – at least before the implementation phase. A heuristics is an educated guess in finding an optimal solution facing a problem. Some researchers implemented heuristics [140-142] to context elicitation, grabbing for the well-known benefit: the search for a solution is pruned in return to an acceptable response time that is hoped to be approximate to the best solution. For example, Bruegmann expects his ConSense system [142] to compare usage patterns in order to deduce document relations. Since their observation component gathers numerous “GUI action clusters”, it requires a method to shortcut the finding process. This approach is still under investigation and requires more observations.

c) Sensor Acquisition - How to Find a Sensor

It is inevitable to search for “sensor acquisition” and not hit on “sensor networks” [122]. This term strongly focuses on wireless physical sensors. In this thesis we limit the sensing to enterprise and PC sensors and therefore neglect this area. We restrict ourselves to sense locally, as shown in Table 4, and

receive remote information from known sources within the enterprise. In contrast to sensor-delivered data, the information is already enriched by other equitable or central entities.

Up to now, there is no best-practice in theory on how to dynamically acquire sensors, although the Gaia project [113, 143] implemented a local lookup-service that lets other distributed entities query available sources. By the majority, acquisition often relies on building a customized plug-in or wrapper as researchers did for SituAwarePhone [144], CHIL [116], CAMUS [145], CoBra [146] and many others. It's certainly a lack in research that no abstract mechanism has been developed and deployed which is able to identify and connect to a promising context provider. It is especially peculiar, because a logic sensor does not differ in its main characteristics from a semantic web service. Thus, a self-describing annotation is the only requirement that a sensor must endorse. Current context-aware systems are distinctively connected. But popularity of dynamic sensor acquisition seems to be rising in the field of remote sensor networks only e.g. in earth observation science [147].

4. User Demand Sensing and Prediction

On-demand sensing [148] is an admissible way to construct a context. But to formulate an information-demand that is not immediately raised, is challenging because it must be chronologically augmented with various - but not exclusively - user indications. It requires complex and formalized constructs like workflows [44] for a system to predict the demand of information. This is barely done in applied research, because it requires continuous computing in terms of re-evaluating context and prediction of actions on the bases of those complex constructs. After all, continuous knowledge-demand elicitation is a very capacity consuming task.

Formalizing information demand is therefore often deduced on-demand, e.g. two opened windows could imply a demand for a third application to open. A system can easily support this need and action [148].

a) Profiling User Information Demand and Bootstrap

Since we know that there is an information possession dilemma which is ultimately true to any knowledge worker or system, we want to depict a number of general recommendations that assist to resolve the dilemma but also remain within the borders of this thesis focus. The upper quarters in Figure 10 represent the awareness of possession. A user or system knows what information it posses and what is missing. The missing information needs to be collected or derived. For this purpose enterprise repositories and sensors can be consulted and reasoning engine can try to develop individual demand profiles.

To reach a mission, sometimes one is uncertain about the missing or possessing information. The lower quarters of the possession dilemma requires clarification. This problem addresses the bootstrapping of a knowledge base and personal demand profiling. The NEPOMUK Aperture Framework [149], for example, has the ability to crawl existing sources e.g. email, files, etc., to extract data objects. For the lower right quarter two scenario proffer: 1) accepting the dilemma or 2) continuously search to resolve incompleteness e.g. in a model that can detect incompleteness or inconsistency; the decision depends on capacity and capability. Bootstrapping concerns context elicitation and demand profiling on system start-

up. However, continuous demand profiling remains a gap. As a fact, a system requires a model – same as context - to store an emerging user demand that is complementary to an instant user demand.

CEO of Decisionality Ltd., Freddie McMahon, names in a blog [150] three relevant options among many others that assist to determine knowledge – and also the knowledge of demand. 1) “The aggregation of *decision-strings* is the basis for demand-sensing.” means that all implications on demand rely on an array of decisions made by the user or the system. 2) “Demand-sensing is the *emergent* behavioural flows, patterns and outcomes.” means that a system takes note of a frequent and increasing actions of a user or system. 3) “Demand-sensing provides *hindsight, insight and foresight* for continuous learning.” means that a system could revise its rules in favour of superior future decision-making.

b) Collective Demand – Collective Intelligence to Predict Individual Demand

In the past years research intensified their work [53] on distributed autonomous applications that act within a P2P-network. With this idea a primary objective is to outsource computational capacity [151] but still share information and – if available - knowledge. Autonomous applications are referred as agents [123]. “Large market leads to optimal price” is a simple example that Segaran [152] provides as the fundamental advantage of shared and distributed information that often results in knowledge. The bids and sell represent the information on demand, while the price represents the knowledge of accurate value – ignoring financial speculations for this moment. There are many straightforward mathematical techniques available like Euclidean Distance or Pearson Correlation Score [152] to detect similarities of user demands; and the annotation of information objects contributes in providing the numbers for recommendations. Many system proposals leverage collective annotation and information capturing [153-154]. Further, many architectural proposals [142, 144, 146, 155] still rely on a central unit that processes all gathered annotations and makes them available for the clients. Other systems derived from the SAMOA Project [115] and others [143, 156] disclaim on this advantage but have to deal with a serious challenge in return: the heterogeneity of contextual representation – and knowledge representation in general – is still challenging to control.

5. Fundamental Components – Ontologies and System Architectures

What are the basic components and what is the enabling technology researchers used to establish context and demand formalization? – Questions in this chapters.

a) Ontologies and Ontological Vocabularies

There’s no project we cited so far, which would ignore the mature advantages of this expressive structure as depicted in chapter II.A.1.c). Thus, a number of proposals emerged but only a few were frequently applied [157-159]. The NEPOMUK project [159] delivered a substantial set of ontologies with its NEPOMUK Ontology Pyramid. It surrounds the desktop user’s context with standard vocabulary – as we asked for in an early stage of this document – and several low-level ontologies like the file ontology (NFO), contact ontology (NCO), Image Annotation Ontology (NEXIF) and many more. The development of the semantic desktop continues but the ontologies are maintained by the Open Semantic Collaboration Architecture Foundation (OSCAF), which was exclusively founded to avoid the reinvention of the wheel in this research field. The Sile Model [157] extends and combines NEPOMUK, Dublin Core, Friend-Of-A-

Friend (FOAF) and many more have a wider predefined vocabulary with semantics available. The Sile Model and Context Unity [158] are limited in terms of handling heterogenic ontologies. Their narrow concepts make it difficult to merge ontological structures, especially in a distributed environment. NEPOMUK avails itself of the ontology pyramid which allows abstracts and general concepts in the upper layer and concretises concepts towards the lower layers. That is to say, for example, a foreign system sends a concept “energy contract” for knowledge sharing. Since the receiving system cannot classify “energy contract” it can at least deduce it belongs to a subset of the concept “contract” of a mid-layer, which in turn is a specialized concept of “document”. It is an easy way of reducing the matching effort among ontologies without risking to create redundant high-level concepts. A sophisticated approach of multiple representations of ontologies (MRMO) using SWRL is described by Moser [42].

We discussed context models already earlier in this paper, II.A.1. Context elicitation, though, is often outlined by human cognition [160] while performing a process, a series of tasks. Because a person can often only concentrate on few elements, it is eligible to assume that these elements appear within a chronological thread or string of actions. These threads are used to form an educated guess on context, resources involved and knowledge required to efficiently complete a task. Threads remain stored, often in a dedicated ontology, to be comparable with upcoming observed threads. Methodologies to recognized previous context reach from case-based reasoning to heuristics and others [113, 159-160].

When captured contexts are distributed in decentralized collaborative environment, the approach of MRMO becomes crucial, because ontologies like personal information models might have totally different semantic structures.

b) Types of System Architectures

Does theory mention considerable differences between *centralized versus disparate* systems? A centralized system has got different tasks and concentrates heavy intelligence within a single system ready to supply knowledge but not willing to share how it creates it. It is a knowledge provider that hides the knowledge transformation from the knowledge consumers [65, 123]. A central entity comes up with compelling advantages over a society of entities on an equal footing with each other. For example, it is less exposed to knowledge inconsistency and conflicts, and if, it may resolve it without having to compete with other entities about the suzerainty of truth. On the other hand, it does not consider user-specific demands and context as Walton argues in the introduction of his book [123]. This is for a simple reason; a central system has got limited memory and processing power. Of course this is true for a decentralized entity as well, as Walton outlines this as problem. But processing its own data is, first of all, easier to do because the semantic does not have to be explained and, secondly, a network of entities quickly tops the capacity of the single dedicated entity. We can expect extra performance from disparate systems. However, we need to consider the previously mentioned drawback, because this approach comes with an attitude we depreciate in line with our thesis objective: Inability of *fostering common and unified organizational know-how*. It requires new concepts in how we exchange knowledge and handle inconsistency and conflicts among numerous *knowledge propagating* entities.

While we focus on enterprise solutions, a middleware respectively a mediator layer is recommended considering our argumentation. Ranganathan and Campbell [113] identified five requirements for such environment of agents and a middleware, especially for the context-awareness support.

- Support for gathering of context information from different sensors and delivery of appropriate context information to different agents [clients].
- Support for inferring higher level contexts from low level sensed contexts [within a central instance].
- Enable agents to use different kinds of reasoning and learning mechanisms
- Facilities for allowing agents to specify different behaviours in different contexts easily.
- Enable syntactic and semantic interoperability between different agents (through the use of ontologies)

A *monolithic* agent is the Haystack client of the correspondent open-source project of MIT [161]. In contrast to NEPOMUK and Gnowsis, Haystack builds application within, whereas the formers follow an *integrative* approach of cross-linking applications and their data sources. Adapters and plug-ins [162] are techniques used to unidirectional collect and process data in a semantic engine that conveys important information towards a GUI. Both raise downsides: an adapter-oriented system is burdened to manage the heterogeneity of data structures, while monolithic systems require the applications to be programmed in their very own development environment as Sauer mann et al. argue [163]. The choice of an inference method affects greatly the level of obtrusiveness on the user-system interaction level. Segaran [152] provides several types of algorithms, some require to be trained by the user, others don't but come at the expense of less qualitative reasoning power in terms of less quickly improving their (artificial) intelligence. Training can be expected to be of similar effort and handling as for a SPAM-mail filter.

A system to be helpful requires three basic functions [164-165] when it comes down to data processing: *annotation, inference and query support*. The last is often simplified by an intuitive GUI that sends, for example, SPARQL queries towards a data endpoint and displays the filtered content. Annotations is used by any semantic desktop implementation because it provides the ability enrich existing resources like images, contacts and documents with information that can be used to create relations and - therefore - knowledge discoverable by an inference engine. Annotation is predominantly applied manually by a user. Nevertheless, some projects [166-167] do more than unidirectional data retrieval and "semantiphize" their application with graphical annotation (and task) support. It goes without saying, that a monolithic semantic desktop is advantaged when it comes to a mutual incorporation of semantic tooling.

B. Knowledge Assembling

On the way towards awareness support, the knowledge assembling has experienced highest attention among all four phases of our CCKM, as Hong et al. [168] state in their literature review, but also is the most mature. We split this phase in two; knowledge creation which supports the basics of the assembling process but lacks of techniques which allow an application to defy critics; this subsequent part we term deficiency management and is integrative towards the knowledge creation.

1. Knowledge Creation and Process-Integrated Deficiency Management

Knowledge creation - regardless if it is demand profiling, context or domain knowledge elicitation - is in most systems started with a crawling effort of available resources. However, it is also eligible to skip this

initial effort and bootstrap by listening to local sensors, delaying the valuable usage of the system as illustrated Figure 13. Retrieving this sensor data allows to loop the process and start respectively restart the reasoning about existing knowledge and new information.

Local, remote-central and remote-agents systems compete among architects' favour. Local systems concentrate on the data they observe and then try to reason about them as effectively as possible. In this process uncertainty and incompleteness appears. A local system can leverage community knowledge. Uncertainty is battled by extending knowledge structures with probabilistic or fuzzy-logic which allows direction towards Tim Berners-Lee's cited idea. Many recommendations go towards an extension of OWL [131, 133, 136] or modify rule languages [131] to make a system soft-logic capable. In general, the extensions describe a fact's probability given a certain condition, e.g. the probability a document is a contract is high, if a manager and an unknown person signed it. In this case document, contract, manager and unknown-person are concepts which require an individual to calculate the probability and create belief in the fact, that the document is a contract.

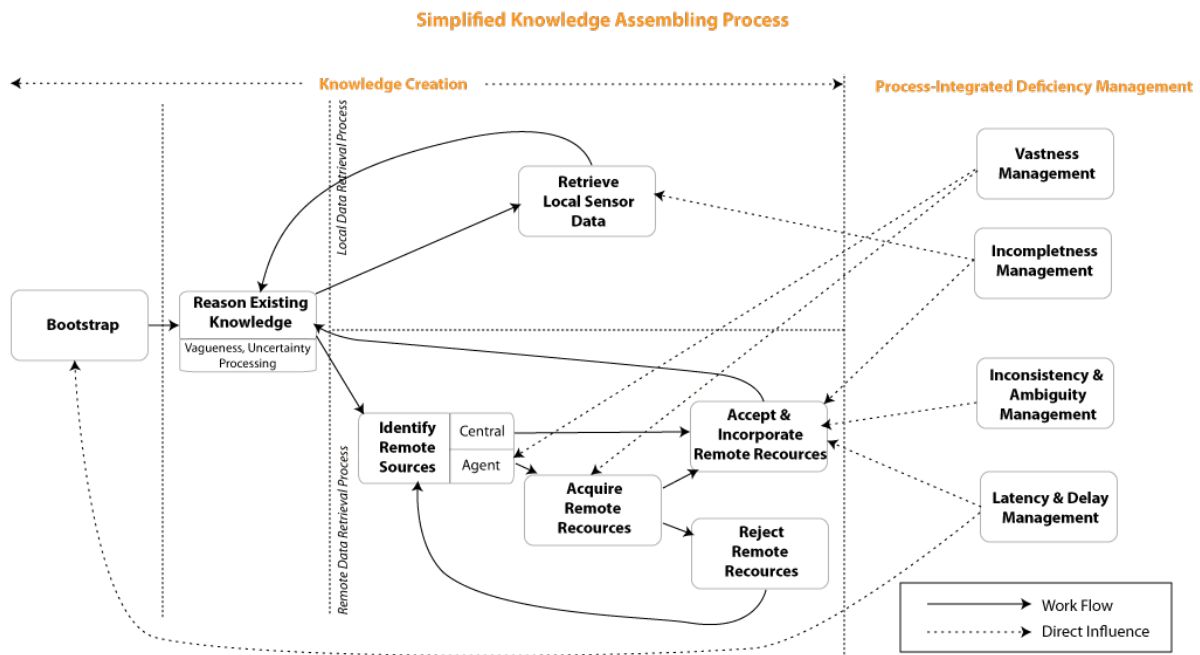


Figure 13 - The Simplified Knowledge Assembling Process Illustrates the Process-Integrated Actions against Deficiency

The early statement from Tim Berners-Lee in 1998 implies two knowledge states, that we tend to accept. In return

- we accept assumptions (when probability is high), and
- we accept assertions (if we trust the source).

From these „non-demanding“ acceptance criterion, we draw the following conclusion.

- We do NOT attempt to falsify facts (in the favour of creating information that are reasonable helpful to our user)

This conclusion aligns itself among other ideas of the semantic web, but entails numerous extra deficiencies as seen in Figure 13. Systems using central instances rely on common and internal standards – one could also say limitations in variation – with the aim of easing the management of vastness in terms

of filtering irrelevant sources and resources. In particular resources, because the entities profit from a standardized vocabulary that only needs to merge with personal mental models [159], for example, embodied in the Personal Information Model Ontology [169]. A collective concept could be pimo:Person, which the central unit as well as clients understand in common sense. These standards foster *belief propagation*, in other words, the propagation of conditional and marginal probability.

Incompleteness can be post-fought by interpolation [170-172], which describes a process of guessing the facts in a knowledge gap between two or more given facts. Interpolation is barely implemented in applications but is expected to exploit potentials for incomplete and “porous” knowledge bases.

“We remove the centralized concepts of absolute truth, total knowledge, and total provability, and see what we can do with limited knowledge.”
- Tim Berners-Lee, 1998 -

The knowledge representation OWL adheres to the open-world assumption (OWA), saying that failing to derive a fact does not imply the opposite. E.g. if the person Jane Doe is not known as individual of the concept employee, one could conclude her to be an individual of the concept contractor – but it is unknown respectively undefined. A system of closed-world assumption (CWA) would decline any speculation and deduce “Jane Doe is NOT a contractor”. From this we understand that not getting an answer from a semantic retrieval system, results in the claim “false”. The NEPOMUK Representational Language [159] builds on a *close-world assumption to pre-emptively cope with local heterogenic knowledge interpretations* – it upholds OWA for environment-foreign reasoning. However, applied interpolation and handling of incompleteness is still a recognized gap in research. “OWA [...] hinders computability and largely increases the complexity of RDF data. [...] while CWA is much more error-prone.” as the NRL Specification [173] outlines.

Inconsistency and ambiguity is probably the most challenging deficiency and most critical of all. This is because from a contradiction, a type of inconsistency, anything can follow, which is referred to as the principle of explosion in classical logics.

$$\{\emptyset, \neg\emptyset\} \vdash \psi \tag{4}$$

If a knowledge base claims both ϕ and $\neg\phi$ to be true and not true, anything could be derived. It is infeasible reasoning, better known as deductive reasoning, which does not allow the claim to be debatable. In contrast, it is different if probabilities are proportionally correct distributed to the claims of facts ϕ . Other non-demonstrative methods are inductive, statistical, abductive and paraconsistent reasoning. A rather simple way to counteract inconsistency and incompleteness in OWA is to create preferences among rules as shown in Example 1 - Preference Rules Allow to Handle OWA Incompleteness and Inconsistency . It’s called rule inheritance or rule of specificity, because one rule is more specific than its general antecedent.

(R0) if p then (defeasibly) $\neg q$	e.g. if contractor then \neg hasEmployeeNumber
(R1) if r then (defeasibly) q	e.g. if co-worker then hasEmployeeNumber
(R0) if p then (deductively) r	e.g. if contractor then co-worker

Example 1 - Preference Rules Allow to Handle OWA Incompleteness and Inconsistency

Notice, no popular reasoning engine implements defeasible reasoning, although there are a number of implementation recommendations [174-176]. This is also due because it is difficult to create the rules, even for users which are familiar with a domain. Also, applying CWA requires a consequent processing of incomplete and inconsistent knowledge on the way to a stable knowledge base. That is ambitious for collaborative environments as this paper focuses on.

Another approach to manage inconsistency is paraconsistent reasoning. It is based on the idea of supporting inconsistency-tolerant systems that process a world of likely unconfirmed, interpreted, abstracted, incomplete and incorrect knowledge - as the world wide web. The first research efforts are fairly new [177-178], but it can be expected to increase due to the patterns that heterogenic and remote environments.

