 CMES, vol.76, no.2, pp.133-161, 2011

Rebirth of a Discipline: "Knowledge Engineering"

Ziya Aktas[1] and Semih Cetin[2]

Abstract: The knowledge society has been developed and shaped by amazing improvements during the last two decades. On that development and improvement, social sciences such as psychology or anthropology have also had significant impact as much as real sciences like medicine or engineering, in particular, Information Technology or Information and Communications Technology. The new trends and explosion of knowledge due to Internet and Web technologies have radically changed the way we structure business and its main building block, i.e. "knowledge". Though information/knowledge system development efforts have been regarded formerly as mere information technology activities, now we have been experiencing alternative ways that a business department models, designs and executes the actual business where information technology professionals assist them with the relevant tools and techniques. Hence, all these advancements force us to revisit the classical definition of "knowledge engineering" which was merely "expert system development". This paper points out the needs for this redefinition by reviewing the steps forward in software engineering and how these steps have supported the knowledge engineering so far. Also, authors put forward an improved definition of knowledge engineering, which raises on the pillars of emerging engineering disciplines such as domain, service and business engineering, as well as transdisciplinary approaches. In short, the authors claim the need for the "rebirth" of Knowledge Engineering.

1 Introduction

The whole world has been experiencing amazing developments in "knowledge society" during the last two decades. The "Knowledge Age" has put knowledge to the forefront of every resource for countries, public or private organizations, and individuals as noted by OECD (2006). Hence, Information Technologies (IT) or rather

[1] Department of Computer Engineering, Baskent University – Baglica - 06810 Ankara – Turkey. e-mail: zaktas@baskent.edu.tr
[2] Cybersoft Information Technologies, Silikon Blok No: 18, ODTU Teknokent, 06531 Ankara – Turkey. e-mail: semih.cetin@cs.com.tr

Information and Communications Technologies (ICT) have had an interesting role in this development: on one side it became the key reason of all the developments in the new society, on the other side yet, it has been restructured, reshaped and continuously reconfigured, thus also became one of the results as stated by Aktas (2006).

Since 1990s, Information or Knowledge Society developments have been changing and re-shaping all the countries and institutions on earth. The Internet and electronic commerce have been changing the way business is conducted and have increased the need to improve operations of the enterprise as a whole. A new type of economy, i.e. knowledge-based economy, is growing globally.

Following the Information or Knowledge Society developments in the USA during nineties and in parallel with the Japanese Millennium Project started in the end of 1999, the creation of a Europe of knowledge has been a new strategic goal for the European Union after the European Council Summit in Lisbon of March 2000. The outcome of the Summit became the "eEurope Project", in order to strengthen employment, economy and social cohesion as part of a knowledge-based economy.

Some countries, e.g. OECD countries, are already being compared in "Investment in Knowledge", which is defined and calculated as the sum of expenditure on research and development, on total higher education (public and private) and on software OECD (2006). It is also noted in the same report that for most countries, increases in software expenditure were the major source of increased investment in knowledge. The United States and Japan are moving rapidly towards a knowledge-based economy than EU: since 1994, their investment in knowledge-to-GDP ratios has grown at a higher rate than that of the EU.

Being aware of the key role of education in the new society, Knowledge Society, European Commission began supporting a mobility program named Erasmus Program for European students and academicians in 1987, which is later incorporated under the Socrates Program starting in 1995. It is Europe's education program and its main objective is to build up a Europe of knowledge and thus provide a response to the major challenges of this new century. The Erasmus action and its different activities fit later into the mobility program promoted by the Bologna Process that is initiated in 1999 and followed up by Prague-2001, Berlin-2003, Bergen-2005, London-2007 and Leuven-2009 meetings of the Ministers of Education of the signatory countries of the Bologna Process. The Bologna Process has basically aimed at the creation of a European Higher Education Area by 2010 and promotion of the European system of higher education worldwide.