An insufficient covered issue is the delay of building a solid knowledge base. Reseraches take into account, that some logical algorithms, in particular Bayesian networks, require training time and therefore get smart with a delay. The fight against latency of knowledge creation and the often subsequent obsolescence of knowledge is also not discussed in literature.

Not considered a managerial deficiency task is vagueness and uncertainty handling. They are not simply process-integrated as the others; they literally shape the knowledge models. These are tasks that cannot be done after a system is developed. They require an engrained technical implementation into the basics of an architectural concept [94, 130, 133, 136, 179].

“Deficiency” emphasizes the negative aspects that show up during the live-time of a semantic and ontology-driven - and especially of disparate - system. When research progresses towards the social semantic desktop and sustainable mechanisms, deficiency can be replaced by “efficiency” and probably these managerial task will move to where vagueness and uncertainty is handled in Figure 13.

2. Sources for Knowledge Creation

Holsapple et al. [5, 25] mentions two major groups on how to generate knowledge: discovery and derivation. Chapter Context & Demand Formalization emphasized the potential of sensors to discover information, but not particularly knowledge. Within an enterprise environment social knowledge discovery channels like tagging, wikis, and collaborative decision-making are emerging. Derivation, which is rather a complementary group, that relies on discovered knowledge and ought to extrapolate existing knowledge pieces. This idea does mainly comply with our Figure 13, which we want to keep in mind together with the precise recommendations[5] of Holsapple et al.

C. Establishing Trust & Accuracy

If we recall Figure 9, we notice the Trust & Accuracy element of the CCKM goes out to logic, proof and trust. Logic is required to prove assertions, because at some point a user will ask: Whom do I trust? Where is this data from? How do you reason? Most logical methods as we introduced in earlier chapters can provide the latter questions. But transparency stops for the former. However, it is denotative that new books like The Social Semantic Web [180] dedicate just about two out of 300 pages to security and policy. Trust remains a faintly explored field, and therefore accuracy of knowledge is often criticized by

consumers. In an enterprise environment we – of course – face extenuating circumstances because internal information sources are more trusted than in the World Wide Web.

Trust can be established by pulling transparency further to the trustworthy identification of a resource. Dan Connolly et al. [181] suggest *rule-based policies*. Besides “annotating” the resource and identifying user with the full URI, a check-sum is added to verify that the data was not maliciously compromised. Not only a trust-verification system will employ the URI to check the integrity, the URI is simultaneously used to see if the resource springs from a trusted source – of course also store in a local repository. The method used to declare a source trustworthy can be achieved manually entering sites, employing PKI or relying on a social group judgment. E.g. intranet.alpiq.ch is manually added, a certificate for intranet.alpiq.ch is added, or employees mark intranet.alpiq.ch as trustworthy in a folksonomy-manner, and thus can be stored in the enterprise repository. Grading of the level of trust can be accomplished for all of them – on a personal mental as well as on enterprise repository. Rule-based policies allow a semantic system to remain fully automated. It leverages the same logic power used in the knowledge creation process. It uses organizational knowledge; that is to say hierarchy, roles, delegation, etc., to comply with rule-based policies, which in turn decide on proof and trust. It is up to the architect and user to reject or accept resources. If policies are formally defined in a governance repository, a system could also autonomously execute those policies.

Olaf Hartig introduced tSPARQL [182], a trust-aware SPARQL version. In this approach he enriches individuals in a semantic database with a precomputed probability value that can be joined with other data computed by a trust merging function. The query itself is extended with a selection command that defines the range of acceptable trust. It is a promising and fast technique of trust verification executed by a database engine. It is proven faster than a rule-based engine, when query-processing is rewritten.

D. Information and Knowledge Delivery

Recalling the title of our thesis “awareness stimulation” is an objective we set. Although awareness is considered a secondary or supportive action [107], it is crucial in vivid and quickly changing collaborative work in sequence to up-shift the user’s cognitive energy to more complicated tasks – as mentioned in the introduction. We have thoroughly observed context, because context is our main “listener” – if you will – to appraise states of various kind and important to our collaborative environment.

From this point, what did research contribute to the system architecture in terms of guidance on awareness stimulation? How does theory inspire to handle the delivery of awareness and therefore knowledge? Other questions are listed in chapter I.B.4.

Knowledge-Based Networks [183-184] is one of few serious ideas that is considering a reasonable knowledge delivery service that is capable to “distribute heterogeneous knowledge to, and only to, nodes that have expressed an interest in that knowledge”. This is achieved if a consumer marks ontology concepts and individuals and subscribes in the network for updates. A P2P and a centralized network likewise can store these subscriptions and try to match the subscribed classes with their own and common concepts, in upwards or downwards directions. That is to say, searching for updates of more general or specific concepts. Furthermore, this research area strongly overlaps with ontology mapping

efforts, because it also relies on identifying semantically similar or equal concepts. Knowledge representation language elements like owl:equivalentClass elements are key.

Other questions not researched in detail is how to alert knowledge; whether new incoming knowledge is to retain until demanded of the consumer is given according to spatial and temporal conditions; whether proof and trust should be delivered for continuous process etc.

A research gap is the non-autonomous prescription to KBNs. Researcher still expect it to be done manually, although such a subscription could – at least theoretically – be derived from a personal mental model [163] or a formalized demand description as we highlighted in chapter II.A.1.

E. Summary: The Research-GAP & the Significance of the Study

Many gaps of applied research were closed by the NEPOMUK [159] and Haystack [161] project. Following our CCKM we still see potential to discover and acquire logical sensors that not simply create knowledge but also help to understand the user's demand. CCKM step 1 and 2, are partially overlapping to its requirement for hard and soft logic [94-95, 125, 128, 130, 172, 185-186] and thus the gaps remain the similar: the challenges of knowledge uncertainty, vagueness, incompleteness, inconsistency and ambiguity. Step CCKM-Knowledge-Assembling lacks often from knowledge interpolation [170-171]. Sometimes it requires little human reasoning power to create another valuable fact, but machines don't spot the bridging element.

Trust & Accuracy investigated in CCKM is a comparatively unmarked area of research and only little literature is available. Further investigation of rule-based proof and trust mechanisms [181] have to be conducted and findings implemented and need to compete with database-oriented approaches as tSPARQL [182].

Knowledge Delivery, as last element of the CCKM, is missing ideas of non-obtrusive knowledge dispatching. That describes ways of how to limit the distraction of the user by acting dominant as an application.

This master study ought to cope with the identified gaps and give appropriate and practical answers to them. In focus for the prototyping are the gaps of *CKMM 1 and 4*, partially 2. CKMM 2 is of a magnitude that goes beyond what this thesis can bear to enlighten awareness stimulation. CKMM 3 deserves less attention considering our thesis objective.

The massive sections of introduction and literature review will intentionally end up disproportional oversized compared to the core section – also because it is the obligation of the literature review to identify the optional capabilities, which a system architect could incorporate in order to present rocket science. A case study in the timely magnitude of MPhil or PhD study must certainly address these extra capabilities – this could fall within a subsequent project within the research area of SECS but a different focus. However, it is our intention to leverage the literature review and “*only assemble*” the essential parts from *CKMM 1 and 4*, which will significantly reduce the text in length within the body section, IV. Semantic Information Awareness Stimulation. Within this major section we will back-reference the previous sections, in particular the literature review, to justify our architectural decisions.

III. Research Design

At this point our thesis objective, research efforts and gaps are clear. This main chapter will outline the design of our own research, which will contribute to theory but primarily to applied science and business benefits. The research design consists of the *selection and justification of the selected research method*, the *application* and the *limitations of the method*.

A. Research Method Selection and Justification

Our thesis objective requires an inductive research approach, because we claim to be able to stimulate awareness within an enterprise environment, applying semantic technologies. The outcome is expected to support this tentative hypothesis. Another reason not being able to apply a deductive research methodology is the fact that examples for testing rely on specific organizational activities. This entails high expenditures to simulate a chosen scenario within a single *observation* object e.g. to simulate awareness in a concrete enterprise scenario. Thus, we select the method of case study.

The study is supported by prototyped software, which represents the technical core environment where scenarios can be applied and *patterns* recognized. The assessment is conducted together with specialists of a chosen business environment, which compares the results against a previously created criterion list. We want to clearly articulate our assessment objective: We observe the collaborative business benefits that result from awareness support – and not technical behaviours – and test them against a set of criteria.

The set of criteria are elaborated within the body of this paper, in the chapter case analysis. It will supply the criteria, which shape the *tentative hypothesis*. From the given business requirements the software requirements will emerge. After the development process, first, selected business cases will be tested together with the business focus group using the exploratory prototype; subsequently compared with the predefined criterion list. The criterion performance assessment is the basis to conclude on the upraised tentative hypothesis and the prototype's success.

The hypothesis is considered tentative, because the result of the assessment cannot be justifiably extrapolated towards other complex businesses. As a fact, the exploratory prototype does not implement influential enterprise *concepts* like governance, security, integration of business processes policies and workflows. All of them can significantly affect the criterion analysis and the assessment. Thus, *propositions* of these concepts are also neglected. However, the only tendency we want to prove and measure is the stimulation of knowledge awareness, a proposition, as mentioned in our thesis statement of the introduction. For this matter, a tentative hypothesis is sufficiently abstracting the concepts and propositions for our theory. Also because, the nature of the tentative hypothesis is to give directions an to discover tendencies, rather than clearly rejecting or accepting the hypothesis statement.

- Concepts
 - Semantic Sensors
 - Knowledge Supply
 - Information Awareness
 - Enterprise Co-Workers
- Propositions:
 - Stimulation of Information Awareness by Semantic Sensors
 - Knowledge Supply to Co-Workers on time

Nevertheless, the *theory* concluded from the evaluation of the tentative hypothesis is not certain and thus rather an educated guess, which allows only a vague conclusion and a statement of the research’s direction. On the other hand a case study as we apply it, can be verified repetitively whereas the deviation of findings are little, if criterions are thoroughly elaborated.

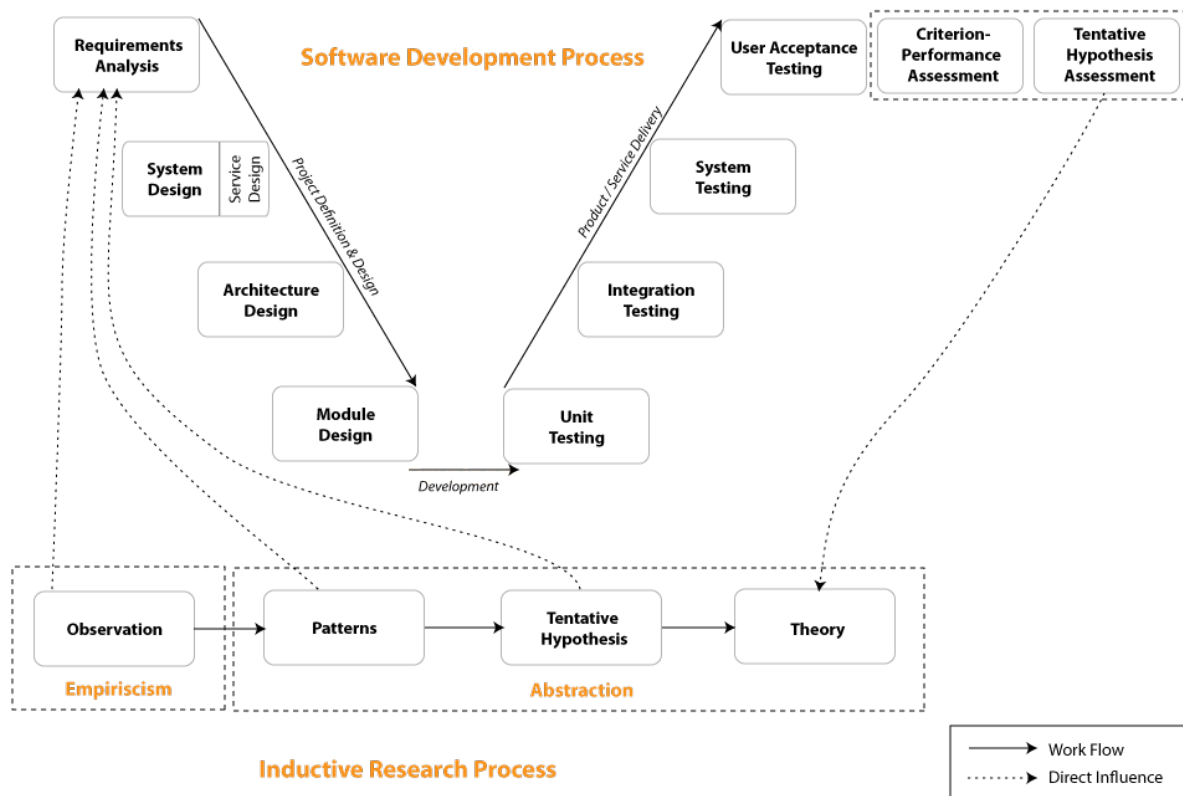


Figure 14 - Inductive Research Process Embracing the V-Model Software Development Process

Figure 14 illustrates how the inductive research process influences the requirements analysis, which defines what needs to be developed in order to test and confirm the tentative hypothesis and draw a conclusion in a final abstracted theory.

B. Research Strategy – The Case Study Applied

The research type case study was selected because it is “useful in the preliminary stages of an investigation” as Abercrombie et al. [187] argues. Considering our objectives, a case study complies with the meta-goal to find out if it makes sense to continue the research for semantic enterprise collaborative support (SECS) with awareness stimulation. The selected type of case study is paradigmatic, because we assume that for all cases we observe, we experience similar patterns, concepts and propositions.

Recalling Figure 11 and Figure 14, one finds the body of our study that is split in three chapters. The first chapter is simultaneously the start of the use case. The situation analysis will educe a series of use cases by interviewing people from the focus group at Alpiq. The observation objects, which are business tasks with deficiencies, are required to exhibit the same patterns. Having determined a case, a criterion analysis will be sets up a list of criteria that needs to be fulfilled by the new exploratory prototype and the case. In other words, the focus group at Alpiq defines the required outcome, it expects from a semantic system in order to consider it valuable in a specified metric. From case selection we define a tentative hypothesis which, of course, leans on the thesis statement. The empirical observation, the abstracted patterns and tentative hypothesis are then conveyed to requirements engineering phase, which is essential because it defines the business and derived technical specifications. When the software leaves the V-model development process the user acceptance testing is conducted to ensure the conformance of the submitted spec. That is to say, that the prototype is acceptable and ready to test against the criterion list. The criterion assessment is the fundament to appraise the tentative hypothesis. It follows that if the hypothesis creates sufficient tendencies – not logic evidence – that a theory can be form which justifies or rejects it our thesis statement. This last step closes the research circle.

C. Methodological Limitation

The theory about case studies couldn't be any more controversial as Bent Flyvbjerg brings forward in his publication [188]. A persistent claim, however, is common to all understandings: one cannot generalize from a single observation i.e. a single case. And this is what it is about in research. Observations in big numbers are used to generalize, to abstract in order to apply the concluded theory on other occurrences. In detail "apply" means, that one can make repetitive and accurate statements about any occurrence of the previously observed object and its selected properties. A case study limits its extent to a single occurrence of an observation object. In contrast, proponents argue with the precious intuition of a (wo)man, that can rule out several occurrences not helpful to observations. E.g. a business process that was created by coincidence and very likely would not occur again. Also it often applied for pilot studies, because no occurrences are to observe.

D. Summary

Ultimately we say that case studies are detailed examinations of single but complex situations, which is applicable in our research engagement. Another argument backing-up our research-type decision is that theory and technology of the core of semantic has evolved to a solid state. For the case study this means, we can make precise statements on the excrescence of the case study, in particular about how volatile the examinations could become. For a negative example, it is totally unacceptable to propose a case study to the examination of a football game. Why? Because the noise, disturbance parameters, are volatile and therefore the outcome of a sport game cannot be predicted. In contrast, a system like ours can – repetitively and with low deviation in outcome.

As a final statement: we test our prototype against "fit for purpose" – and not "fit for use".

IV. Semantic Information Awareness Stimulation

This section is our very own contribution to the research area of semantic collaboration support with the focus on instant awareness stimulation. As outlined in the Research Design in Figure 14, empiric observations in the representation of use cases are conducted within a situation analysis. The purpose is the extraction of criterions and the shaping of a tentative hypothesis that both influence the requirements engineering vital to the success of the exploratory prototype. The second chapter within this section describes the prototype’s architecture in detail. It sequentially addresses the challenges of our CCKM, with a strong emphasis on CCKM 1 and 4. In the section’s last chapter we will non-verbosely describe the success of implementation from a business perspective, followed by a user acceptance testing together with Alpiq AG that will – if positive – allow us to continue with the criterion-performance assessment. This assessment is fundamental to the verification of our previously tentative hypothesis from which one can draw the final conclusion about the thesis statement, case study and the success of the project.

A. Case Analysis

1. Situation Analysis

Alpiq AG, our research partner company, selected three use cases. The originals submitted are printed in Appendix A – Examples of Theory and Implementation, but are additionally illustrated in Figure 15 in UML2 compliant notation.

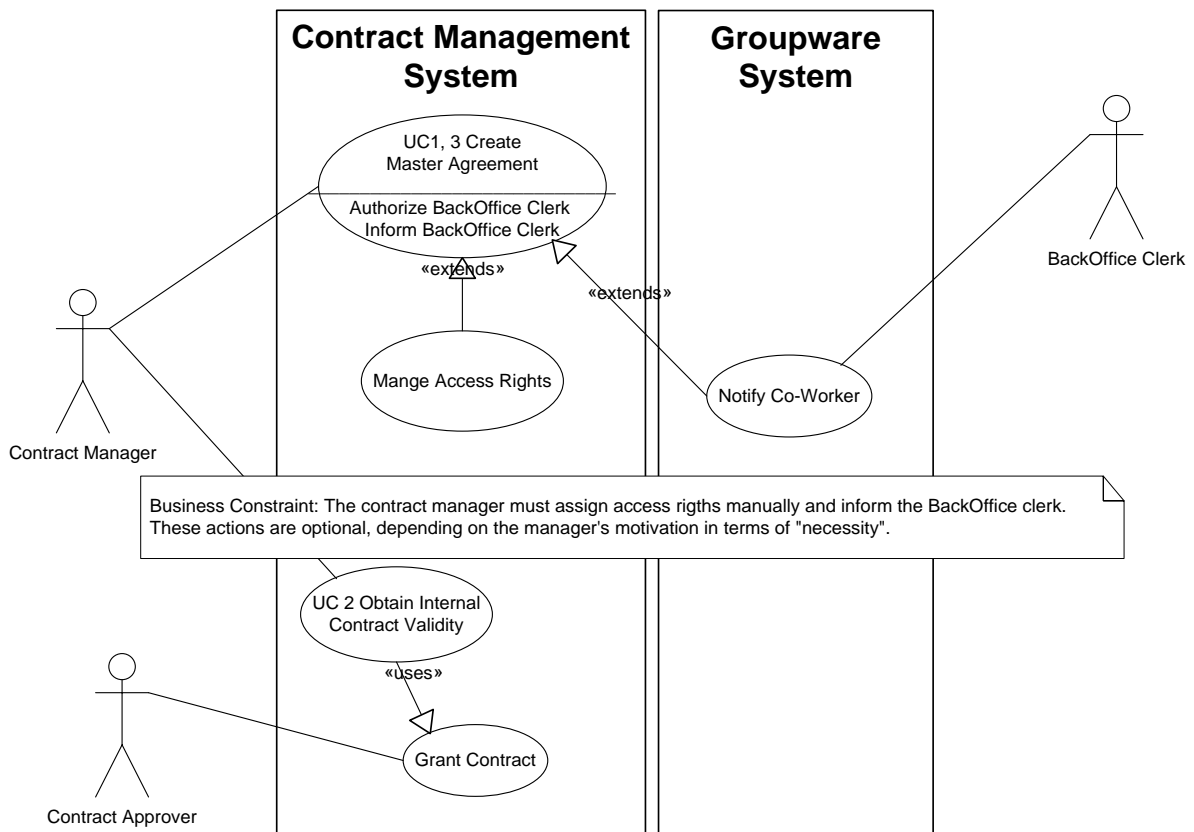


Figure 15 - Three Use Cases featuring Alpiq's Contract Management System (VMS)

Alpiq appreciates their knowledge workers' individual approaches and how they get their work accomplished. From this follows, that they do not allow workflow systems to restrain tacit knowledge and impede effective and efficient operations. The same applies in sections of contract management process.