In the Leuven-April 28 and 29, 2009 conference, the Ministers responsible for higher education in the 46 countries of the Bologna Process convened on to take stock of the achievements of the Bologna Process and to establish the priorities for the European Higher Education Area (EHEA) for the next decade. As it was stated in the official Bologna Process website 2010-2012[1 2 3], The European Higher Education Area (EHEA) was launched along with the Bologna Process' decade anniversary, in March 2010, during the Budapest-Vienna Ministerial Conference. As the main objective of the Bologna Process since its inception in 1999, the EHEA was meant to ensure more comparable, compatible and coherent systems of higher education in Europe. During 1999 - 2010, all the efforts of the Bologna Process members were targeted to creating the European Higher Education Area, that became reality with the Budapest-Vienna Declaration of March, 2010. The next decade would be aimed at consolidating the EHEA. In the Communiqué of the Conference, the following priorities are stated under the heading of **Learning for the future: higher education priorities for the decade to come:** *Social dimension: equitable access and completion, Lifelong learning, Employability, Student-centered learning and the teaching mission of higher Education, Education, research and innovation, International openness, Mobility, Data collection, Multidimensional transparency Tools(such as Quality Assurance and Recognition), Funding.*

The next regular ministerial conference will be hosted by Romania in Bucharest on 27-28 April 2012. The following ministerial conferences will be held in the years 2015, 2018 and 2020.

In the year 2010, IBM started a move that they call "Let's build a smarter planet" and declared "The decade for a smarter planet". In that move, IBM claims that "Our political leaders are not the only ones who have been handed a mandate for change. Leaders of businesses and institutions everywhere have a unique opportunity to transform the way the world works. We find ourselves at this moment because the crisis in our financial markets has jolted us awake. We are seriously focused now on the nature and dangers of highly complex global systems. And this isn't our first such jolt. Indeed, the first decade of the twenty first century has been a series of wake-up calls with a single theme: the reality of global integration. The problems of *global climate change and energy*, *global supply chains for food and medicine*, *new security concerns ranging from identity theft to terrorism* - all issues of a hyper connected world - have surfaced since the start of this decade. The world continues to get "smaller" and "flatter." But we see now that being connected isn't enough. Fortunately, something else is happening that holds new potential: the planet is becoming *smarter*. That is, intelligence is being infused into the way the world

[1] www.ehea.info
[2] http://www.ibm.com/smarterplanet/us/en/smart_grid/ideas/index.html?re=CS1
[3] GridWise Global Forum 2010, Washington, DC, September 21, 2010

literally works — into the systems, processes and infrastructure that enable physical goods to be developed, manufactured, bought and sold. That allows services to be delivered. That facilitates the movement of everything from money and oil to water and electrons. And those help billions of people work and live.

The European Commission has launched the 'EUROPE 2020 Strategy' to go out of economic crisis and prepare EU economy for the next decade on 3^{rd} of March 2010. The Commission identified three key drivers for growth to be implemented through concrete actions at EU and national levels: *smart growth* (fostering knowledge, innovation, education and digital society), *sustainable growth* (making our production more resource efficient while boosting our competitiveness) and *inclusive growth* (revising participation in the labor market, the acquisition of skills and the fight against poverty). Right after declaration of EUROPE 2020 Strategy on March 3^{rd}, the European Commission declared the European Digital Agenda on 19 April 2010[4] which consisted of a series of flagship initiatives. One of these initiatives was expressed as: *A digital agenda for Europe-All Europeans should have access to high speed internet by 2013.*

In her foreword to the *UNESCO Science Report 2010*[5] Bokova, Director-General of UNESCO, was noting that during the last five years while the disparities between countries and regions remained huge, the proliferation of digital information and communication technologies was increasingly modifying the global picture. By making codified information accessible worldwide, it has been having a dramatic effect on the creation, accumulation and dissemination of knowledge, while at the same time providing specialized platforms for networking by scientific communities operating at a global level.

Information System (IS) and Knowledge System (KS) development efforts have been regarded so far as the mere activity of IT departments, stated earlier by Aktas (1987). Also, for the sake of correctness and entirety, business professionals have been incorporated into the life cycle no more than necessary. However, it has been now realized that such an approach does not prevent the chaos or crisis in software, which is first named in 1968 in NATO (1968) and reiterated somewhat recently by a Standish Report (2003). Instead, alternative models have been put into action with Business Process Management (BPM), Service-Oriented Architecture (SOA) and more recently by Cloud Computing trends, where "IT departments" are providing tools and infrastructure and aligned "business departments" are using them to model their businesses, apply the rules, vigorously change the flows, introduce the document exchange triggers, and even monitor and measure the productivity in

[4] i2010 - A European Information Society for growth and employment, DIGITAL AGENDA, Brussels, 19 April 2010

[5] UNESCO Science Report 2010, UNESCO, 2010

business cycles stated by e.g. Aktas & Cetin (2006) and Brown (2007). Such a vision has brought deeper agility to Business Process Engineering (BPE) since the businesses are getting more complex, and decisions require greater analysis and expertise. Hence, much of the responsibility for managing an enterprise has been more and more being delegated to domain experts, or “knowledge workers”, as noted by Awad & Ghaziri (2004).