Use case 1 describes the task of having uploaded a new master agreement by a contract manager, and the need to *share* it with a limited number of clerks from the back office. Authorization is currently assigned manually by the contract manager according to internal and - inconsistently stored but - available policies.

Use case 2 describes activities from a contract manager and a contract approver, while the former will ask for the *contract validity check* from the latter.

Use case 3 describes the same initial task but also requires *informing the back office* clerk. Notification is predominantly executed by groupware software and not the contract management system.

a) Criterion Analysis of Business Maturity and Collaboration Support

We want to lean against the popular recommendation for maturity and quality integration, the Capability Maturity Model Integration (CMMI) approach. This is for the purpose to *identify criteria and criteria of different organizational evolution levels* that facilitate efficiency and subsequently competitiveness as denoted in section 1.

All three use cases are expected to be supported by the exploratory prototype. IT-supported means, that although there is a high deviation of how tasks are executed by individuals, the individual deviation of task execution should not, and neither be *unpredictable nor uncontrollable* in terms of the CMMI Maturity Level 1 specification [189]. "High quality and exceptional performance is possible as long as the best people can be hired." is a universally true statement for the ML 1 and implies that the prototype must support an employee to act "heroic" or at least highly professional in his ad-hoc and individually processing tasks. The outcome of a use case must be predictable by a supervisor.

On maturity level 2 work products must be documented, e.g. the "master agreement document" in UC1. And at this stage, access management is already embedded in a life cycle. More important is that business requirements are defined - but not processes and policies. Nevertheless, ML2 activities are still mostly *reactive*, e.g. in UC1 a back office clerk needs to remind the contract manager to notify her about newly created master agreements.

ML3 is a pro-active state, which tackles porous definitions and process, and missing instructions, but is *unable to measure the execution*. ML3 requires actors to follow established organizational policies and processes - regardless whether they are automated or informally expressed. Actions must be traceable and therefore create management visibility.

ML4 is dedicated to organizational measurement and controlling. It overcomes the deficiency of non-predictive outcomes that allow the management to make transparent and reasonable decisions. However, the methods apply are *not sufficient to continuously optimize operational efficiency*.

ML5 is state-of-the-art in organizational management and allows selecting improved processes and promising innovations. This level often embeds continuous service improvement (CSI) programmes that

employ quality improvements methods like SixSigma, Deming Cycle, FMEA, etc., which a can measure process variations.

Having recalled our knowledge of CMMI, we can say that the maturity levels represent the quality levels and the requirements for the levels represent the criteria we looked for. The next sub chapter will allow us to create an overview of criteria and quality levels that can be projected on our defined use cases. Before we do so; let us reminded another influencing aspect from the introduction section: the collaboration feature. We declared collaboration support as a target in Figure 7 of chapter I.A.3.

b) Criterion Analysis on Use Cases

Below is an overview of the three use cases, maturity levels and criteria of the levels. This means at the end of the project we do not simply test for accomplished or not, but also value the accomplishment that will give us a more granular reasoning on the assessment of the prototype appliance and our tentative hypothesis – all *from a business perspective*, rather than a narrow technical view. Simultaneously it is also the reason why we reject to assess technical aspects like the elements from deficiency management described in Figure 13, because no matter how sophisticated deficiencies are solved, the quality of application service determines the prototype’s success.

The defined criteria are upwards-recursive requirements, derived from the CMMI definitions [189] and depicted as follows.

ML1 can despite its chaotic attribute “initial” be supported by the prototype. The outcome of a service can be at least predictable (OutPre) – however, not the detailed procedures. A system can also support the ad-hoc tasks (AdSup) e.g. by recommending to align executed tasks with previously observed processes, or more general, align any activity that has a relation of any special kind.

Product and service requirements (ProdReq), are advantageous to know, because it allows to identify and allocate resources (ResAll). In ML2 for example, a competent co-worker or knowledge artefact could be acquired to improve task execution to meet a service quality standard.

Maturity Level	Use Cases					
	UC1 - Sharing		UC2 - Validating		UC3 - Informing	
	Maturity Support Criterion	Collaboration Support Criterion	Maturity Support Criterion	Collaboration Support Criterion	Maturity Support Criterion	Collaboration Support Criterion
ML1, initial	<input type="checkbox"/> OutPre <input type="checkbox"/> AdSup	<input type="checkbox"/> CoEx <input type="checkbox"/> Comm <input type="checkbox"/> Coord <input type="checkbox"/> Coop	<input type="checkbox"/> OutPre <input type="checkbox"/> AdSup	<input type="checkbox"/> CoEx <input type="checkbox"/> Comm <input type="checkbox"/> Coord <input type="checkbox"/> Coop	<input type="checkbox"/> OutPre <input type="checkbox"/> AdSup	<input type="checkbox"/> CoEx <input type="checkbox"/> Comm <input type="checkbox"/> Coord <input type="checkbox"/> Coop
ML2, managed	<input type="checkbox"/> ProdReq <input type="checkbox"/> ResAll		<input type="checkbox"/> ProdReq <input type="checkbox"/> ResAll		<input type="checkbox"/> ProdReq <input type="checkbox"/> ResAll	
ML3, defined	<input type="checkbox"/> PolEnf <input type="checkbox"/> ProdVer		<input type="checkbox"/> PolEnf <input type="checkbox"/> ProdVer		<input type="checkbox"/> PolEnf <input type="checkbox"/> ProdVer	
Instant Awareness Support	<input type="checkbox"/> Socially-aware <input type="checkbox"/> Task-aware <input type="checkbox"/> Concept-aware <input type="checkbox"/> Workspace-aware		<input type="checkbox"/> Socially-aware <input type="checkbox"/> Task-aware <input type="checkbox"/> Concept-aware <input type="checkbox"/> Workspace-aware		<input type="checkbox"/> Socially-aware <input type="checkbox"/> Task-aware <input type="checkbox"/> Concept-aware <input type="checkbox"/> Workspace-aware	

Table 5 - The Multivariate Criterion List is the basis for testing "fit for purpose" for Alpiq’s use cases.

Policy enforcement (PolEnf) is probably the most important requirement in ML3, and fosters business and product compliance, which is a post-condition to allow product verification (ProVer) to return a customized and meaningful feedback.

ML 4 and 5 are about strong managerial support and focus. CMMI documentation states “Quality and process performance is understood in statistical terms.” and is therefore not applicable for the research type of a case study, which relies on the feedback of a focus group and not statistical analysis. In other words, we test “fit for purpose” and not “fit for use”, where the latter is a rather quantity-performance test as ML 4 and 5 require it.

In regards to enterprise collaboration depicted in chapter I.A.3, we want to integrate support for co-existence awareness (CoEx), communication (Comm), coordination (Coor) and cooperation (Coop) and thus we employ them as boolean criterions as seen in Table 5 as we test the use cases in the criterion performance assessment.

The last row in Table 5 adds awareness criterions to our multivariate criterion matrix. The types of awareness from Gutwin et al. [107] is accepted. We do so because awareness stimulation is our research focus and if the prototype complies with those criterions the assessment will result positive, and furthermore the tentative hypothesis formulated in the next chapter can likely be confirmed. The types of awareness support must be implemented in at least once of the collaboration support elements, which facilitate the maturity levels.

2. Tentative Hypothesis

Based on our criterion matrix that emerged from the introduction and literature review and keeping the thesis statement in the back of our minds, we claim the following *tentative hypothesis* is true and can be verified. It is formulated from a business perspective.

«Instant awareness support significantly fosters business maturity levels and is driven by knowledge-sharing excellence.»

It is pre-conditional to verify that the awareness stimulation is triggered from at least one of the collaboration support features. If we would refuse this verification, the claim of our tentative hypothesis is too vague because awareness support could be affected by coincidental inputs or pure interpretation by the user, instead of substantial knowledge. Therefore we formulate a *tentative sub-hypothesis*.

«Collaboration elements are actively and significantly driving awareness support. »

The adverb “actively” emphasizes the need for progressive actions of the prototype, in contrast to reactive information delivery, which besides is a characteristics of maturity level 3. Manual information-fetching is a even worse indicator for non-progressive collaboration features. From a technical view, progressive actions are mostly triggered by sensors.

It is understood and agreed, that in the last chapter of the section, the focus group of Alpiq AG will verify if the maturity criterions are accomplished and to what degree. Before this assessment, we will verify the sub-hypothesis for validity in a test applying one predefined use case. If the latter and subsequently the former hypothesis are confirmed to be true by trend, the prototype can also be functionally accepted.

3. Requirements Engineering

Since development is about a prototype the requirements are limited to the functionalities expected to support parts of the three use cases. The three use cases are merged in a process landscape in Figure 16 with different actors, respectively responsibilities. Next to the actions remarks are placed, which indicate the need for special treatment, so called functional knowledge.

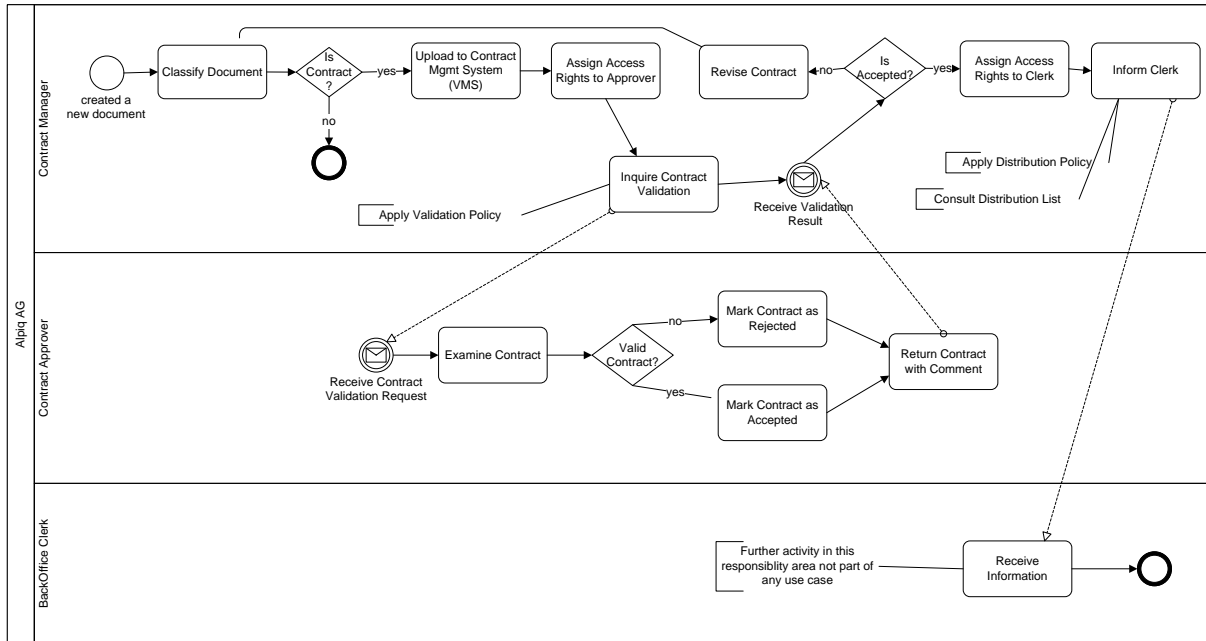


Figure 16 - The 3 Use Cases Packed in a Business Process “Contract Creation”

It is the business process that brings light over potential improvements and promising appliance of collaboration support – which is our sub-hypothesis that testifies the appliance on Use Case 1.

Selected Use Case		Business Service Requirements	Application Service Requirements		Technical Service Requirements	
#	Activity		Functionalities	Properties	Functionalities	Properties
UC1	Classify Document	Coop	Identify change. Client to suggest upload.	Document is “contract” and status is “Final Draft”.	Context elicitation by local document monitoring. Identify the business mission. Know what to do next from previous and similar missions e.g. Upload document. Know VMS URL.	Constantly monitor properties of an open document.
	Upload to Contract Mgmt System	Coop	Client to suggest URL to VMS		Offer probabilities of similar tasks in the past.	
	Assign Access Rights to Approver	Coop, Co-Ex	Web Client System to suggest user to assign access rights.	Only co-workers, we can recommend with a high probability to be in a approver role.	Offer probabilities of cooperating users in the past. Have access to a user directory or similar. VMS to assign suggested user rights.	
	Inquire Contract Validation	Coop, Co-Ex, Coor, Comm	Find and suggest contract approver for contract validation.	Suggest only available contract approver.	Offer access to roles and availabilities of users. A notification system with a user directory or similar.	

Table 6 - Deriving Application Service Requirements from Requirements of Promising Business Service

Table 6 extracts the core business service requirements (BSR) of functionality and properties, and derives the application service requirements. Table 6 does also provide technical service requirements (TSR) that are directly derived from the application service requirements (ASR). TSRs define the technical functionalities and properties that an architecture system must supply in order to fulfil ASRs and subsequently the BSRs. These TSRs shape the architecture, patterns and system functionalities as explain in the next chapter.

B. Architecture and Implementation

Once again, our Compact Chain of Knowledge Management (CCKM) will help to keep our focus during development. The system landscape will describe the systems we need to satisfy the TSRs which we previously identified. From this point on forward, we design the context model we need within such a system; a system that predominantly retrieves its data from ontologies. These ontologies are what we need to model. Afterwards we decide on a single exemplary sensor that actively supports collaboration especially for instant awareness support. This is all within the frame of CCKM-1.

The Knowledge Assembling element of the CCKM will illuminate how all our designed elements fit together and how they processes and assemble information. CCKM-3, Trust & Accuracy, is where we only try to propose a simple limited trust indicator, which will not be implemented in the prototype. Awareness Support in CCKM-4 enlightens how we convey collaborative information to the user without being obtrusive. This chapter includes also brief discussions about GUI decisions. Issues mentioned in the problem statement of chapter I.B are tried to be solved with existing knowledge gathered from the literature review in section II and own methods or resolution.

1. System Landscape and Deployment Diagram

The prototype is designed as an agent. An agent's major characteristic is its autonomous existence [123, 155] and so is the SECS Desktop Client. This entails that the client can operate independently without a central instance. Being autonomous doesn't necessarily restrain the capability of retrieving previous information that together with existing knowledgebase can infer new knowledge. This is an option that is planned for the SECS Desktop Client but not implemented in Version 0.2.43. It requires a SECS server as illustrated in Figure 17 in a dashed device frame.

The client is developed and tested on Java 6 platform, incorporating several foreign libraries and using Swing components as user interfaces.

Semantic Information Awareness Stimulation

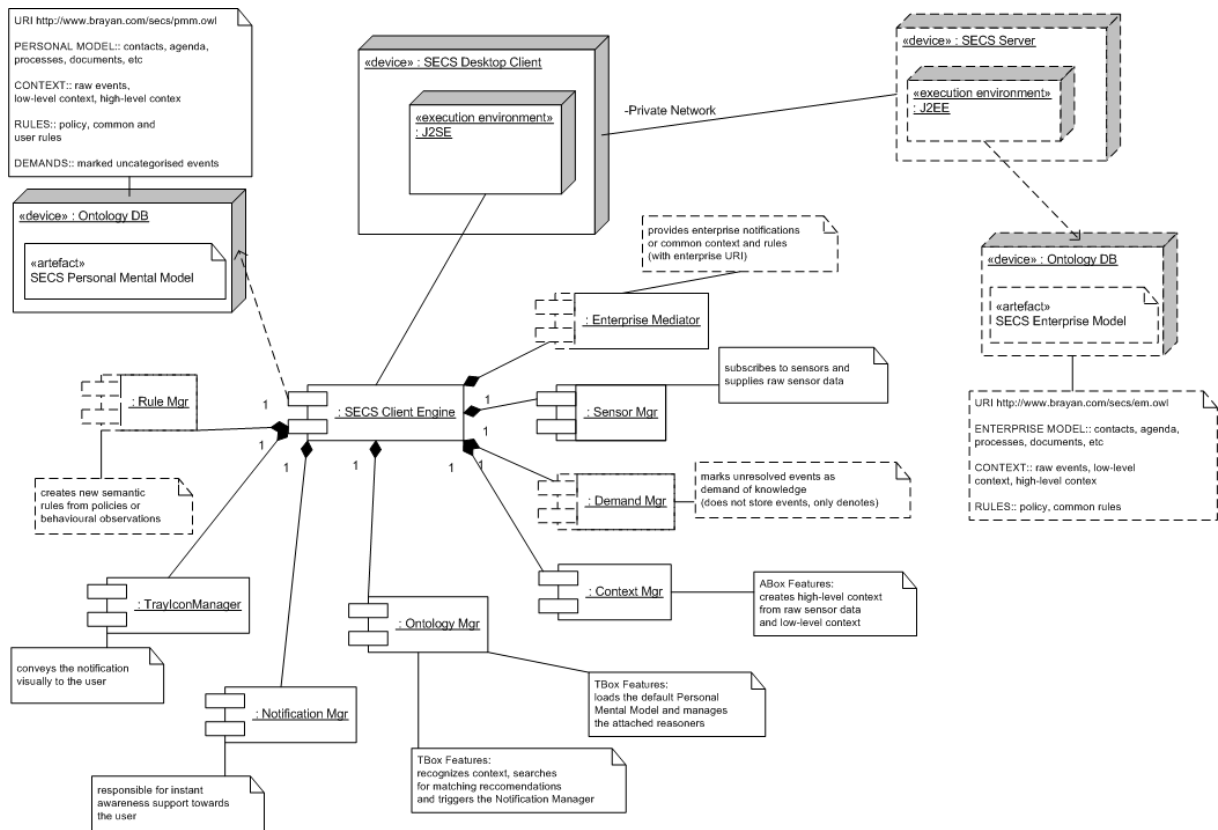


Figure 17 - SECS System Architecture with developed and planned (dashed) components

The Documentation of the implementation is generated by JavaDocs and listed in the Appendix A – Examples of Theory and Implementation Documentation. Below a brief overview of the most important components of the n-tier architecture, excluding the GUI classes which do not contribute to the logic applied in the program.