Software and software engineering of seventies and eighties had been followed by knowledge-based systems (expert systems) or “knowledge engineering” of nineties. Though the cliché terms of “knowledge” or “knowledge-based systems”, and even “knowledge engineering” are not new at all, however we strongly believe that the true “Knowledge Engineering” has not been realized yet. Because, all the previous developments and especially a knowledge explosion due to Internet and Web technologies have been forcing us to rename “Knowledge Engineering” and to have a new content as well as a new meaning as if “rebirth” of this discipline.

Before making a proposal in the paper, the term ‘knowledge’ will be elaborated first in the next section. In a very short historical summary, software development process from its early years to the most recent and modern ways of software engineering is presented next. The software crisis or chaos in software engineering is discussed with a search for the solution. Later, knowledge-based systems or expert systems are summarized and the role of business professionals in software development has been debated next. After presenting the knowledge management, due to the relevance to the topic, transdisciplinary approaches have been included in the paper. Later the proposal has finally been made: *knowledge engineering* with a new content. The paper ends with a conclusion and relevant references.

2 Knowledge and Knowledge Worker

Before proceeding further, it may now be proper to have a closer look at the term “knowledge”. Intellectual capital might be any asset that cannot be measured but is used by a company to its advantage. Knowledge, collective expertise, good will, brand value, and patents fail to show up on conventional accounting documents. In the end, the only competitive edge that sustains is knowledge as stated by Tiwana (2002).

Human knowledge has been classified as explicit or tacit knowledge coined by Nonaka (1995) as the two main types of human knowledge. “Tacit knowledge” is considered to be the most valuable knowledge as stated by Awad & Ghaziri (2004) and Becerra-Fernandez et al. (2004). “Explicit knowledge” is the knowledge codified in documents, books, or other repositories. Tacit knowledge is the knowledge used to create explicit knowledge; the mind-set of individuals that includes intuitions, values, insights and beliefs that stem from experience.

One may define knowledge as "understanding gained through experience or study". It is "know-how" or a familiarity with how to do something that enables a person to perform a specialized task. It may also be an accumulation of facts, procedural rules, or heuristics.

Another way to classify knowledge is to determine whether it is shallow or deep e.g. Awad & Ghaziri (2004):

The knowledge based on reading and training is much different from the knowledge based on practical experience that spans many years. The knowledge based on know-how or accumulated lessons of practical experience are the things needed for building expert systems. Awad & Ghaziri (2004) define the conversion of knowledge between tacit and explicit forms as Figure 1.

Tacit to Tacit (Socialization) Team Meetings and Discussions	**Tacit to Explicit** (Externalization) Dialog within Team Answer Questions
Explicit to Tacit (Internalization) Learn from a Report	**Explicit to Explicit** (Communication or *Combination*) e-Mail a Report

Figure 1: Knowledge conversion

Choo (1996) described that conversion as follows in Figure 2.

Such an organization is called Knowing Organization which possesses information and knowledge so that it is well informed, mentally perceptive, and enlightened. Its actions are based upon a shared and valid understanding of the organization's environments and needs, and are leveraged by the available knowledge resources and skill competencies of its members. The Knowing Organization possesses information and knowledge that confers a special advantage, allowing it to maneuver with intelligence, creativity, and occasionally, cunning. Drucker (1993) has called knowledge, rather than capital or labor, the only meaningful economic resource of the post-capitalist or knowledge society. For him, the right role of management is to ensure the application and performance of knowledge, that is, the application of knowledge to knowledge. The creation and use of knowledge is a particular organizational challenge. Knowledge and expertise is dispersed throughout the organization, and is often closely held by individuals or work units.