Ontology Manager invokes the Jena framework library. The manager loads the SECS Personal Mental Model using the JENA framework library. In SECS 0.2.43 the PMM is loaded into memory and not to a persistent endpoint. The manager also attaches a generic OWL reasoner which is to incorporate the SWRL rules that are either hard-coded or supplied by the Rule Manager, expected to be implemented in a later version. The models reside in a base, inferred model and deduced. While the most former represents the pure data, so-called resources, the inferred model stores reasoned information. The deduction model maintains new inferences triggered by the rules. The Ontology Manager is implemented as Singleton pattern.

Sensor Manager starts and stops all sensors the manager and user subscribe to as shown in

- . The manager picks up events invoked by these sensors. The passed object is of type SensorEvent. Every sensor tries to populate as much data in the event object as possible. The manager itself adds additional data that are retrieved from local sources like the properties file. The Sensor Manager is implemented as

Singleton pattern. Sensors use the Observer pattern, which allows the Sensor Manager or Sensor to register to an observable sensor.

- Context Manager receives a complete Event object and inserts it to the base model. The manager this required to know the context elements e.g. identity, time, location, activity, etc. In a later version, the Context Manager is expected to enrich local events with enterprise information. Either to state event date more precisely, combine or to discredit. E.g. Retrieving a city's name by a IP-location search might be misleading, because in NAT-networks there's often only a single IP exposed to the internet and that might be spatially different from where the user resides. The Context Manager is implemented as Singleton pattern.
- Notification Manager prepares and coordinates the notifications towards the user. Notifications are created by rules within the Ontology Manager, fired upon SWRL forward-chaining supplied with antecedents and consequents. The Notification Manager is implemented as Singleton and Observer pattern. Any component can register to the object – so does the TrayIcon Manager, which user-friendly forwards notifications to the tray bar. The object passed between these managers is of type Notification.
- Demand Manager is a planned class of a future release. It is supposed to employ soft-logic in order to form demands from all the events captured by the Sensor and Context Manager.
- Rule Manager is a planned class of a future release. It is supposed to ease the maintenance of the rules and allow the automatic generation of rules by analyzing the demand. The Ontology Manager should retrieve the rules from this object.
- Enterprise Manager is a planned class of a future release. It is supposed to plug the enterprise knowledge to the SECS Desktop Client. Events, Context, Rules and Notifications should be exchanged through this manager.

The following chapters will describe how these components and classes interact outlined by the CCKM, while keeping earlier identified problem statements of I.B in the back of our mind. The explanations are based upon an event extracted from our simple business process in the Figure 16: *The Contract Manager classifies a document as a contract. This activity, which rather occurs as an event in real live, should be succeeded by a helpful notification. The notification should prompt the user to inquire a contract validation with the responsible person.* This example fits perfectly as verification example of our thesis statement. It is about the Use Case 1 and comprise of the activities “Classify Document” and “Inquire Contract Validation” as listed in Table 6.

If we give it a thought, the difficulties raise in a pile. How do we recognize the change? How do recognize the correct change and not just any? How do we know what task comes next in the process? How

compulsory is that next task? How do we know who to contact? Are we positive to have the right contact for our location? How do we contact that person?

With a bit more effort one could come up with additional questions in regards to challenges one faces during the support of this business task.

2. Context & Demand Formalization

In this sub-chapter we explain the model’s most essential entities and why we have elected them. If analysing the ontology model you will discover others resources and properties (predicates) fostering completeness, granularity, extensibility and avoid redundancy, as demanded by Krummenacher and Strang [83].

Four context elements model our context. Notably it is: Identity, Location, Activity and Date. While the first three are conceptualized the Date is a simple literal denoting the date and time of the occurred event. An instance of an Event owns the predicate `hasIssuer` which marks an identity. An identity stands for any object that autonomously can create an event. For Example a user, a role, a computer but also a organization. This flexibility in value range allows sensors to listen to nearly any object that could provoke an event. The Location concept is divided in a logical and physical party. Logical is primarily about network attributes like an IP address, while room, building, city, region and so on represent physical locations concepts that can be assigned to describe where the event occurred. A location can be easily attached to an Event by the `hasLocation` predicate.

We see the instances of the concept Event as historic records that do not appear as duplicates. How could they, since time is stored in milliseconds. Therefore, we use the predicate `isConceptuallyPartOf` and the inverse predicate `isConceptuallyComprisedOf` to mark only the Activity’s and the Context’s relationship, decoupled from Date, Identity and Location. This way an Event’s activity could be multiply part of a context. The technique how we establish the link between individuals of these two concepts as seen in Figure 18, is subject to the next chapter.

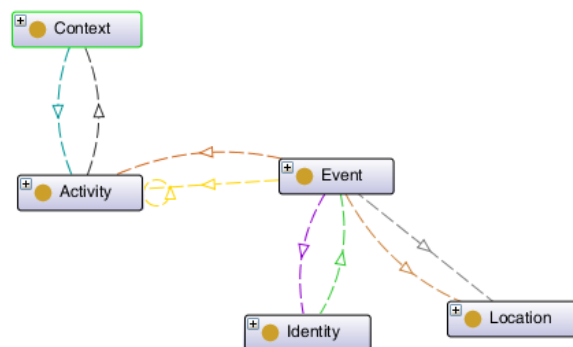


Figure 18 - Context Model Graph with a relation to the Context Concept.

After explaining the context entities involved, we want to see where the data comes from that create new instances of these concepts.

We gather the *identity* from the widget shown in Figure 19. A desktop client like SECS could extract the user’s first and family name, which is part of the identity, from a login name as demonstrated in the

prototype. But it also can use the user account name to store an enterprise unique and fully qualified name (FQN). This FQN can be used a URI of the very personal mental model. Best of all – it works without being bound to an enterprise user repository. There is another identity encapsulated within the activity. More to this down a few paragraphs.

The same widget providing user identity also offers to fetch and store the user’s *location*. The SECS prototype implements a complex html-parser accessing an internet site. It’s free, but less comfortable than implementing geographical web service with liability to pay. See the Appendix for html-parser details. The internet-fetched data comes with certainty. This certainty is currently not considered in any semantic operation. But we can think of trivial limitation policies, rejecting datasets if they arrive with a low certainty level. The place to apply them in a semantic system is in the rules engine [131] respectively in our planned Rule Manager. If data retrieval fails or data is inaccurate, the user could manually enter the information.

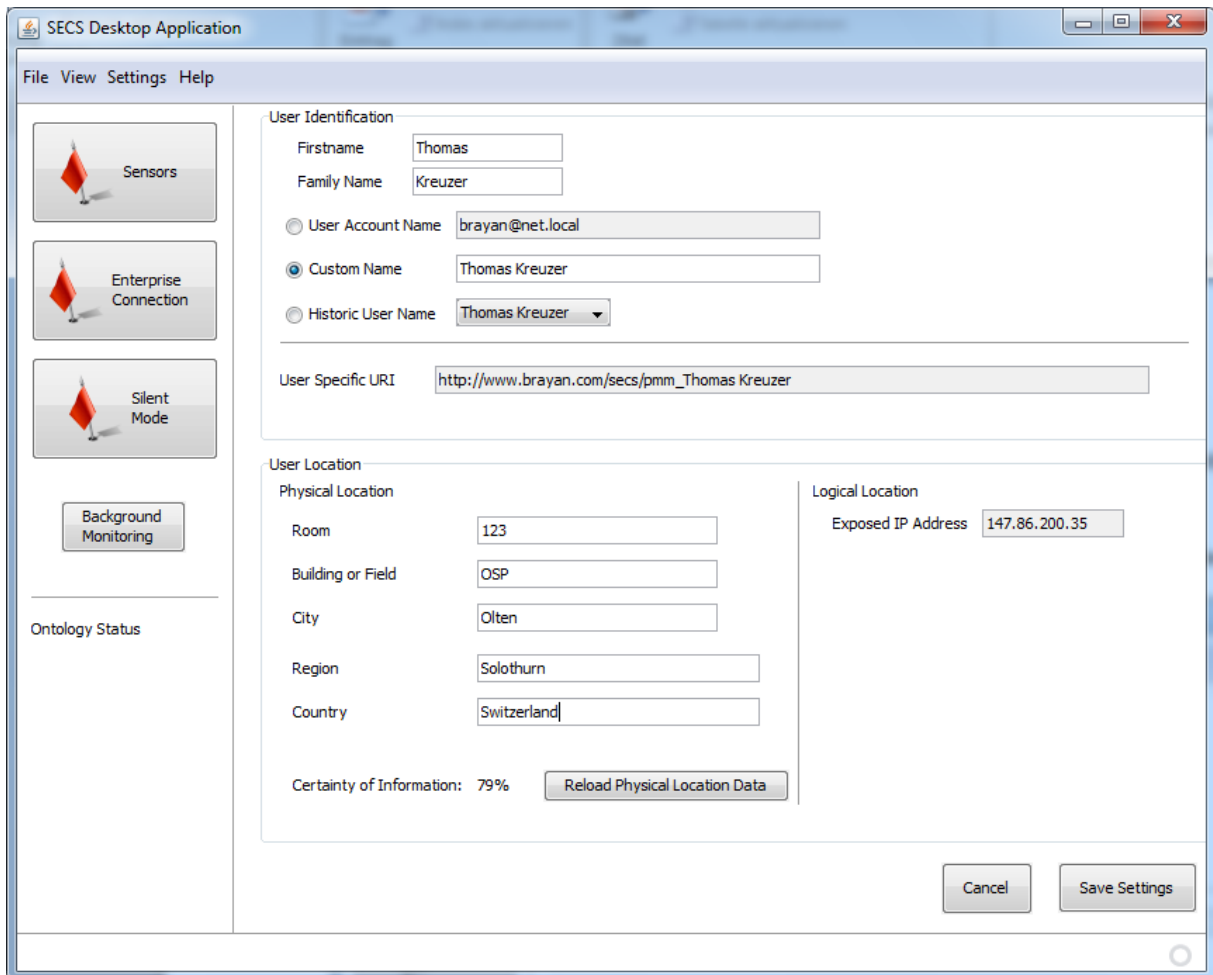


Figure 19 - Helping to Populate Spatial and Identity in a Semi-Automatic Approach

In our ontology model the concept *activity* is regarded as an unordered “doing”, which is not embedded in an array of equivalents. Furthermore, it does not hold any specific objective – at least not for system in a raw state. On the other hand it optionally uses the predicates *hasInputResource* and *hasInteractionType* - to name only three - to ease the process of context elicitation. An input resource is expected to be of the type *Person*, *Information* including all subclasses like

regulations and policies, and Organization. Output resources are limited to the type Information and its subclasses. Different from these two predicates the interaction type of an activity is compulsory and quantified by

$$\forall x(\text{Activity}(x) \wedge \exists y(\text{InteractionType}(y) \rightarrow \text{hasInteractionType}(x,y))) \quad (4)$$

It is the Context Manager's and the rule set's duty to consider these constraints. But why do we depend on them and how do we apply it? Because the more information we have about an activity, the higher is the probability for a successfully establishing a link between an activity and a context. More about how we apply and assemble information in the next chapter.

So far we have described the four context elements, which we can employ to find the appropriate context a user operates in. It is the theoretical answer to this part of the problem statement. On the other hand we are required to store *demands*. In the SECS prototype the demand is represented by the array of events. A critical mind immediately questions this approach, because events recognized by the prototype simply represent historical information and do not automatically equal the user demand. Right. On the other hand one could argue, that events fired by the user are of interest to the him/her, if input and outputs resources of these events require new action. Thus, those new requirements represent implicit demand often not know while the event is provoked. In the context of our example, think of a word document that is categorized as a contract with some keywords. If the company wants contract drafts to be verified by a internal lawyer, than a new implicit demand raises for the user from this policy.

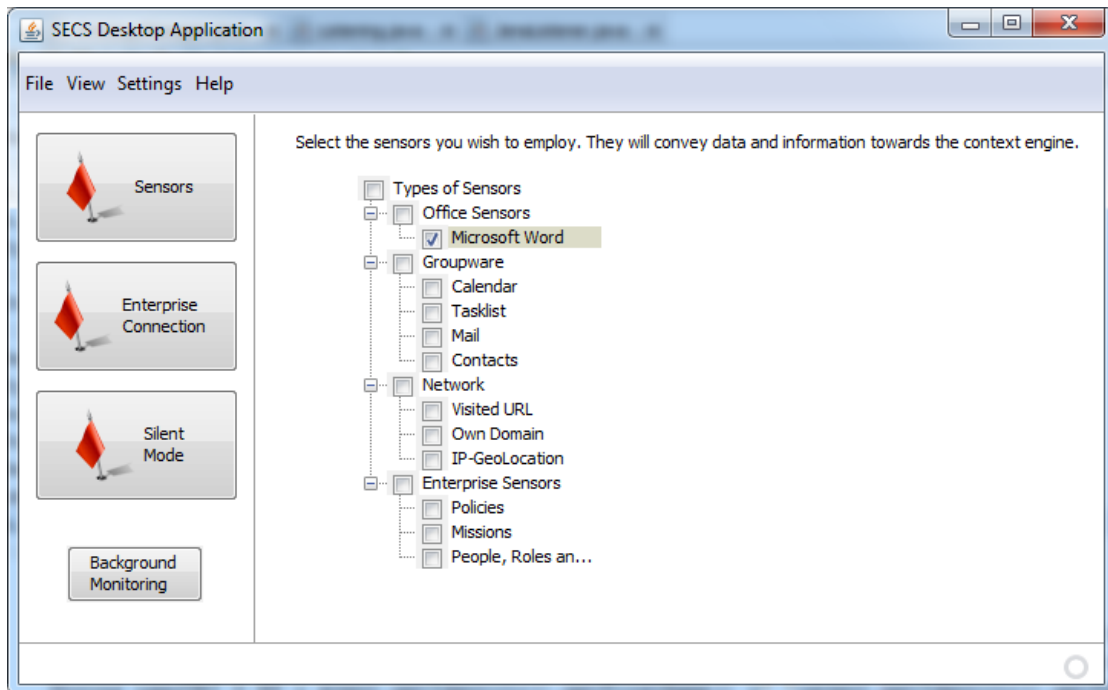


Figure 20 - Sensors Employed by the Sensors Manager

Manually the user can steer his own information support for implicit demands by subscribing to specific types of sensors. The sensors will listen to environmental events that are mostly of logical nature. In other words, issued by the applications or system services of the very own personal computer. Others come from the network e.g. if a subnet mask and IP address is changed, another location is assumed and other

implicit demands could arise. Enterprise Sensors can detect inconsistencies within locally stored policies, missions and organizational states. These inconsistencies can also be reported as events with the interaction type value change, create or – if no longer needed - delete. But the issuer is not declared as a user, rather its a computer, network or the enterprise instance. The predicate is termed `isIssuerType`, has domain Identity and range of individuals of the concepts `IssuerType`. Figure 20 gives you an idea of the different types of sensors one could employ to harvest new information.

Sensors implementation is costly in time. The only sensors that we implemented to demonstrate the success of our model is the *Microsoft Word sensors* as indicated in Figure 20. The sensor observes a customized file system detector which immediately creates an instance of the class `SensorEvent`, which is a subclass to `Event`, upon user's save command. This class is a copy of our ontological concept of `Event`. It also can store in it's body date, identity, location and activity. The activity stores document resources and type of issuer, software type, event interaction type. Since the `MsOfficeFileSensor` subscribed to updates in the `MsOfficeFileDetector`, the sensor grabs the passed object of type `SensorEvent` and starts to read meta information from the file and appends it to document information. We do this at this stage, because the prototype will never store original source like binary file or openxml documents like Microsoft Word 2007+. Experts of informations retrieval already now assume what we are going to use these meta data in a later semantic process step. Apart from that, a programmer has also the option to compose a meaningful activity description. When accomplished, the sensor event is forwarded to the sensor manager.

The *Sensor Manager* picks up the delivered sensor event object and populates missing information. This is often the location and identity of a fired event. Remote sensors like an enterprise or P2P sensor would certainly have to deliver it's identity. We remember, a computers full qualified name can also represent a identity. The activity must be inserted at this stage, otherwise the event is not forwarded to the Context Manager. Our example with the contract document would pass this test and therefore be dispatched.

The *Context Manager's* class structure and methods are designed to implement heuristics and meta-heuristics to increase the quality accuracy of the event information. To give an example, the manager could search historic event's from the ontological data repository and try to disclose information patterns. E.g. If an word document is always created before the end of a month, stored at the same place, by the same other, the probability is high that it's a report and the document meta data stored within the event object could be classified as report. It's a business decision weather a user should be prompted to accept these vague implications. After all, the event and all its sub objects, namely date, identity, location and activity, are stored in the ontology using the JENA Framework. The ontology can now be queried for that relation-rich information.

On the basis of this example we illustrated how we implemented context and how demand arised in a implicit way. Now, we want to jump to the next level and see how we assemble information to create new knowledge.

3. Knowledge Assembling and Awareness Support

We learned that the Context Manager inserts events into the ontology. One could expect that adding new information to an ontology store, could lead to new information. This is according to what we highlighted in chapter II.A.1.c) and other researchers praised [83, 99, 137, 146, 190-192] about ontology models. The inability to do so would impede our strain to up-shift the cognitive layer described by Carpenter [20], and thus disqualify our prototype. In the search for a proper technical approach on how we treat this new information we implemented a change listener, namely the Ontology Listener. It would fire on all additions to the listener, to be precise, if we add a statement to the model. So, the previous describe event insertion would trigger a method that would allow us to react.

Another way to react on model changes are rules in conjunction with a reasoner. If we previously deposit rules, register them with a reasoner and attach the reasoner to the model, then we qualify to create new instances. The reasoner will create the new resources right away, without a programmer to write a single line of Java code. To this advantage we obey, and thus, implement those rules directly in our Ontology Manager. In a later prototype the reasoner could be fed by the Rule Manager.

The first objective we have is to link an activity with a context. We can avail ourselves of all predicates and resources to do so. In our example we keep the rule rather simple and limit it to eight-element condition representing the antecedent and a single consequent. The multi-conditioned antecedent employs a SWRL built-in functionality: regular expression. It's a powerful function to match literals as we can find in the meta data of our resource document, which is attached to the activity.