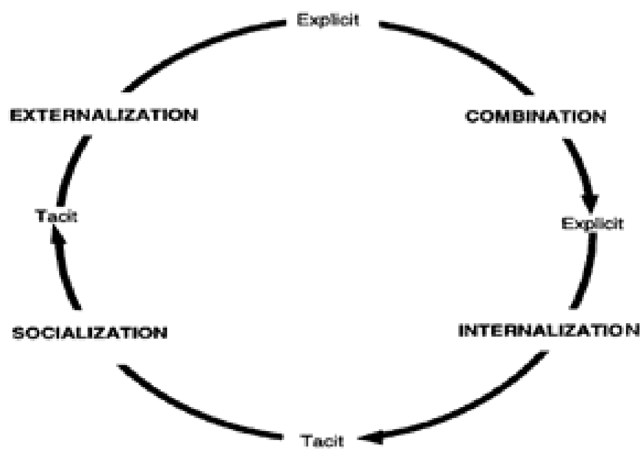

Figure 2: Organizations as knowledge creating enterprises

Davenport (2005) states that Drucker was the first person to describe *knowledge workers* to any substantial degree first in 1969 and he later stated in 1997 that: “The productivity of knowledge and knowledge workers will not be the only competitive factor in the world economy. It is, however, likely to become the decisive factor, at least for most industries in the developed countries.” Davenport, himself in 2005 was saying that” I’ve come to the conclusion that the most important processes for organizations today involve knowledge work. In the past, these haven’t really been the focus of most organizations-improving administrative and operational processes has been easier- but they must be in the future”. Davenport defines a knowledge worker as follows: “Knowledge workers have high degrees of expertise, education, or experience, and the primary purpose of their jobs involves the creation, distribution, or application of knowledge.” And he claims” Knowledge workers think for a living”.

Maruta (2011) tries to identify useful approaches to improve the productivity of non-production work by exploiting information technologies. According to him knowledge creation involves the interaction of three factors: Information, Insight and Knowledge. Knowledge workers perform their assigned tasks by applying or creating knowledge.

Becerra-Fernandez et al. (2004) states that knowledge resides in several different locations. They encompass people, including individuals and groups; artifacts, including practices, technologies, and repositories; and organizational entities, including organizational units, organizations, as well as inter-organizational networks.

3 A brief story of software and Software Engineering

As noted by Aktas & Cetin (2006), if you are at the age of fifty or nearly so, then you will still remember the good old days of data and information. It was all the way simple: raw information was data and the processed data was information. The machines that processed data were called computers or Electronic Data Processing (EDP) machines. The administrative units were Computer Centers or EDP Centers. Programmers of that time were highly prestigious, untouchable and lofty creatures. Punch cards were almost symbol of high status. The young generation of today would not understand how difficult it was for us to differentiate between mainframe and mini-computers. It did not take too long to amplify our misery when we got the new brand of machines: desktop computers, microcomputers, personal computers and so on. From vacuum tubes to integrated circuits, silicon chips etc., everything had been so fascinating technologically and we were like in a dream.

We then woke up and saw the fact that electronic revolution that was going on is fantastic, BUT… Afterwards, we started debating whether computer expertise to be an art or science or engineering. More than that, together with the technological developments in computers, we were on the way to point out information systems. If you point out any system, you would then try to define it and to model it. Eventually that is what we did. Near the end of seventies, the specialty was in deep details so we were talking about databases and database management systems as parts and / or tools of the information systems.

In early 80's it was a kind of excitement for all of us when we first heard that Japanese initiated a project called "Knowledge-Based Machines" or Fifth Generation Computers instead of our good old (!) data processing machines or computers. Yes, we had been using the term "knowledge" in conjunction with Artificial Intelligence (AI) for two or three decades but knowledge-based machines were a new enthusiasm for all of us.

Well, it did not take too long to put the project aside and forget it. But the term "knowledge" has already left a mark in our mind. Soon we realized that the elements of the triple "data-information-knowledge" were not the same but were very closely interrelated. A message may be either data or information or knowledge for a decision-making process depending on the person, time and conditions. Knowledge is known to have the utmost meaning and related to a future intelligent decision making situation.

When it was realized that modeling of information systems is a must we then started talking about modularity and structured system development after a few years of modular programming and structured programming in the late 70's and early 80's.

Before elaborating the differences between classical approach and structured approach for software development, it is worth to recall Information System Life Cycle (ISLC) e.g. Aktas (1987). We then defined the "classical approach" as the one that follows information systems life cycle phases with the proper steps including some tools. Although there may be some variations among individual steps and tools utilized, we expected to have a successful information system at the end.

Classical approach or Waterfall approach for information systems had been a familiar model embodied explicitly in contracts and institutionalized as the development and documentation standards in various types of organizations, more a managerial than a technical tool. Indeed, the classical approach emphasizes documentation essence almost to the exclusion of development.

Yes, documentation is a must for an accomplished information system development, but a list of problems and numerous failures in information systems have shown for years over and over that the documentation alone is not enough for a thriving information system development process. On the contrary, the development process itself must receive more emphasis than documentation, and documentation should be recognized simply as a by-product of a development process.

Limitations of the classical approach such as over intended discrete steps in the life cycle and heavy documentation had yielded a new school of thought called "structured approach" for the complicated information systems development arising in early eighties. The modularity and decomposition principles of structured programming proposed earlier by Parnas (1972) have had supported that new approach.