The rule conditions the antecedent: matching Events which have an Activity, are of the InteractionType Update, have an InputResource with a Category and match at least one keyword of a Context with of the InteractionType Update.

```
rules = ""+
"[mapEventToContext: " +
" (?evt rdf:type pmm:Event) " +
" (?evt pmm:hasActivity ?act) " +
" (?act pmm:hasInteractionType pmm:Update) " +
" (?act pmm:hasInputResource ?res) " +
" (?res pmm:hasCategory ?cat) " +
" (?con rdf:type pmm:Context) " +
" (?con pmm:hasInteractionType pmm:Update) " +
" (?con pmm:hasKeyword ?conKeyword) " +
" regex(?conKeyword, ?cat) " +
" -> " +
" (?act pmm:isContextuallyPartOf ?con)" +
"] ";
```

The consequent of a match is the creation of the predicate `isContextuallyPartOf` in the direction from the Activity towards the Context. According to our model the inverse predicate from the Context to the Activity is `isContextuallyComprisedOf` and applied by our reasoner when needed. Meaning, from now on, we could also query a context and all its linked activities, even though we never explicitly defined it. Most the predicates in our model use the inverse function. Furthermore, one

might understand the power of the `hasInteractionType` predicate. Sometimes a context is closely associated with a type of interaction. In our case, we don't want to link a change in a contract document in a context named "Contract Cancellation" but rather in "Contract Creation", especially when the document contains the keyword "draft", implicating that a new contract is created that in the temporary state.

As soon as the link between an activity and a context is established, we demand another rule. A rule, that creates a notification for the user. Why? It is because at Alpiq AG wants internal lawyers to rubber-stamp new contract drafts. According to this business requirement derived from UC1 we setup a policy concept that allows the prototype – with some complex deduction – to identify this regulation and bring it to the user's attention. This is our awareness support. The awareness support must be enriched with collaborative information. If not, our tentative sub-hypothesis does not hold and it is barely possible to prove our main tentative hypothesis. However, we are not there yet; so, back to our ontology model and the rules.

The complex relations to stimulate the awareness support are crucial and are not as trivial as the first rule. Figure 21 illustrates the consistent link of relationships from the concept `Event` over the `Policy` down to the `Employee`. Sometime that path does not directly lead from a concept we use to concept we seek. An imperative task for example, represents a dictated task by a policy. It is required to be executed to comply with a `Policy`. Since the superior concept `Task` has a defined predicate `isExecutedBy`, the `ImperativeTask` can profit from this defined generalization and can also identify the assigned `Authority`, `Role` and finally the very `Employee` performing the role.

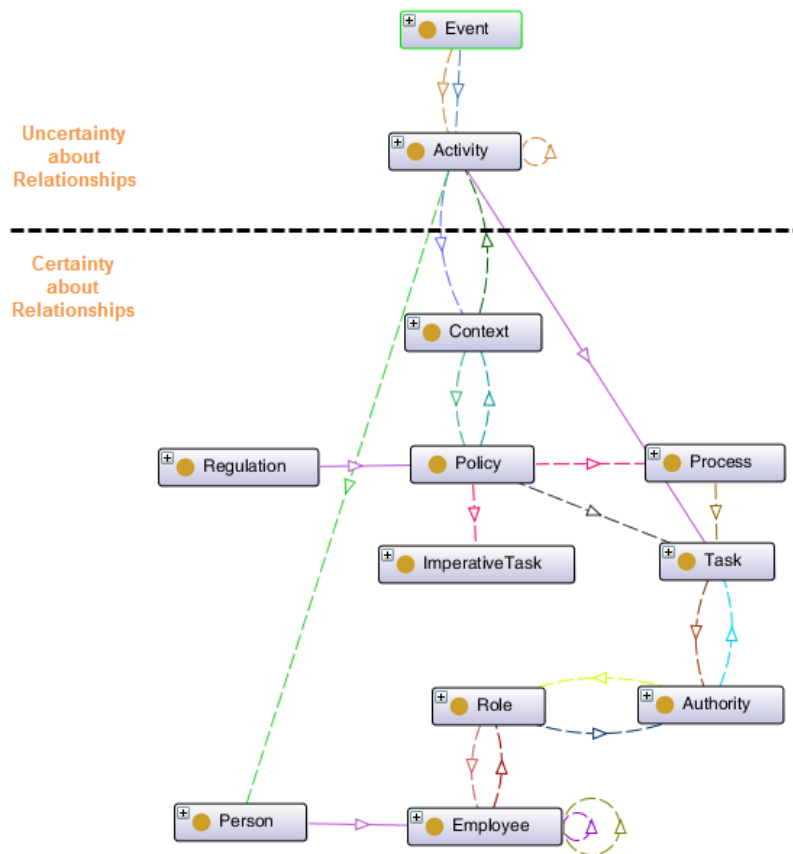


Figure 21 - Relations Used to Apply a Rule that Triggers Awareness Support

Along this verbose relational chain we can extract valuable information that might improve the awareness and collaboration support. From the policy that is bound to the context one could gather the enactment date represented by the predicate `enactedSince` and date value. Since we know the task to be executed, we may also use our model to find the process the task belongs to and therefore know about allegedly preceding tasks, using the predicates `isPartOfProcess` and `hasPrecedingTask`. While we retrieve the role through the nodes authority, we may also retrieve the geographical area that a role is responsible for, bearing in mind that Alpiq AG is a supranational enterprise. From the employee that performs the role we can imagine heap of information available: first name, family name, phone, email, location, supervisor, organizational unit, etc.

All these given and derived facts can be extracted and packed into a notification object. An example of this and how it's implemented is shown in the antecedent of the rule below.

```
rules += "+
  "[createNotificationUponActivityContextMapping: " +
  "(?act pmm:isContextuallyPartOf ?cont) " +
  "(?cont pmm:compliesWith ?pol) " +
  "(?pol pmm:dictates ?task) " +
  "(?task pmm:hasUserFriendlyDescription ?descrT1) " +
  "(?task pmm:hasNextTask ?nTask) " +
  "(?nTask pmm:hasUserFriendlyDescription ?descrT2) " +
  "(?nTask pmm:allowedToExecuteBy ?auth) " +
  "(?auth pmm:isCharteredBy ?role) " +
  "(?role pmm:isPerformedBy ?emp) " +
  " -> " +
  "makeInstance(?act, pmm:triggersNotification, pmm:Directive, ?dire), " +
  "(?dire rdf:type pmm:Directive) " +
  "(?dire pmm:hasMessage ?descrT1) " +
  "(?dire pmm:isHot \"true\"^^xsd:boolean) " +
  "makeInstance(?act, pmm:triggersNotification, pmm:Hint, ?hint), " +
  "(?hint rdf:type pmm:Hint) " +
  "(?hint pmm:hasMessage ?descrT2) " +
  "(?hint pmm:recommendsContact ?emp) " +
  "(?hint pmm:isHot \"true\"^^xsd:boolean) " +
  "(?dire pmm:precedesNotification ?hint " +
  "]" ";

//Attention: some of the consequent statements were removed to reduce the complexity. The idea
remains the same. See the OntologyManager.java file for the full implementation.
```

In our example, the consequent consists of several implications. From a technical perspective we look at them as insertion commands running against the knowledge base, our ontology model. In sequence that is as follows. We first create a `Directive`, a sub-class of `Notification`, which has `Message` that is populated with the user-friendly `Task` description. Remember, the `Task` is dictated by the `Policy`. We create another `Notification`, this time a `Hint`, which is populated with the user-friendly task description of the next `Task` found in the antecedent. At the end we mark both notifications as “hot” and link them. This will help the programmer to mark a notification as “read” and to interlink the notifications when accessing them later and out of this very context.

At the end, the collaboration is supported by one or multiple instantly fired notifications. That is the implication, respectively the creation of the consequent. With strong confidence we assert that no one would dare to challenge this semantic deduction with a conventional database manipulation language.

A major risk of this described inference path is the correct linking between the upper area and the lower area as denoted in Figure 21. While the information deduced from the latter can be guaranteed as certain and applicable, the information from the former is always vague. Information could be wrongly collected, misinterpreted or simply provide by a provenance with unreliable data. The pursuit of optimization in the upper area is always favourable, but especially in the early hours of an application. We recommend applying the Pareto-Principal to handle this ongoing challenge.

At the very end of this interaction with the prototype we need to convey the notification to the user. Most semantic desktops prototypes like NEPOMUK [159] and Haystack [161] are dominant and expect the user to watch it for semantic changes in the knowledge base. The SECS Desktop Client is different and can be run in the background and in silent mode. The latter is used when in a meeting or when performing a presentation while others watching. In this mode the notifications are suppressed – but not lost. They can be reviewed in the notification centre of the client at any time as shown in Figure 22. The Notification Manager class holds the business logic that extracts the notifications from the knowledge base. Any GUI or other classes can subscribe to the Notification Manager and retrieve new notifications in order to stay up to date. This is because it is implemented as an observable class.

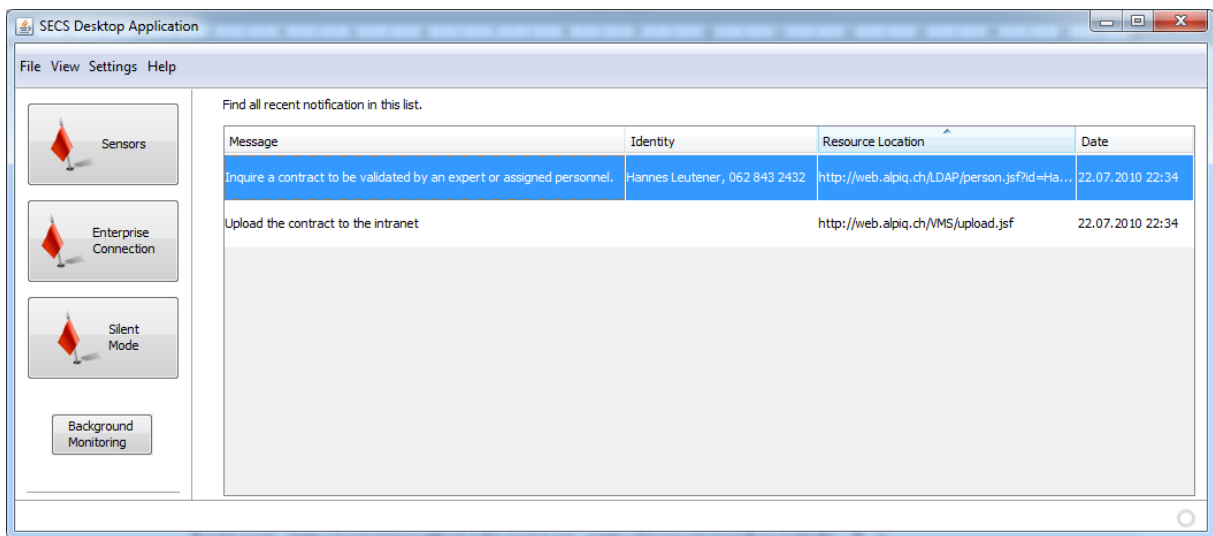


Figure 22 - The Notification Centre Stores all Notifications

When silent mode is not activated, notifications will appear above the system tray icon of the semantic desktop as a short message. It leaves the decision upon the user whether the content of the message is important to his/her current activity. It remains unobtrusive, avoids dominating the screen, but still offers information that could increase collaborative interaction within an enterprise. The notification consists of a message, either a directive, hint or recommendation; an identity and contact details; and resource location. `hasResourceLocation` is a predicate with two sub-predicates. `hasFilePath` stores local or network paths to files and can only be applied to individuals of the concept `Artefact`. The generalized concept `Information` can use the second sub-predicate `hasURL`.

Semantic Information Awareness Stimulation



The SECS desktop client is non-dominant and non-obtrusive towards the user and remains in the system tray until prompted.

The SECS desktop client presents a directive that must be followed by the user. User contacts are included. A click on the message opens the Notification Centre with additional information.

Figure 23 - Tray Icon Messages Issued by the SECS Desktop Client.

After all, these knowledge assembling and awareness supporting functionalities, which were prescribed by the BSR and TSR, represent the SECS prototype implementation.

4. Trust & Accuracy

A fact is that our world is surrounded by uncertainty. Only a few items are black or white but rather maintain a degree of greyness. Probabilities quantify greyness. But during the literature review and the design phase we have learned that uncertainty of trust and accuracy is a focus on its own. It requires massive architectural effort and cannot be implemented by catching a glimpse of theory. However, there is a simple way to implement a low-cost solution, although it cannot meet the excellence of other recommendations [94, 129, 131, 136, 138, 182, 193-195].

The concept `QualityThreshold` represents a limit of quality an individual must fulfil. Otherwise the individual is not taken into consideration by the caller. A caller could be a rule or a query. Thus, every caller can set its own threshold and the qualifying individuals will be part of its very own computational endeavour. `Quality` is a resource that can be connected to any arbitrary object, in OWL that is named `Thing`. The `Quality` of an individual is comprised of the predicates `hasTrust` and `hasAccuracy`. The range is floating numeric values. A caller has similar predicates: `hasTrustThreshold`, `hasAccuracyThreshold`. It is easy to imagine how a rule or a query could be extended: two quality conditions are programmatically appended to the regular condition in the antecedent, respectively in the where clause of a query. Queries could even leverage the argument “optional” of the SPARQL standard. That is to say, a query with the keyword `optional` returns individuals with and without quality values. Analogous to the nature of our intentions, an application logic could then favour the individuals with quality values and consider the rest of individuals in a subordinate position. Our planned Rule Manager is the optimal place for this implementation because it will maintain all rules applied on the ontology model.

The drawback of this solution is that values of individuals are hard to get and need to be assigned manually. It requires another predicate, `hasCertainty` with `hasCertaintyUpperBound` and `hasCertaintyLowerBound`, to provide a confidence interval of the certainty of the assumed quality values. Of course, this could be automated and computed with sophisticated algorithms, e.g. the use of probabilities and other techniques described in chapter II.C.

The advantages of the our proposal is that it fosters the genericity, flexibility and scalability of information modelling and assembling, saying they assure ontology criteria [83]. Genericity refers to the fact that every `Thing` can apply the idea of information quality. Scalability is represented by the free degree of intensity in the appliance. If millions of triples with the same predicate exist, one could choose to abandon

the quality computation by the callers for the most unnecessary individuals. Apart from these advantages, our approach is inexpensive in computing time, because values are assigned manually e.g. by sensors, and only where needed. This is our limited contribution to process-integrated deficiency management as we conceptually illustrated in Figure 13.

C. Assessment

Our stated tentative sub-hypothesis requires us to verify the existence of awareness support featured by collaborative knowledge. Since Alpiq AG selected parts of Use Case 1 to test against, highlighted in Table 6, we implemented those features in the prototype as explained in this section.

Maturity Level	UC1 - Sharing			
	Maturity Support Criterion	Instant Awareness Support	Collaboration Support Criterion	Explanation
ML1, initial	<input type="checkbox"/> OutPre <input type="checkbox"/> AdSup	<input type="checkbox"/> Socially-aware <input type="checkbox"/> Task-aware <input type="checkbox"/> Concept-aware <input type="checkbox"/> Workspace-aware	<input type="checkbox"/> CoEx <input type="checkbox"/> Comm <input type="checkbox"/> Coord <input type="checkbox"/> Coop	CoEx is supported , since the user is provided with the existence of the responsible expert his/her availability information. Comm is supported , since the user is provided with phone number and email of the responsible expert. Provides clickable mailto:: link in Notification Centre. Coord is supported , since user is briefed on what to do next with the contract document. Coop is partially supported . The user is told to cooperate with the responsible expert, but the system cannot get the cooperation support started nor can it.
ML2, managed	<input type="checkbox"/> ProdReq <input type="checkbox"/> ResAll			
ML3, defined	<input type="checkbox"/> PolEnf <input type="checkbox"/> ProdVer			

Table 7 - Assessing Collaboration Features for Parts of UC1

The assessment of the collaboration support criterion in Table 7 shows that Co-Existence, Communication and Coordination can be provided with the prototype. Co-Existence suggests a qualified person. Communication can profit from existing applications like a mail program, that can be invoked through a mailto:: link. Cooperation – and also Coordination – is stimulated, but both could profit from integration in the To-Do list and calendar of their groupware system. All together, we summarize that collaboration support is reasonably established. The tentative sub-hypothesis holds true.

Maturity Level	UC1 - Sharing			
	Maturity Support Criterion	Instant Awareness Support	Collaboration Support Criterion	Explanation
ML1, initial	<input type="checkbox"/> OutPre <input type="checkbox"/> AdSup	<input type="checkbox"/> Socially-aware <input type="checkbox"/> Task-aware <input type="checkbox"/> Concept-aware <input type="checkbox"/> Workspace-aware	<input type="checkbox"/> CoEx <input type="checkbox"/> Comm <input type="checkbox"/> Coord <input type="checkbox"/> Coop	Social-awareness is supported , since the prototype tells the user that the contract approver (expert) is important to the mission of creating a new contract. Task-awareness is supported , since the prototype explains why a task must be executed. Concept-awareness is supported , since the prototype explains the interdependencies that the concept of a contract bears. Workspace-awareness is partially supported . The tray bar icon informs instantly. On the other hand the prototype cannot inform other users or servers.
ML2, managed	<input type="checkbox"/> ProdReq <input type="checkbox"/> ResAll			
ML3, defined	<input type="checkbox"/> PolEnf <input type="checkbox"/> ProdVer			

Table 8 - Assessing Awareness Support for Parts of UC1

We take a different but similar perspective from Gutwin et al. [107] and verify awareness support in Table 8. Social-awareness is supported by CoEx, Coop and Comm. In this use case it clearly brings forward the

Semantic Information Awareness Stimulation

people involved in the mission of the contract creation activity. A user is also task-aware respectively sensitive, because the prototype explains why the task must be executed. Apart from that, the user would also understand that this document is not simply a document but is of the concept “contract” that must be specially treated. The integration with groupware is – as previously mentioned – not existing. On the Workspace/Desktop the user is non-obtrusively informed of new knowledge but has no capabilities to provoke interactions with other users. For example, if the draft contract is created an expert might want to be alerted. Since there is no Enterprise instance in place or a P2P-architecture established with the SECS Desktop Clients, it is not possible for the prototype to multicast messages to interested employees, and thus stimulate workspace-awareness.

From the two observations in Table 7 and Table 8 two strong deficiencies are illuminated. First, the limited Coop support hampers the workspace-awareness. It is not possible to monitor cooperation and progress of the single task. In other words, the clients are isolated as long as no enterprise instance picks on this limitation. E.g. a contract could already be approved by an expert, but the SECS client doesn’t recognize it and prompts the user again and again with the Directive shown in Figure 23. The system does it as long as the contract is categorized as “contract” and the status is not set to “final”. A cooperation supportive system would recognize the expert’s validation activity and prompt the user to set the document to “final” or whatever status applies.