We recall early 80's not only because of Knowledge-Based Machines endeavor, but also together with the revival of Artificial Intelligence (AI) and also for the debates on the terms informatics (IT-ICT) / software engineering / computer engineering. Instead of Classical Approach for Information System (IS) development, a new modeling and relevant approach emerged from such a debate; and thus "structured approach" seemed as the cure, the panacea.

During eighties, many structured system development methodologies and relevant tools have had appeared e.g. Aktas (1987). Such a methodology would specify how to recognize a good design, how to create a good design, how to communicate over a good design. In addition, an effective methodology would plainly incorporate purposeful structuring and modularity of the system under consideration.

After structured approach of eighties, throughout the nineties, object-oriented software engineering had become the paradigm of choice for many software product builders and for a growing number of information systems and professionals. By that time object technologies had replaced the classical and even structured information system development methodologies.

In an Object-Oriented (OO) approach the objects are manipulated with a set of functions (called methods, operations, or even services) and communicate with each other via a set of messages. Objects are categorized into classes and subclasses and supposed to survive on their own in a distributed space. The intent of object-oriented analysis (OOA) is to define all classes (and the relationships and behavior associated with them) that are relevant to the problem to be solved, e.g. Pressman (2010).

Object-oriented software engineering also follows the same major phases and steps as conventional approaches such as classical and structured approach. Object technologies lead to reuse, and reuse (of program components) leads to more rapid software development and higher-quality programs. Object-oriented software is easier to maintain because its structure is inherently decoupled. In addition, object-oriented systems are more adaptable to changes and measurable to scaling (i.e. large systems can be created by gathering reusable subsystems).

Definitely, the arrival of Internet and Web-based applications by the beginning of 90's, signaled the beginning of a new era through which almost everything is reshuffled and redefined. Then we started using the vocabulary of components, interfaces, services (even Web Services), dot-com, .NET, Java Enterprise Edition (formerly J2EE), Service-Oriented Architecture (SOA), Plain Old Java Objects (POJO), Cloud Computing and many other things in practice. In the same way, Capability Maturity Model (CMM), Software Process Improvement Capability dEtermination (SPICE), Capability Maturity Model Integrated (CMMI), and agile approaches have appeared as the software process management approaches in industry. In addition to the practical and academic thoughts and tools, the Software Factories concept was recently put into reality and Software Product Lines (SPL) became the new relevant technique as stated by Aktas & Cetin (2006). Those process centric capability measurement models would facilitate the acquisition of information systems as well, commented by Cetin (1998). One such frame approach has been introduced in Europe as EUROMETHOD during as early as nineties and it has now turned out to be Information Services Procurement Library – ISPL, see ISPL (2001) after CMMI, SPICE and RUP and UML developments worldwide.

In parallel with these developments in ICT, there are also some attempts in business management to build a bridge on the gap between business and ICT fields. The Business Rules Management (BRM), Business Process Management (BPM), Business Process Orchestration (BPO), and Business Intelligence (BI) have become the fascinating topics. Knowledge Management became the new term coined nearly more than a decade ago.

4 The software crisis

Despite the advancements in ICT summarized in the previous sections, there has been still a very keen effort to search for the better in software. Yes, the software development life cycles were more definite than before, information systems analysis and design techniques were even more capable, software development environments and tools were dramatically enhanced as compared to 60's, 70's and even 80's, but anyhow end users were not so much happy even at the beginning of a new century, say the years 2000's, with the current status of software industry.

This unbridled discontent was mainly stemmed from overall complexity of the software process, variety of applications and the relative immaturity of software development as a vision. Later pointed out by several people as "software crisis", this catastrophic state manifested itself in several ways: projects running over-time and over-budget, low quality end products, and unmet requirements.

Unfortunately, after three decades, even after biting a few silver bullets, we are more or less debating the similar problems of software crisis today but in a different extent, stated by Harel (1992). As an example, take "2003 Chaos Chronicle Report" of Standish Group(2003), which stated: "Over 66% of all software development projects fail or are severely compromised, meaning they are drastically over budget or time, and/or they deliver significantly less scope than promised. Despite the introduction of a multitude of decent technologies, tools, techniques, methodologies, and industry standards organizations, the software industry remains in the state of crisis".

5 Search for a solution

The widespread use of Internet all over the world during the late 90's has blown the strong wind in the sail of deep oceans of software engineering. Known as the "new era of software engineering", Web-based development has introduced enormous amount of reflections into the field. As noted by Cummins (2002), the Internet and electronic commerce have been changing the way business is conducted and have increased the need to improve the operation of the enterprise as a whole. Businesses need to exploit commercial-off-the-shelf (COTS) applications where custom solutions no longer provide a competitive advantage. Undoubtedly, another most important concept was "software architectures". It may be defined as the structure or structures of a program or computing system, which comprise software elements, the externally visible properties of those elements, and the relationships among them, e.g. Bass, Clements & Kazman (2003) and Braude & Bernstein (2011).

Web has changed the life in a way that any Web-based solution today has millions of potential users over there waiting to use your application. This new “e-style” of computing also named the period of 90’s as the “programming in the world” as the last style of computing.

Software quality was one of the major concerns of 90’s if it was not the most significant. Business world had been seriously revolutionized to have an increasing demand on high quantity and quality of services, which should have been met by the information technology solutions. Faced to such a tremendous demand, the software industry acted upon by the extensive vision of “process maturity” and “final product supremacy”.

Software process maturity was not a new concept at 90’s. Improvements in software development life cycles, iterative-incremental techniques, spiral approach, and clean-room development model have all been put into reality beforehand. Even more, the Software Engineering Institute at Carnegie Mellon University has developed the well-known CMM, which has been extensively used for a while for avionics software and government projects since it was created in the mid-1980s. By the end of 1999, SEI has subsequently released a revised version known as the CMMI, which contains 25 key process areas indicating the aspects of product development that are to be covered by company processes as noted by Chrissis et al. (2003). In parallel to CMM and CMMI efforts, the Joint Technical Subcommittee formed by International Standardization Organization (ISO) and International Electro-technical Committee (IEC) has developed a "framework for the assessment of software processes" under the well-known name of SPICE or “ISO/IEC 15504”. SPICE is used much like CMM and CMMI.

Both CMMI and SPICE try to conceptualize the software processes in a well-defined and well-tracked manner. That is why they are mostly known as formal or rather heavy methodologies since they mainly discipline the software development and ease the project monitoring efforts. Nevertheless, such formalism sometimes obstructs the nimbleness of small to medium software organizations noted by Cetin, Tufekci, Buyukkagnici & Karakoc (2006). Agility in software development is then needed badly today to cope with the ever-changing requirements of business departments. Unfortunately, such agile attempts sometimes have been misinterpreted like “anarchy in the software engineering”, thus pioneers of the agility have agreed on establishing “The Agile Alliance” as a non-profit organization for stressing the “discipline” in agility in a much more comprehensive manner producing Agile Manifesto as stated by Braude & Bernstein (2011).

Although people has been only busy with process maturity models to assure the quality in software for more than three decades, the perception of software quality has been slightly changed in the second half of 90’s. Unsurprisingly, this change has been primarily stemmed from the difficulty of Web-based implementations. Then, people started rethinking about the definition of software quality and quantifiable factors affecting it. Such attempts have ended up with the formal definition of software quality and the associated quality factors in ISO/IEC Draft 9126-1. Anticipating the idea of more assembling instead of individual development obligated the practice of proper application partitioning. The foremost attempt was the provision of very clearly defined and independent interfaces among components under the name of “services”. This idea later on emerged a brand new concept called as Service Oriented Architecture (SOA), which includes activities, services and processes. For more granulated autonomy, another school of thought has leveraged the well-known idea of “separation of concerns”. In fact, everything has started with particularly isolating those crosscutting concerns, which yielded the Aspect-Oriented Programming (AOP) models like AspectJ e.g. Laddad (2003). Today Aspect-Oriented Software Development (AOSD) has an extensive vision to identify and separate the aspects of software systems as early as possible and even before coding. So-called “early aspects” targets at applying the aspect-oriented methods even at the requirements analysis phase and propagating to the later stages of design and coding as in the case of OO modeling as noted by Cetin, Altintas & Solmaz (2006).

For more dynamic information systems, the study of software architectures has even stretched the development efforts to the automatic generation of entire code. Somehow like the incarnation of Computer-Aided Software Engineering (CASE), this new school of thought has an objective to model information systems regardless of the architectures and generate the executables for a specific platform. Known as Model-Driven Development (MDD) in general, the Model-Driven Architecture (MDA) proposed by Object Management Group (OMG) has blown a strong wind both in academia and industry.[6]

6 Where are the business professionals?

Until recently, information system developments have been acknowledged as the mere activity of IT or ICT departments. Moreover, for the sake of correctness and entirety, business professionals have been incorporated into the life cycle no more than necessary. However, this model is known to be static and inflexible to the dazzling changes in the very dynamic business environments. Instead, just imagine an alternative model where ICT departments are providing tools and infrastructure and business departments are using them to model their businesses, apply the rules, vigorously change the flows, introduce document exchange triggers, and even

[6] OMG: http://www.omg.org/mda/

monitor and measure the productivity in business cycles. This model should also enable us to understand and design the information systems in terms of very coarse-grained elements, which should not expose the sizzling facts. The technical facts such as objects, components, tables, scripts etc should be hidden but more intelligible services, processes, rules and workflows will be the essential ingredients of information system development noted by Altintas and Cetin (2005).