At last, Alpiq AG verifies criteria of the last perspective, the CMMI maturity levels. Note, in the CMMI documentation, they are called capability levels, when only used in specific processes and not institutionalized as requirements over the entire organization.

Maturity Level	UC1 - Sharing			Explanation
	Maturity Support Criterion	Instant Awareness Support	Collaboration Support Criterion	
ML1, initial	<input type="checkbox"/> OutPre <input type="checkbox"/> AdSup	<input type="checkbox"/> Socially-aware <input type="checkbox"/> Task-aware <input type="checkbox"/> Concept-aware <input type="checkbox"/> Workspace-aware	<input type="checkbox"/> CoEx <input type="checkbox"/> Comm <input type="checkbox"/> Coord <input type="checkbox"/> Coop	<p>OutPre is supported, because the prototype’s ontology model knows the concept of a process and can therefore define that task but also the output resource.</p> <p>AdSup is supported, because the prototype is able to react with sensors immediately to ad-hoc actions without prompting the user for his/her process step.</p> <p>ProdRequ is supported, because the prototype displays the next task of an identified process. If a regulation (policy or recommendation) exists, the required input and output resources are displayed.</p> <p>ResAll is supported. It knows and recommends resources by name.</p> <p>PolEnf is partially supported, because the prototype can link activity to context and context are governed by policies. However, it cannot literally enforce the user to comply with regulations.</p> <p>ProdVer is supported, because the prototype prompts the user to have the contract document verified.</p>
ML2, managed	<input type="checkbox"/> ProdReq <input type="checkbox"/> ResAll			
ML3, defined	<input type="checkbox"/> PolEnf <input type="checkbox"/> ProdVer			

Table 9 - CMMI Maturity Level Test Conducted by the Focus Group

The comments in the explanation column are accepted by Alpiq AG. They attest the model’s capability and the feature supported by the instant awareness support offered by the SECS Desktop Client. A weekly supported capability is the Resource Allocation. This weakness results from the previous identified

deficiency: now enterprise resource sharing. How could a company's resource - that is a Person, Organization or Information - be allocated during a task execution if no enterprise repository is available? Another great prove of advantage of this prototype reveals in the test of PolEnf. Policy enforcement is required by corporate governance; a hot topic in any industries these days and a field where semantic collaboration could assist to improve effectiveness and efficiency.

In a final assessment statement we can testify that the tentative hypothesis holds true. Instant awareness fosters - at least partially - the selected maturity levels. It basically builds on knowledge sharing as exemplary shown with the policy that provided the directive for the user. The confirmation of the CMMI-assessment is attached to this thesis. The product deserves the predicate "fit for purpose".

D. Summary

The cooperative CMMI-assessment with the focus group of Alpiq AG proved the reasonable implementation of our prototype, referenced in Table 9. In other words, for the limited use case 1 provided by Alpiq AG, one could expect non-obtrusive but concise awareness support and attests their contribution to the maturity-increase of the process "Contract Creation".

The provided use cases allowed choosing activities from a real-life environment; with true business criterions that allowed us to empirically observe real business operations, but also offered sufficient abstraction to make our solution generic allowing it to be universally applicable to any environment. From this we developed our tentative hypothesis, expecting instant awareness support to significantly foster business maturity, as we have tested against successfully. It is an entailment of the sub-hypothesis, which claims that the prototype employs sufficient collaboration features to stimulate the awareness support.

This findings result from a solid architecture that from now on can be arbitrarily extended on the graphical, business, data and layer. We applied SWRL rules on our ontology to overcome the challenges of context elicitation. Furthermore, we use them to react on new event invoke by an issuer. If the event can be linked to a context, notifications are fired that non-obtrusively convey awareness towards the user. Talking by example, we help a user to comply with a policy in the context of a contract draft creation. All these features contribute to collaboration and awareness support criterions, which turned out valuable and therefore significant. They meet our tentative sub-hypothesis, which was a condition on the way to the earlier mentioned verification of the main hypothesis.

V. Findings, Conclusion & Outlook

With this conclusion our inductive research process - explained in Figure 14 - comes to an end.

The motivation for this project started by trying to understand which effects hamper the efficiency of an organization and its individuals. A verbose introduction disclosed the bulk of information and resources as main factor. To this adds weak effectiveness in information processing, mainly due the quality of information. These obstacles impede the cognitive energy up-shift (Figure 4) in the knowledge pyramid.

One of the requirements we identify to support this up-shift is collaboration. We quickly understood that such appropriate solutions can be categorized in two dimensions: tacit- and explicit-oriented solutions (Figure 7). If we could stimulate the awareness of these two types of knowledge solutions, a system could tackle the problems of inefficiency. This provided the direction to look for a solution.

From Figure 9 of our problem statement we identified the areas we would need to focus at an early stage. Building on a substantial literature review, we were able to limit our research intention to elements 1 and 4 of our Compact Chain of Knowledge Management (CCKM) which comprised some ambitious research gaps we identified in this section. Context elicitation and awareness stimulation were identified as our focus. Besides of this identification another massive group: the inaccuracy, inconsistency, vastness and trust and uncertainty in during the knowledge assembling activities (CCKM-2). Later in the literature review we would name it the deficiencies of knowledge assembling (Figure 13) which we don't tackle extensively in this thesis.

The literature review also revealed the components we need to be able to create a solution that would stimulate awareness of knowledge: ontologies, reasoning and sensors are the main components.

In our research design (Figure 14) section we illustrate our research approach to stringently explain the identification of requirements, design, implementation and the assessment of the final solution. The solutions had to be accepted by our customer, Alpiq AG, which provided a focus group. That group provided the project with use cases and business requirements. From these requirements we inferred application and technical service requirements that enormously eased the design process along clear and transparent thread. Those business requirements delivered where tested in a user acceptance testing and against standardized maturity terms from the CMMI recommendations.

After all and from a business perspective, we can claim that our tentative hypothesis holds true; instant awareness support can significantly foster – not enforce – business maturity. We also learned that semantics is a solid approach in pushing governance endeavour forward, especially when policies come into play.

In the near future we certainly see additional effort in the deficiency management (Figure 13) of knowledge assembling activities. In regards to our SECS Desktop Client we recommend the creation of either a P2P socket or an enterprise SECS server that could offer leverage collective intelligence. Apart from that one might develop additional sensors to multiply the usability of the SECS Desktop Client.

Findings, Conclusion & Outlook

Appendix A – Examples of Theory and Implementation Documentation

A. Examples of Context Description

1. UAProf Example

A User Agent Profile describing the identity and capability of a device, if employed, allowing various inference statements. E.g. decide what person to contact when knowing exact GPS-Position.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschem-20021212#"
xmlns:mms="http://www.openmobilealliance.org/tech/profiles/MMS/ccppschem-20050301-MMS1.2#">
  <rdf:Description rdf:ID="Profile">
    <prf:component>
      <rdf:Description rdf:ID="HardwarePlatform">
        <rdf:type
rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschem-
20021212#HardwarePlatform"/>
        <prf:BluetoothProfile>
          <rdf:Bag>
            <rdf:li>Headset Profile</rdf:li>
            <rdf:li>Handsfree Profile</rdf:li>
            <rdf:li>Object Push Profile</rdf:li>
          </rdf:Bag>
        </prf:BluetoothProfile>
        <prf:BitsPerPixel>16</prf:BitsPerPixel>
        <prf:ColorCapable>Yes</prf:ColorCapable>
        <prf:CPU>Wacek</prf:CPU>
        <prf:ImageCapable>Yes</prf:ImageCapable>
        <prf:InputCharSet>
          <rdf:Bag>
            <rdf:li>ISO-8859-1</rdf:li>
            <rdf:li>UTF-8</rdf:li>
          </rdf:Bag>
        </prf:InputCharSet>
        <prf:Keyboard>PhoneKeyPad</prf:Keyboard>
        <prf:Model>WikiPhone-1</prf:Model>
        <prf:NumberOfSoftKeys>2</prf:NumberOfSoftKeys>
        <prf:OutputCharSet>
          <rdf:Bag>
            <rdf:li>ISO-8859-1</rdf:li>
            <rdf:li>UTF-8</rdf:li>
          </rdf:Bag>
        </prf:OutputCharSet>
        <prf:PixelAspectRatio>1x1</prf:PixelAspectRatio>
        <prf:PointingResolution>Pixel</prf:PointingResolution>
        <prf:ScreenSize>640x480</prf:ScreenSize>
        <prf:ScreenSizeChar>80x24</prf:ScreenSizeChar>
        <prf:StandardFontProportional>Yes</prf:StandardFontProportional>
        <prf:SoundOutputCapable>Yes</prf:SoundOutputCapable>
```

Appendix A – Examples of Theory and Implementation Documentation

```
<prf:TextInputCapable>Yes</prf:TextInputCapable>
<prf:Vendor>Fantasy</prf:Vendor>
<prf:VoiceInputCapable>Yes</prf:VoiceInputCapable>
</rdf:Description>
</prf:component>
<prf:component>
  <rdf:Description rdf:ID="SoftwarePlatform">
    <rdf:type
rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschem-
20021212#SoftwarePlatform"/>
    <prf:AcceptDownloadableSoftware>Yes</prf:AcceptDownloadableSoftware>
    <prf:CcppAccept>
      <rdf:Bag>
        <rdf:li>application/java</rdf:li>
        <rdf:li>application/vnd.wap.mms-message</rdf:li>
        <rdf:li>application/vnd.wap.wbxml</rdf:li>
        <rdf:li>application/vnd.wap.wmlc</rdf:li>
        <rdf:li>application/vnd.wap.wmlscriptc</rdf:li>
        <rdf:li>application/vnd.wap.xhtml+xml</rdf:li>
        <rdf:li>application/xhtml+xml</rdf:li>
        <rdf:li>audio/midi</rdf:li>
        <rdf:li>image/gif</rdf:li>
        <rdf:li>image/jpeg</rdf:li>
        <rdf:li>image/jpg</rdf:li>
        <rdf:li>image/vnd.wap.wbmp</rdf:li>
        <rdf:li>multipart/mixed</rdf:li>
      </rdf:Bag>
    </prf:CcppAccept>
  </rdf:Description>
</prf:component>
<prf:component>
  <rdf:Description rdf:ID="NetworkCharacteristics">
    <rdf:type
rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschem-
20021212#NetworkCharacteristics"/>
    <prf:SupportedBluetoothVersion>1.2</prf:SupportedBluetoothVersion>
    <prf:CurrentBearerService>TwoWayPacket</prf:CurrentBearerService>
    <prf:SecuritySupport>
      <rdf:Bag>
        <rdf:li>SSL</rdf:li>
        <rdf:li>TLS</rdf:li>
      </rdf:Bag>
    </prf:SecuritySupport>
    <prf:SupportedBearers>
      <rdf:Bag>
        <rdf:li>GPRS</rdf:li>
      </rdf:Bag>
    </prf:SupportedBearers>
  </rdf:Description>
</prf:component>
<prf:component>
  <rdf:Description rdf:ID="BrowserUA">
```