In leading our way towards the solution for software crisis or chaos, we strongly believe that the basic problem, among others, in software development still stems from the fact that there exists a habitual misconception or disaccord between business and information technology units. All the methodologies, techniques, software standards, process models so far have been primarily devised to resolve this misconception. However, a relatively recent research nevertheless shows that only 54% of the originally defined features are delivered, and even more troubling is the realization that, of those features that are delivered, a full 45% are never used. The requirements that are developed and agreed (and supported by large paper documents) often do not represent what the customer wants or needs - but this isn't apparent until significant cost has been devoted to development and testing based on these flawed requirements. The rework required and the trade-offs expected from the customer are simply mind-boggling as stated by Standish (2003).

Today, business agility urges the use of "enterprise intelligence". Enterprise intelligence is the integration of people and systems, sharing information (and knowledge) collaborating on solutions and plans, and communicating decisions and events so that the enterprise responds intelligently in a manner reflecting enterprise-wide optimization. It is a strategic objective to enable the enterprise to achieve and maintain competitive advantage as noted by Cummins (2002).

Building a bridge between business professionals and ICT departments is the key reason and motivation behind the proposal that we are going to make at the end of our paper. Before we do that, it will be useful to present some more relevant topics in the next sections such as AI and Knowledge-Based Systems, Knowledge Management, and Transdisciplinarity.

7 Artificial Intelligence and Knowledge-Based Systems or Expert Systems

One may define Artificial Intelligence (AI) as the area of computer science that endeavors to build machines exhibiting human-like cognitive capabilities as stated by Becerra et al. (2004). Historically, computers have excelled at performing logical, repetitive tasks such as complex arithmetic calculations or database storage and retrieval. One aspect common across these repetitive tasks is that they are algorithmic in nature. That is, they involve a precise and logically designed set of instructions that yield a single correct output.

Humans, on the other hand, excel at solving problems using symbols to which meaning can be attached. The manipulation of these symbols is arguably the basis of AI: symbol manipulation. One may then redefine AI as the science that provides computers with the ability to represent and manipulate symbols so they can be used to solve problems not easily solved through algorithmic models. Tiwana (2002) defines AI as the science that "...encompasses computational techniques for performing tasks that apparently require intelligence when performed by humans."

Most modern AI systems are founded on the realization that intelligence is tightly intertwined with knowledge. Knowledge is associated with the symbols we manipulate. Many other technologies either currently fall, or at one time fell, under the AI umbrella of technologies. Becerra et al. (2004) classified them as: a) Case-based reasoning; b) Neural networks; c) Inductive learning.

Becerra et al. (2004) states that KBSs are different from general search systems and from conventional software because of three fundamental concepts:

- The use of highly specific domain knowledge;
- The heuristic nature of the knowledge employed from how it is used;
- The separation of the knowledge from how it is used.

Rules (also called productions) are the most popular and natural way to represent the former; frames are typically the most suitable way to represent the latter. The automated reasoning process is closely associated with the inference engine. It defines how the knowledge is exercised to yield the desired results.

8 Knowledge management

Knowledge Management (KM) simply involves the retention of and access to knowledge that is of value to the enterprise. The KM process is concerned primarily with capturing, preserving, and using knowledge as an asset of the enterprises noted by Cummins (2002). According to Tiwana (2002), KM is the management of organizational knowledge for creating business value and generating a competitive advantage. KM is not a business process reengineering. It is about supporting critical processes such as business decisions with the right knowledge at the right time for the right decision maker. Roknuzzaman&Umemoto (2008) defines KM as a relatively new area of investigation which integrates a wide range of concepts, theories and practices from different disciplines. The emergence of knowledge has led to the transformation of post-industrial information society into knowledge based society. Referring to the Gartner Group, they define KM as a discipline that promotes an integrated approach to the creation, capture, organization, access, and use of the information assets of an enterprise. These assets include structured databases, textual information such as policy and procedure documents, and most importantly, the tacit knowledge and expertise resident in the heads of individual employees.