```
<rdf:type
rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschem-
20021212#BrowserUA" />
  <prf:BrowserName>WikiBrowser</prf:BrowserName>
  <prf:BrowserVersion>1.0</prf:BrowserVersion>
  <prf:FramesCapable>Yes</prf:FramesCapable>
  <prf:HtmlVersion>4.1</prf:HtmlVersion>
  <prf:TablesCapable>Yes</prf:TablesCapable>
  <prf:XhtmlVersion>2.0</prf:XhtmlVersion>
  <prf:XhtmlModules>
    <rdf:Bag>
      <rdf:li>XHTML1-struct</rdf:li>
      <rdf:li>xhtml-basic10</rdf:li>
    </rdf:Bag>
  </prf:XhtmlModules>
</rdf:Description>
</prf:component>
</rdf:Description>
</rdf:RDF>
```

B. Use Cases Submitted by Alpiq AG

Alpiq AG Olten submitted three suggested use cases for the purpose of a situation and criterion analysis by May 3, 2010. Benjamin Mühleisen is the single point of contact during the situation analysis. Find below the original uses cases. This business information is confidential and can only be shared among stakeholders of this thesis.

1. Zugriffsrecht auf benötigte Dokumente ausserhalb der Business Unit.

Beispiel: Wird von Trading ein Rahmenvertrag ausgehandelt, wird dieser auch vom Backoffice benötigt.

Da diese jedoch nicht in derselben BU sind, haben sie nicht automatisch das Recht das Dokument zu sehen.

Lösung System: Durch Eintragen einzelner Personen oder auch Gruppen kann diesen das Recht explizit zugewiesen werden. Dies muss jedoch manuell passieren und muss vom "Importierer" bewusst gemacht werden.

2. Benachrichtigung beim Import von ebensolchen Verträgen.

Genau wie beim oben beschriebenen Fall müssen die Abteilungen informiert werden, wenn neue solche Rahmenverträge importiert werden. Hier gibt es keine automatische Benachrichtigung.

Lösung System: Der "Importierer" muss von sich aus die Leute informieren. Per Mail, per Telefon, oder per System-internen Postkorb.

3. Freigeben von Verträgen (nach Kompetenzregelungen).

In der Alpiq gibt es bestehende Kompetenzregelungen, welche definiert welche Verträge von welchen Instanzen abgesegnet werden müssen. Diese wird im System nicht abgebildet.

Lösung System: Der Ersteller des Vertrags muss die Regelung kennen und Sie von sich aus einhalten. Als Bestätigung für die Freigabe dient entweder ein separat abgelegtes Mail mit der Korrespondenz (kann aber natürlich manipuliert werden), oder eine an das Dokument angehängte Notiz. (Hier ist revisions sicher hinterlegt wer zu welcher Zeit die Notiz erstellt hat.)

C. Assessment Documentation

Find a signed confirmation of the “CMMI Capability Verification Against SECS Desktop Clients” document attached to this thesis.

D. SECS Desktop Documentation and Resources

1. Installation Hints

There is no JAR-executable available. Please follow these steps to load and run the project.

1. To setup the project, download NetBeans 6.8 or later. No Enterprise version required.
2. Ensure all Libraries are attached to the project as denoted in the next section.
3. Ensure you downloaded the SECS ontology from website.
4. Adapt the ontology path in the OntologyManager.java file.
5. Run the project.

If you wish to edit the ontology use Protegé 4.1.

2. Resources

	Resource	Resource Location
Java Libraries	Java Standard Edition 6	http://www.oracle.com/technetwork/java/javase/overview/index.html
	Java API	
	SECS Desktop Client 0.2.43	http://www.brayan.com/projects/MasterThesisBIS/JavaDocs/index.html
	JENA 2.6.3	http://jena.sourceforge.net/downloads.html
	ApachePOI 3.6	http://poi.apache.org/download.html
	JNA 3.2.6	https://jna.dev.java.net/
	Waffle-JNA 1.2	http://waffle.codeplex.com/releases/view/43113
HTMLParser 2.0	http://htmlparser.sourceforge.net/	
Data	SECS Personal Mental Model	http://www.brayan.com/projects/MasterThesisBIS/pmmrdfxml.owl
	Java Properties	User.properties, com.brayan.secs.resources
Modelling	Protegé 4.1 beta	http://protege.stanford.edu/download/protege/4.1/installanywhere/

Appendix B – References

A. Literature References

- [1] Horgan, D.J. *Time Pressure Prevents Good Decision Making*. CIO 2001 04-03-2010]; Available from: http://www.cio.com/article/30703/Time_Pressure_Prevents_Good_Decision_Making.
- [2] Ernest-Jones, T. and G. Lofthouse (2005) *Know how - Managing knowledge for competitive advantage*. 22.
- [3] Carbonaro, A. and R. Ferrini, *Personalized Information Retrieval in a Semantic-Based Learning Environment*, in *Collaborative and Social Information Retrieval and Access: Techniques for improved user modeling*, M. Chevalier, C. Julien, and C. Soule-Dupuy, Editors. 2009, Information Science Reference: Hershey. p. 294-312.
- [4] Göbler, W., *The alignment of knowledge strategy and business strategy : a field study of authorized deposit-taking institutions in the Australian financial services industry*, in *International Graduate School of Business*. 2009, University of South Australia Elizabeth. p. 55-58.
- [5] Holsapple, C.W. and K. Jones, *Exploring Primary Activities of the Knowledge Chain*, in *Gatton College of Business and Economics*. 2004, University of Kentucky: Lexington, USA.
- [6] Berners-Lee, T. *What the semantic web can represent*. 1998 15-03-2010]; Available from: <http://www.w3.org/DesignIssues/RDFnot.html>.
- [7] Initiative, D.C.M. *DCMI Metadata Terms*. 14-01-2008 02-02-2010]; Available from: <http://dublincore.org/documents/dcmi-terms/>.
- [8] Magdalena, et al., *Personal Knowledge Management and 'cues to knowledge'*, Magdalena, Editor. 2005, University of Heidelberg: Heidelberg.
- [9] OMG. *Semantics of Business Vocabulary and Business Rules (SBVR), Version 1*. 01-2008 04-03-2010]; Available from: <http://www.omg.org/spec/SBVR/1.0/>.
- [10] OMG. *MOF 2.0 / XMI Mapping Specification, v2.1.1*. 2007 06-03-2010]; Available from: <http://www.omg.org/technology/documents/formal/xmi.htm>.
- [11] Standardization, I.O.f., *ISO/IEC 19503:2005 XML Metadata Interchange (XMI) in Information technology*. 2005, ISO: ISO Website.
- [12] *Oxford Dictionary of English*. 2010, Oxford University Press: Oxford.
- [13] North, K. and S. Güldenber, *Wissensorientierte Unternehmensführung*. Vol. 2. 2008: Dr. Th. Gabler Verlag.
- [14] Thon, I., N. Landwehr, and L.D. Raedt, *A Simple Model for Sequences of Relational State Descriptions*. Vol. Volume 5212/2008. 2008, Heidelberg: SpringerLink.
- [15] Goasduff, L., *Creating Resiliency With the E-Workplace*. 2002, Gartner Inc.: Stamford, CT.
- [16] Petty, C. *Gartner Says IT Leaders Must Take Ownership of the Business Outcome, Not Just Concentrating on the IT Elements*. Press Release 2007 24-03-2010]; Available from: <http://www.gartner.com/it/page.jsp?id=529408>.
- [17] Gil, Y. and P. Groth, *Using Provenance in the Semantic Web*, Y. Gil and P. Groth, Editors. 2010, Journal of Web Semantics.
- [18] Standardization, I.O.f., *ISO 9000:2005 Quality Management Systems*. 1998, International Organization for Standardization.
- [19] Standardization, I.O.f., *ISO 9241-11:1998 Guidance on usability*. 1998, International Organization for Standardization.
- [20] Carpenter, S.A. *Enterprise Wisdom Management and the Flow of Understanding*. 24-03-2010]; Available from: <http://www.cognitivecybernetics.com/PrimerFoU.html>.
- [21] Carpenter, S. and J. Cannady. *Tool for Sharing and Assessing Models of Fusion-Based Space Transportation Systems*. in *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*. 2004. Fort Lauderdale, Florida.

- [22] Clark, A., *Natural-born cyborgs : minds, technologies, and the future of human intelligence*. 2003, Oxford ; New York: Oxford University Press. viii, 229 p.
- [23] Abecker, A., et al., *Context-Aware, Proactive Delivery of Task-Specific Information: The KnowMore Project*, in *Information Systems Frontiers in Knowledge Management*. 1999, Kluwer Academic Publishers: Boston. p. 253-276.
- [24] Davenport, T.H. and G. Probst, *Knowledge management case book : Siemens best practises*. 2nd ed. 2002, Erlangen and New York: Publicis Corporate Pub.; Wiley. 336 p.
- [25] Holsapple, C.W. and M. Singh, *The knowledge chain model: activities for competitiveness in School of Management, Department of Management*. 2001, University of Kentucky, Texas Christian University: Lexington and Fort Worth.
- [26] Schilit, B.N. and M.M. Theimer, *Disseminating active map information to mobile hosts*. IEEE, 1994. **Volume 8, Issue 5**: p. 22-32
- [27] Dourish, P. and V. Bellotti (1992) *Awareness and Coordination in Shared Workspaces*.
- [28] Leimstoll, U., *MSc BIS Workflow Management and SOA*, in *Institute for Information Systems*. 2009, University of Applied Science of Northwestern Switzerland: Basel.
- [29] Jarke, M., et al., *Distributed cooperation in integrated information system : proceedings of the Third International Workshop on Intelligent and Cooperative Information Systems, Schloss Dagstuhl, Saarland, April 6-8, 1992*. Aachener Informatik-Berichte,. 1992, Aachen: RWTH Aachen Fachgruppe Informatik. xvii, 131 p.
- [30] Hertweck, M., *Die koordinierungstheoretische Gestaltung und Bewertung alternativer Geschäftsprozesse unter Berücksichtigung des Einsatzes von Workflow Management und Workgroup Computing: Dargestellt am Beispiel eines Kundenanfrageprozesses*, in *Department of Informatics*. 1998, University of Freiburg im Breisgau: Freiburg im Breisgau.
- [31] Gross, T. and M. Koch, *Computer supported cooperative work*. 2007: Oldenbourg.
- [32] Kraut, R.E., et al., *Informal Communication in Organizations: Form, Function, and Technology*, in *Human Reactions to Technology: The Claremont Symposium on Applied Social Psychology*. 2002, Sage Publications: Beverly Hills, CA.
- [33] Sinclair, N. (2010) *Making sure knowledge management aligns with your business strategies*.
- [34] O'Dell, C.S., C.J. Grayson, and N. Essaides, *If only we knew what we know : the transfer of internal knowledge and best practice*. 1998, New York: Free Press. xvii, 238 p.
- [35] Patricia M. Jones, P. (2004) *Collaborative Knowledge Management, Social Networks, and Organizational Learning*.
- [36] Schmitz, C., *Self-Organized Collaborative Knowledge Management*. 2007, Kassel, Germany: kassel university press GmbH.
- [37] Howard, A. (2008) *What are the characteristics of enterprise 2.0 technologies?*
- [38] e2conf. (2009) *Enterprise 2.0: What, Why and How*.
- [39] McAfee, A.P. (2006) *Enterprise 2.0: The Dawn of Emergent Collaboration*.
- [40] Hinchcliffe, D. (2007) *The state of Enterprise 2.0*.
- [41] Wagner, C., *WIKI: A Technology for conversational Knowledge Management and Group Collaboration*. Communications of the Association for Information Systems, 2004. **Vol. 13**.
- [42] Moser, P., *Model for Multiple Representations for Ontology*, in *Scholl of Business*. 2010, University of Applied Science of Northwestern Switzerland: Olten, Switzerland. p. 76.
- [43] Wagner, C., *Breaking the Knowledge Acquisition Bottleneck Through Conversational Knowledge Management*, in *Advanced topics in information resources management*, M. Khosrowpour, Editor. 2008, Information Science Reference: Hershey, PA. p. xix, 428 p.
- [44] Moser, P. and B. Zimmerli (2009) *Developing a Semantic Enterprise Repository*.
- [45] Wieland, M., et al., *Towards Context-aware Workflows*, in *Institute of Architecture of Application Systems*. 2007, University of Stuttgart: Stuttgart.
- [46] Lei, H.D.C., *Extending the reach of business processes in Dept. of Computer Science & Electronic Engineering*. 2004, Maryland University: Baltimore, MD.
- [47] Ardissono, L., et al., *Context-Aware Workflow Management in Web Engineering*. 2007, Springer Berlin / Heidelberg: Heidelberg.

- [48] Modafferi, s., et al., *A Methodology for Designing and Managing Context-Aware Workflows* 2005, Boston: Springer Boston.
- [49] Baresi, L., et al., *Context-Aware Composition of E-services in Technologies for E-services : 4th international workshop, TES 2003, Berlin, Germany, September 7-8, 2003 : proceedings*, B. Benatallah and M.-C. Shan, Editors. 2003, Springer: Berlin ; New York. p. 28-41.
- [50] Chen, H., T. Finin, and A. Joshi, *Semantic Web in a Pervasive ContextAware Architecture*. 2004, University of Maryland.
- [51] Medeiros, A.K.A.d., et al., *An Outlook on Semantic Business Process Mining and Monitoring in On the move to meaningful internet systems 2007 : OTM 2007 workshops*. 2007, Springer: New York.
- [52] *Semantic Desktop*. 2009; Available from: http://semanticweb.org/wiki/Semantic_Desktop.
- [53] Hadjiantonis, A.M. and G. Pavlou, *Policy-Based Self-Management in Wireless Networks*, in *Chapman & Hall/CRC studies in informatics series*, W. Dargie, Editor. 2009, CRC Press: Boca Raton. p. 201-264.
- [54] Eijk, R.v., O. Coutand, and S. Holtmanns, *Sharing of Preferences and Context in Groups of Mobile Users*. 2006, Telematica Instituut: Enschede, NL.
- [55] Ma, J., J. Cao, and Y. Zhang, *A Probabilistic Semantic Approach for Discovering Web Services*, in *School of Computer Science and Mathematics*. 2007, Victoria University et al.: Victoria.
- [56] Corella, M.Á. and P. Castells, *A Heuristic Approach to Semantic Web Services Classification in Engineered Applications of Semantic Web – SWEA*. 2006, Springer Berlin / Heidelberg: Heidelberg. p. 598-605.
- [57] Weise, T., et al., *Different Approaches to Semantic Web Service Composition*, in *Distributed System Group*. 2008, University of Kassel: Kassel.
- [58] Solothurn, C.R.o.C. *Alpiq AG*. 2010; Available from: http://www.hraso.ch/cgi-bin/fnrGet.cgi?fnr=2493000044&amt=241&lang=4&hrg_opt=11000&shab=000000.
- [59] Dey, A.K. and G.D. Abowd (2000) *Towards a Better Understanding of Context and Context-Awareness*.
- [60] Ryan, N., J. Pascoe, and D. Morse, *Enhanced Reality Fieldwork: the Context Aware Archaeological Assistant*, in *Computing Laboratory*. 1997, University of Kent: Kent, UK.
- [61] Kaenampornpan, M. and E. O'Neill, *An integrated context model: Bringing activity to context*. 2004, Department of Computer Science, University of Bath: Bath, UK.
- [62] Dey, A.K., *Modeling and using context : 5th international and interdisciplinary conference, CONTEXT 2005, Paris, France, July 5-8, 2005 : proceedings*. Lecture notes in computer science,. 2005, Berlin ; New York: Springer. xiv, 572 p.
- [63] Abowd, G. and E. Mynatt, *Charting Past, Present and Future Research in Ubiquitous Computing*, in *College of Computing & GVU Center*. 2000, Georgia Institute of Technology: Atlanta.
- [64] Abowd, G.D., B. Brumitt, and S. Shafer, *UbiComp 2001 : ubiquitous computing : International Conference, Atlanta, Georgia, USA, September 30-October 2, 2001 : proceedings*. Lecture notes in computer science,. 2001, Berlin New York: Springer. xiii, 372 p.
- [65] Dargie, W., *Context-aware computing and self-managing systems*. Chapman & Hall/CRC studies in informatics series. 2009, Boca Raton: CRC Press. xxiii, 381 p.
- [66] Schmidt, A., M. Beigl, and H.-W. Gellersen, *There is more to Context than Location*, in *Telecooperation Office*. 1999, University of Karlsruhe: Karlsruhe.
- [67] Bolchini, C., et al., *A Data-Oriented Survey of Context Models*. 2007, Dip. di elettronica e Informazione, Politecnico di Milano: Milano.
- [68] Prekop, P. and M. Burnett, *Activities, context and ubiquitous computing*. Computer Communications, 2003. **26**(11): p. 1168-1176.
- [69] Koile, K., et al., *Activity zones for context-aware computing*. UbiComp 2003: Ubiquitous Computing, 2003. **2864**: p. 90-106.

- [70] Schmitt, J., et al., *Adapting the User Context in Realtime: Tailoring Online Machine Learning Algorithms to Ambient Computing*. Mobile Networks & Applications, 2008. **13**(6): p. 583-598.
- [71] Juarez, J.M., et al., *Computing context-dependent temporal diagnosis in complex domains*. Expert Systems with Applications, 2008. **35**(3): p. 991-1010.
- [72] Chen, G. and D. Kotz, *A Survey of Context-Aware Mobile Computing Research*. 2000, Department of Computer Science, Dartmouth College.
- [73] Schmidt, A., M. Beigl, and H.-W. Gellersen, *There is more to Context than Location*, in *Telecooperation Office*. 2002, University of Karlsruhe: Karlsruhe.
- [74] Coutaz, J., et al., *Context Is Key*. Communications of the ACM, 2005. **48**(3): p. 49-53.
- [75] Fahy, P. and S. Clarke, *CASS - Middleware for Mobile Context-Aware Applications*, in *Distributed Systems Group*. 2004, Trinity College Dublin: Dublin, IE.
- [76] Strang, T. and C. Linnhoff-Popien, *A Context Modelling Survey*. 2004, German Aerospace Center, Ludwig-Maximilians-Universität Munich: Wesling, Germany.
- [77] Raptis, D., N. Tselios, and N. Avouris, *Context-based design of mobile applications for museums: A Survey of existing practices*. 2005, Human-Computer Interaction Group, University of Patras: Patras, GR.
- [78] Baldauf, G.H., *Substitutive chlorination of isobutylene*. 1944. p. vii, 74 leaves.
- [79] Loyola, W., *Comparision of Approaches toward Formalizing Context: Implementation Characteristics and Capacities*. 2006, Escuela Superior Politécnica del Litoral: Santiago de Chile, CL.
- [80] Daniels, H., *Activity theory in practice : promoting learning across boundaries and agencies*. 2010, London ; New York: Routledge. xiii, 239 p.
- [81] Engeström, Y., R. Miettinen, and R.-L. Punamäki-Gitai, *Perspectives on activity theory*. Learning in doing. 1999, Cambridge ; New York: Cambridge University Press. xiii, 462 p.
- [82] Korpipää, P., et al. *Context management for end user development of context-aware applications*. in *International Conference On Mobile Data Management*. 2005. Ayia Napa, Cyprus.
- [83] Krummenacher, R., et al., *Analyzing the modeling of Context with Ontologies*, in *Digital Enterprise Research Institute*. 2006, University of Innsbruck: Innsbruck, AT.
- [84] W3C. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendations 2004 01-05-2010]; Available from: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>.
- [85] W3C. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*. W3C Recommendations 2004 01-05-2010]; Available from: <http://www.w3.org/TR/CCPP-struct-vocab/>.
- [86] Held, A., S. Buchholz, and A. Schill, *Modeling of Context Information for Pervasive Computing Applications*. 2002, Research & Technology, DaimlerChrysler AG: Ulm, DE.
- [87] Indulska, J., et al., *Experiences in Using CC/PP in Context-Aware Systems in Mobile Data Management*. 2003, Springer Berlin / Heidelberg: Heidelberg.
- [88] Chtcherbina, E. and M. Franz. *Peer-to-Peer Coordination Framework (P2PC): Enabler of Mobile Ad-Hoc Networking for Medicine, Business, and Entertainment*. in *International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies and the Internet*. 2003. L'Aquila, IT.
- [89] Ltd., W.A.P.F., *Wireless Application Protocol UAProf*. 2001.
- [90] Henricksen, K., J. Indulska, and A. Rakotonirainy, *Generating Context Management Infrastructure from High-Level Context Models*, in *4th International Conference on Mobile Data Management*. 2003.
- [91] Halpin, T. *Object Role Modeling*. 2010 05-05-2010]; Available from: <http://www.orm.net/>.
- [92] *Technology for Enabled Awareness*, E. Project, Editor. 1998.
- [93] Kim, E. and J. Choi, *An Ontology-Based Context Model in a Smart Home in Computational Science and Its Applications - ICCSA 2006*. 2006, Springer Berlin / Heidelberg: Heidelberg.
- [94] Cheung, R., et al., *A fuzzy-based service adaptation middleware for context-aware computing*. Embedded and Ubiquitous Computing, Proceedings, 2006. **4096**: p. 580-590.

- [95] Huynh, V.N., et al., *A context model for fuzzy concept analysis based upon modal logic*. Information Sciences—Informatics and Computer Science: An International Journal, 2004. **Volume 160**(Issue 1-4): p. 111-129.
- [96] Zhang, D.G., et al., *Kind of context-aware approach based on fuzzy-neural for proactive service of pervasive computing*. Embedded Software and Systems, Proceedings, 2005. **3820**: p. 554-563.
- [97] Kocaballi, A.B. and A. Kocyigit, *Granular best match algorithm for context-aware computing systems*. Journal of Systems and Software, 2007. **80**(12): p. 2015-2024.
- [98] Strang, T., C. Linnhoff-Popien, and K. Frank, *CoOL: A Context Ontology Language to Enable Contextual Interoperability in Distributed applications and interoperable systems : 4th IFIP WG6.1 International Conference, DAIS 2003, Paris France November 17-21, 2003 : proceedings*, J.-B. Stefani, I. Demeure, and D. Hagimont, Editors. 2003, Springer: Berlin ; New York. p. xiii, 309 p.
- [99] Christopoulou, E., C. Goumopoulos, and A. Kameas, *An ontology-based context management and reasoning*, in *School of Science & Technology*. 2005, Hellenic Open University: Patras, GR.
- [100] Baldauf, M., S. Dustdar, and F. Rosenberg, *A Survey on Context-Aware Systems*, in *Distributed Systems Group, Information Systems Institute*. 2007, Vienna University of Technology: Vienna.
- [101] Bolchini, C., IEEE Computer Society. Fault-Tolerant Computing Technical Committee., and IEEE Computer Society. Test Technology Technical Committee., *18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems : proceedings : 3-5 November, 2003, Boston, Massachusetts*. 2003, Los Alamitos, Calif.: IEEE Computer Society Press. xii, 607 p.
- [102] Zimmerli, B. and P. Moser, *Modelling Languages and Semantic Integration in Business and IT Worlds*, in *School of Business*. 2008, University of Applied Science of Northwestern Switzerland: Olten, Switzerland.
- [103] Horrocks, I., et al. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. 2004 05-05-2010]; Available from: <http://www.w3.org/Submission/SWRL/>.
- [104] Schilit, B., N. Adams, and R. Want, *Context-Aware Computing Applications*, in *IEEE Workshop on Mobile Computing Systems and Applications*. 1994, IEEE: Santa Cruz, CA.
- [105] Brezillon, P., *Focusing on context in human-centered computing*. Ieee Intelligent Systems, 2003. **18**(3): p. 62-66.
- [106] P., B., *Contextual Knowledge Sharing and Cooperation in Intelligent Assistant Systems*. Le Travail Humain, 1999(3): p. 263-268.
- [107] Gutwin, C., S. Greenberg, and M. Roseman (1996) *Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation*.
- [108] Bisgaard, J.J., M. Heise, and C. Steffensen, *How is Context and Context-awareness Defined and Applied? A Survey of Context-awareness*, in *Department of Computer Science*. 2004, Aalborg University: Aalborg, DK. p. 8.
- [109] Pascoe, J., N. Ryan, and D. Morse, *Issues in developing context-aware computing*. Handheld and Ubiquitous Computing, Proceedings, 1999. **1707**: p. 208-221.
- [110] Pascoe, J., *Adding Generic Contextual Capabilities to Wearable Computers*, in *Computer Laboratory*. 1998, University of Kent at Canterbury: Canterbury, UK.
- [111] Lieberman, H. and T. Selker. *Out of context: computer systems that adapt to, and learn from, context*. IBM Systems Journal; 200 02-05-2010]; Available from: [http://www.hctp.utoronto.ca/Pdf/Technology%20Course%20Readings/Out of Context.pdf](http://www.hctp.utoronto.ca/Pdf/Technology%20Course%20Readings/Out%20of%20Context.pdf).
- [112] Kim, T.-J. and M.-C. Kim, *Context Awareness using Semantic Web Technology in the Ubiquitous Learning Service*, in *Computational science and its applications : ICCSA 2008 : International Conference, Perugia, Italy, June 30 - July 3, 2008 : proceedings*, O. Gervasi, Editor. 2008, Springer: Berlin. p. 501-514.
- [113] Ranganathan, A. and R.H. Campbell, *A middleware for context-aware agents in ubiquitous computing environments*. Middleware 2003, Proceedings, 2003. **2672**: p. 143-161.

- [114] Moodley, D. and I. Simonis, *A New Architecture for the Sensor Web: The SWAP Framework*, in *School of Computer Science*. 2005, University of KwaZulu Natal: Durban, ZA.
- [115] Bottazzi, D., R. Montanari, and A. Toninelli, *Context-Aware Middleware for Anytime, Anywhere Social Networks*. IEEE Intelligent Systems, 2007.
- [116] Soldatos, J., et al., *A breadboard architecture for pervasive context-aware services in smart spaces: middleware components and prototype applications*, in *Personal and Ubiquitous Computing*. 2007, Springer, London.
- [117] Hebel, J., et al., *Semantic Web Programming*. 2009, Indianapolis, IN: Wiley Publishing, Inc.
- [118] Laube, G.M., G. Reif, and H. Gall, in *Software Evolution and Architecture Lab*. 2007, University of Zurich: Architectural Issues of the Semantic Clipboard as Ontology Mediation Service.
- [119] Grebner, O., E. Ong, and U. Riss, *KASIMIR – Work process embedded task management leveraging the Semantic Desktop*. 2007, SAP Research.
- [120] Sauermann, L., A. Bernardi, and A. Dengel, *Overview and Outlook on the Semantic Desktop*, in *Semantic Desktop*. 2007, Knowledge Management Department, German Research Center for Artificial Intelligence DFKI GmbH: Kaiserslautern, DE.
- [121] Jiang, J. and R. Doraiswami, *Design of a real-time knowledge-based controller with applications in hydraulic turbine generator systems* Journal of intelligent & robotic systems, 1989: p. 229-244.
- [122] Bove, M.J. and J. Mallett, *Collaborative Knowledge Building by Smart Sensors*. 2004, MIT: Boston, MA.
- [123] Walton, C.D., *Agency and the Semantic Web*. 2007, Oxford: Oxford University Press. xvii, 249 p.
- [124] Harroud, H. and A. Karmouch, *Implicit Context-Sensitive Mobile Computing Using Semantic Policies in Autonomic Networking*. 2006, Springer: Berlin ; New York. p. 188-200.
- [125] Sanchez, E., *Fuzzy logic and the semantic web*. 2006, Amsterdam ; London: Elsevier. xvi, 479 p.
- [126] Hinkelmann, K., *Introduction to Knowledge Engineering*, in *Knowledge Engineering*. 2009: Olten, Switzerland.
- [127] W3C, *Semantic Web Rule Language* 2004, W3C.
- [128] Pan, J., et al., *f-SWRL: A Fuzzy Extension of SWRL*. 2007, Department of Computing Science, University of Aberdeen: Aberdeen.
- [129] Lukasiewicz, T., *Expressive probabilistic description logics*. Artificial Intelligence, 2008. **172**(6/7): p. 852-883.
- [130] Lukasiewicz, T., *Fuzzy Description Logic Programs under the Answer Set Semantics for the Semantic Web*. Fundamenta Informaticae, 2008. **82**(3): p. 289-310.
- [131] Damasio, C.V. and J. Pan, *An Approach to Representing Uncertainty Rules in RuleML*. 2004, Centro de Inteligência Artificial, Universidade Nova de Lisboa: Lisbon, PT.
- [132] Mittal, A. and A. Kassim, *Bayesian network technologies : applications and graphical models*. 2007, Hershey, PA: IGI Pub. xi, 356 p.
- [133] Ding, Z. and Y. Peng, *A Probabilistic Extension to Ontology Language OWL*. 2004, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County: Baltimore, MD.
- [134] Mowafi, Y., D. Zhang, and G. Fogionne, *Examining a Bayesian Approach to Personalizing Context Awareness in Ubiquitous Computing Environments*, in *Collaborative Computing: Networking; Applications and Worksharing*, E. Bertino and S. Joshi, Editors. 2008, Springer. p. 44-55.
- [135] Poole, D.L. and A.K. Mackworth, *Artificial intelligence : foundations of computational agents*. 2010, New York: Cambridge University Press.
- [136] Yang, Y. and J. Calmet, *OntoBayes: An Ontology-Driven Uncertainty Model*. 2004, Institute for Algorithms and Cognitive Systems, University of Karlsruhe: Karlsruhe, DE.

- [137] Pearl, J., *Probabilistic reasoning in intelligent systems : networks of plausible inference*. The Morgan Kaufmann series in representation and reasoning. 1988, San Mateo, Calif.: Morgan Kaufmann Publishers. xix, 552 p.
- [138] Fukushige, Y., *Representing Probabilistic Knowledge in the Semantic Web*, in *Uncertainty Reasoning for the World Wide Web Incubator Group Charter*. 2004, W3C.
- [139] Uusitalo, L., *Advantages and challenges of Bayesian networks in environmental modelling in Ecological Modelling*. 2007.
- [140] Yoosoo, O., A. Schmidt, and W. Woontack, *Designing, Developing, and Evaluating Context-Aware Systems*, in *Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on 2007*: Soul, KR. p. 1158 - 1163
- [141] Ejigu, D., M. Scuturici, and L. Brunie, *Semantic Approach to Context Management and Reasoning in Ubiquitous Context-Aware Systems*. 2007, Laboratoire LIRIS-UMR-NCRS: Lyon, FR.
- [142] Brüggmann, H., *Assisting the Discovery and Reuse of Documentbased Knowledge using Semantic Metadata*, in *Department of Information Systems*. 2009, University of Erlangen-Nuremberg: Nuremberg.
- [143] Román, M., et al., *Gaia: A Middleware Platform for Active Spaces*. 2002, Department of Computer Science, University of Illinois at Urbana-Champaign.
- [144] Wang, X., et al., *Semantic Space: An Infrastructure for Smart Spaces*. 2009, Institute for Infocomm Research, National University of Singapore: Singapore.
- [145] Ngo, H.Q., et al., *Developing Context-Aware Ubiquitous Computing Systems with a unified middleware framework*. *Embedded and Ubiquitous Computing, Proceedings, 2004*. **3207**: p. 672-681.
- [146] Chen, H., T. Finin, and A. Joshi, *An ontology for context-aware pervasive computing environments*. *Knowledge Engineering Review*, 2003. **18**(3): p. 197-207.
- [147] Suri, D., et al., *A Multi-Agent Architecture provides Smart Sensing for the NASA Sensor Web*. 2006, Advanced Technology Center, Lockheed-Martin Space Systems Company: Palo Alto, CA.
- [148] Sauer, D., *Automatische Erkennung von Aktionsclustern und deren Zuordnung zu Workflows*. 2009, Friedrich-Alexander-Universität Erlangen-Nürnberg: Erlangen, DE.
- [149] *Aperture Data Wrapper*. NEPOMUK 2007 03-04-2010]; Available from: <http://dev.nepomuk.semanticdesktop.org/wiki/ApertureDataWrapper/Architecture>.
- [150] McMahon, F. *The Future of Artificial Intelligence*. 2009 05-06-2010]; Available from: <http://www.ecademy.com/account.php?id=19183>.
- [151] Viterbo, J., et al., *Context-aware computing and self-managing systems*, in *Chapman & Hall/CRC studies in informatics series*, W. Dargie, Editor. 2009, CRC Press: Boca Raton. p. xxiii, 381 p.
- [152] Segaran, T., *Programming collective intelligence : building smart web 2.0 applications*. 1st ed. 2007, Beijing ; Sebastopol CA: O'Reilly. xxi, 334 p.
- [153] Graham, R., B. Eoff, and J. Caverlee, *Plurality: A Context-Aware Personalized Tagging System*. 2008, Dep't of Computer Science, Texas A&M University.
- [154] Kornienko, S., O. Kornienko, and P. Levi, *Collective AI: context awareness via communication*. 2005, Institute of Parallel and Distributed Systems, University of Stuttgart.
- [155] Soldatos, J., et al., *Agent based middleware infrastructure for autonomous context-aware ubiquitous computing services*. *Computer Communications*, 2007. **30**(3): p. 577-591.
- [156] Mrohs, B., et al., *OWL-SF - A Distributed Semantic Service Framework*. 2005, Competence Center Smart Environments, Fraunhofer FOKUS.
- [157] Schandl, B. and B. Haslhofer, *The Sile Model — A Semantic File System Infrastructure for the Desktop*, Department of Distributed and Multimedia Systems, University of Vienna: Vienna, AT.
- [158] Roman, G.-C., C. Julien, and J. Payton, *Modeling adaptive behaviors in Context UNITY*. 2007, Department of Computer Science and Engineering, Washington University in Saint Louis, United States: Essex, UK.

- [159] Bernardi, A. *NEPOMUK - The Social Semantic Desktop*. 2010 09-06-2010]; Available from: <http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/>.
- [160] Schwarz, S., *A Context Model for Personal Knowledge Management*. 2005, German Research Center for Artificial Intelligence (DFKI GmbH): Kaiserslautern, Germany.
- [161] *Haystack*. 01-06-2010]; Available from: <http://groups.csail.mit.edu/haystack/>.
- [162] Sauermann, L. and S. Schwarz, *Gnowsis Adapter Framework: Treating Structured Data Sources as Virtual RDF Graphs in The Semantic Web – ISWC 2005*. 2005, Springer Heidelberg.
- [163] Sauermann, L., et al., *Semantic Desktop*, in *Social Semantic Web*, A. Blumauer and T. Pellegrini, Editors. 2009, Springer: Heidelberg. p. 337-362.
- [164] Tao, F., et al., *Collaborative Ocean Resource Interoperability: Multi-use of ocean Data on the Semantic Web*, in *The Semantic Web: Research and Applications*. 2009, Springer, Heidelberg: Heraklion, GR. p. 753-767.
- [165] Guigo, R. and T.F. Smith, *Inferring Correlation Between Database Queries: Analysis of Protein Sequence Patterns*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1993, IEEE.
- [166] Iturrioz, J., S.F. Anzuola, and O. Díaz, *Turning the Mouse into a Semantic Device: The seMouse Experience in The Semantic Web: Research and Applications*. 2006, Springer: Heidelberg.
- [167] Scerri, S. *Semanta: your personal email semantic assistant*. in *International Conference on Intelligent User Interfaces*. 2008. Gran Canaria, ES: National University of Ireland Galway.
- [168] Hong, J.-Y., E.-h. Suh, and S.-J. Kim, *Context-aware systems: A literature review and classification*. 2008, POSMIS Lab, Industrial Management Engineering Building, Pohang University of Science Technology.
- [169] OSCAF. *Personal Information Model (PIMO)*. 2009; Available from: <http://www.semanticdesktop.org/ontologies/2007/11/01/pimo/>.
- [170] Shahar, Y., *Knowledge-Based Temporal Interpolation*. 1999, Stanford Medical Informatics, Stanford University.
- [171] Popescu, A., T.F. Serbanuta, and G. Rosu, *A Semantic Approach to Interpolation*. 2006, Department of Computer Science, University of Illinois at Urbana-Champaign.
- [172] Mazzieri, M. and A.F. Dragoni, *A Fuzzy Semantics for Semantic Web Languages*, Università Politecnica delle Marche, Dipartimento di Elettronica, Intelligenza Artificiale e Telecomunicazioni (DEIT).
- [173] Sintek, M., et al. *Nepomuk Representational Language Specification*. 2007 16-06-2010]; Available from: <http://www.semanticdesktop.org/ontologies/nrl/#mozTocId561958>.
- [174] Grimm, S. and P. Hitzler, *Defeasible Inference with Circumscribed OWL Ontologies*. 2008, FZI Research Center for Information Technologies at the University of Karlsruhe: Karlsruhe, DE.
- [175] Bassiliades, N., G. Antoniou, and I. Vlahavas, *A Defeasible Logic Reasoner for the Semantic Web*. 2005, Department of Informatics, Aristotle University of Thessaloniki: Thessaloniki, GR.
- [176] Antonelli, G.A., *Grounded consequence for defeasible logic*. 2005, Cambridge ; New York: Cambridge University Press. xii, 118 p.
- [177] Ma, Y., P. Hitzler, and Z. Lin, *Paraconsistent Reasoning with OWL*. 2006, AIFB, University of Karlsruhe: Karlsruhe, DE.
- [178] Ma, Y., Z. Lin, and Z. Lin, *Inferring with Inconsistent OWL DL Ontology: A Multi-valued Logic Approach* 2006, Department of Information Science, Peking University, Beijing, China: Beijing, CN.
- [179] Schmidtke, H.R. and W. Woo, *A formal characterization of vagueness and granularity for context-aware mobile and ubiquitous computing*. Ubiquitous Computing Systems, Proceedings, 2006. **4239**: p. 144-157.
- [180] Breslin, J., *The social semantic web*. 1st ed. 2009, New York: Springer.
- [181] Connolly, D. *Toward proof exchange in the Semantic Web*. 2006 11-06-2010]; Available from: <http://www.w3.org/2006/09dc-aus/swpf#%2819%29>.

- [182] Hartig, O., *Querying Trust in RDF Data with tSPARQL*, in *The Semantic Web: Research and Applications*. 2009, Springer Heidelberg.
- [183] Keeney, J., D. Lewis, and D. O'Sullivan, *Ontological Semantics for Distributing Contextual Knowledge in Highly Distributed Autonomic Systems*. 2007, Knowledge and Data Engineering Group, School of Computer Science and Statistics, Trinity College Dublin: Dublin, IE.
- [184] Guo, S., et al., *Adaptive Semantic Interoperability Strategies for Knowledge Based Networking*, in *Dublin, IE*. 2009, Knowledge & Data Engineering Group, School of Computer Science & Statistics: Trinity College, Dublin.
- [185] Borbinha, J. and T. Baker, *Research and advanced technology for digital libraries : 4th European Conference, ECDL 2000, Lisbon, Portugal, September 18-20, 2000 : proceedings*. 2000, Berlin ; New York: Springer. xvii, 513 p.
- [186] Strang, T. and C. Linnhoff-Popien, *Location- and context-awareness : first international workshop, LoCA 2005, Oberpfaffenhofen, Germany, May 12-13, 2005 : proceedings*. 2005, Berlin ; New York: Springer. xii, 378 p.
- [187] Abercrombie, N., S. Hill, and B.S. Turner, *The Penguin dictionary of sociology*. 1984, London: A. Lane. 266 p.
- [188] Flyvbjerg, B., *Five Misunderstandings About Case-Study Research*. 2006, Aalborg University, Denmark: Aalborg, DK.
- [189] Institute, C.M.S.E., *CMMI® for Development Version 1.2*. 2006: Pittsburgh, PA.
- [190] Polk, T.A. and C.M. Seifert, *Cognitive modeling*. 2002, Cambridge, Mass.: MIT Press. xxi, 1270 p.
- [191] Sun, J. and Z.H. Wu, *Context reasoning technologies in ubiquitous computing environment*. Embedded and Ubiquitous Computing, Proceedings, 2006. **4096**: p. 1027-1036.
- [192] Zhang, D.Q., et al., *Context reasoning using extended evidence theory in pervasive computing environments*. Future Generation Computer Systems-the International Journal of Grid Computing-Theory Methods and Applications, 2010. **26**(2): p. 207-216.
- [193] Greco, S. and T. Lukasiewicz, *Scalable uncertainty management : second international conference, SUM 2008, Naples, Italy, October 1-3, 2008 ; proceedings*. Lecture notes in computer science,. 2008, Berlin: Springer. xi, 400 p.
- [194] Robinson, P., H. Vogt, and W. Wagealla, *Privacy, security, and trust within the context of pervasive computing*. The Kluwer international series in engineering and computer science SECS 780. 2005, New York: Springer. viii, 167 p.
- [195] Schneider, F.B., *Trusted computing in context*. Ieee Security & Privacy, 2007. **5**(2): p. 4-5.

B. Figure References

Figure 1- The Maturity Evolution within Knowledge-Driven Organizations and the Focus of this Project, © Klaus North, Zimmerli	2
Figure 2 - The Transformation of Data to Wisdom, © XY Scott A. Carpenter	4
Figure 3 - Knowledge Pyramid and Cognitive Energy Application, © Scott A. Carpenter	5
Figure 4 – Knowledge Pyramid and the Up-Shifted Energy of Cognition, © Scott A. Carpenter.....	5
Figure 5 - The Compact Chain of Knowledge Mgmt (CCKM) derived from the Blocks of Knowledge Management, © Probst et al., Zimmerli.....	6
Figure 6 - The Collaboration Matrix Divided by Spatial and Temporal Attributes	8
Figure 7 - On the way to Enterprise Collaborative Knowledge Management, © Schmit et al., Choi et al., Zimmerli	9
Figure 8 - From SLATES to fLATnESseS, © McAfee, © Dion Hinchcliffe.....	10

Appendix B – References

Figure 9 - Locating Area of Appliance of the CCKM, © W3C, © Zimmerli	11
Figure 10 - Four states of information possession, © Hari Srinivas	12
Figure 11 - Thesis Road Map with an Emphasis on Research Stringency	17 -
Figure 12 - Integrating Context and Context-Awareness Requirements into the Software Development Process.....	29 -
Figure 13 - The Simplified Knowledge Assembling Process Illustrates the Process-Integrated Actions against Deficiency	38 -
Figure 14 - Inductive Research Process Embracing the V-Model Software Development Process	44 -
Figure 15 - Three Use Cases featuring Alpiq's Contract Management System (VMS)	46 -
Figure 16 - The 3 Use Cases Packed in a Business Process "Contract Creation"	50 -
Figure 17 - SECS System Architecture with developed and planed (dashed) componentents.....	52 -
Figure 18 - Context Model Graph with a relation to the Context Concept.....	54 -
Figure 19 - Helping to Populate Spatial and Identity in a Semi-Automatic Approach.....	55 -
Figure 20 - Sensors Employed by the Sensors Manager	56 -
Figure 21 - Relations Used to Apply a Rule that Triggers Awareness Support.....	59 -
Figure 22 - The Notification Centre Stores all Notifications.....	61 -
Figure 23 - Tray Icon Messages Issued by the SECS Desktop Client.	62 -

C. Table References

Table 1 - Scenario to Explain the Importance of the Context Aspect "Mission"	20 -
Table 2 – Commonly Agreed Formal Types of Context Models, © Strang and Linnhoff-Popien	25 -
Table 3 - Comparative Matrix of Context Models, Aspects, Model Types and features, © Bolchini et al. .-	26 -
Table 4 – Examples of Sensors, Sensor Types and Their Characteristics	30 -
Table 5 - The Multivariate Criterion List is the basis for testing "fit for purpose" for Alpiq's use cases...-	48 -
Table 6 - Deriving Application Service Requirements from Requirements of Promising Business Service.....-	50 -
Table 7 - Assessing Collaboration Features for Parts of UC1	63 -
Table 8 - Assessing Awareness Support for Parts of UC1	63 -
Table 9 - CMMI Maturity Level Test Conducted by the Focus Group	64 -

D. Glossary

CCKM	Compact Chain of Knowledge Management
CMMI	Capability Maturity Model
CWA	Closed-World Assumption
GPS	Global Positioning System
GUI	Graphical User Interface
JENA	Semantic Web Framework for Java
NIC	Network Interface Card
OWA	Open-World Assumption
OWL-DL	Web Ontology Language – Description Logic
P2P	Peer-2-Peer
RDF	Resource Description Framework

RDFS	RDF Schema
RuleML	Rule Markup Language
SOA	Service-oriented Architecture
SPARQL	SPARQL Protocol and RDF Query Language
SWRL	Semantic Web Rule Language
UAProf	User Agent Profile
UML	Unified Modelling Language

Appendix C – Notice

A. Legal Notice

1. Liability

The University of Applied Science of Northwestern Switzerland (as well as referenced through the university), cannot be held responsible for any element of this paper. The university does not provide any contribution to text, graphics or references unless otherwise noted.

2. Copyright

With the publication of this paper the Swiss, European and international copyright law is applied and makes use of all the claims connected to it, particularly about the disposition. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without prior written permission of the author, not the publisher in case the two differ.

3. Law for trade market protection

Any representation of the University of Applied Science of Northwestern Switzerland used in this paper is registered and protected trademark. It isn't allowed to reproduce, modify or distribute the mentioned representation unless otherwise and explicitly stated.

- End of Document -

- End of Document -