According to Becerra et al. (2004), KM consists of the following four distinct processes:

1. Knowledge Discovery: Combination and Socialization,
2. Knowledge Capture: Externalization and Internalization,
3. Knowledge Sharing: Socialization and Exchange,
4. Knowledge Application: Direction and Routines.

Tiwana (2002) also outlines a few points as an answer to the question: "What Knowledge Management is NOT about?" :

1. KM is not knowledge engineering.
2. KM is about process, not just digital network.
3. KM is not about building a smarter intranet.
4. KM is not about a one-time investment.
5. KM is not about enterprise-wide "infobahns".

As noted earlier, knowledge can be either explicit or tacit. By definition, explicit knowledge is already captured in an understandable form. This is not so for tacit knowledge. It is, therefore, important to elicit tacit knowledge and then capture it in a form that makes it easily manageable. Knowledge elicitation does not need to be limited to the process of developing knowledge-based systems. Humans seek and acquire knowledge in our everyday activities. We read books, magazines, and articles; and we observe others perform tasks. More significantly, we often ask questions of knowledgeable people, for example, at the auto parts store, the computer store, the hardware store, and the travel agency. We occasionally take classes to accelerate our learning process. All these activities are part of the process of managing knowledge.

Awad & Ghaziri (2004) states that Knowledge Management is rooted in many disciplines, including business, economics, psychology, and information management.

It is the ultimate competitive advantage for today's firm. Knowledge management involves people, technology, and processes in overlapping parts. Figure 3 depicts the KM defined by Awad & Ghaziri (2004).

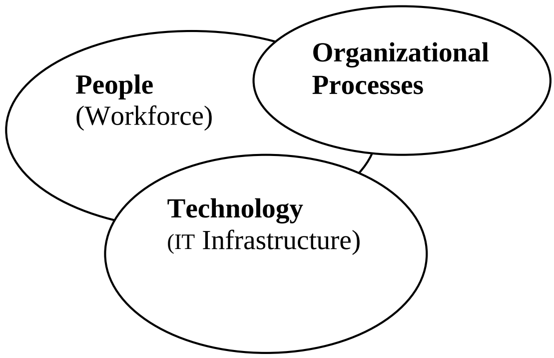

Figure 3: Knowledge management

According to Awad & Ghaziri (2004), up to 95% of information in an organization is preserved as tacit knowledge. It is the fuel or raw material for innovation- the only competitive advantage that can sustain a company in an unpredictable business environment. It is not intended to favor expert systems of the early 1990s, when computers were programmed to emulate human experts' thought processes. The goal is to present a balanced view of how computer technology captures, distributes, and shares knowledge in the organization by linking human experts and documented knowledge in an integrated KM system. The goal is for an organization to view all its processes as knowledge processes. This includes knowledge creation, dissemination, upgrade, and application toward organizational survival.

An alternative definition of KM is also given by Awad & Ghaziri (2004) as "KM is a conscious strategy of getting the right knowledge to the right people at the right time; it is also helping people share and put information into action in ways that strive to improve organizational performance".

The increasing dominance of knowledge as a basis for improving the effectiveness triggered many companies to find the means for utilizing the knowledge they have gained from previous experience. Awad & Ghaziri (2004) outlines the major challenges in building KM Systems as:

1. *Culture*: getting people to share knowledge;

2. *Knowledge evaluation*: assessing the worth of knowledge across the firm;
3. *Knowledge processing*: precisely documenting how decisions are reached;
4. *Knowledge implementation*: organizing the knowledge and integrate it with the processing strategy for final deployment.

The key differences and the similarities of the System Life Cycles of information systems and KM systems are also analyzed by Awad & Ghaziri (2004). It thus indicates that KM is not the Knowledge Engineering that is defined as the knowledge-based systems. Referring to Yang (2009) for another discussion, one may state that KM is a management theory which emerged in the 1990s, since 1995 to be exact. It seeks to understand the way in which knowledge is created, used and shared within organizations. KM comprises a range of practices used by organizations to identify, create, represent, and distribute knowledge for reuse, awareness and learning. Most large companies have resources dedicated to Knowledge Management, often as a part of "Information Technology" or "Human Resource Management" departments, and sometimes reporting directly to the head of the organization.

As effectively managing information is a must in any business, Knowledge Management is already a multi-billion dollar worldwide market. There has been an increased focus in recent years on knowledge engineering, particularly in academic institution and enterprises. Artificial intelligence has a relatively long history in dealing with knowledge from both a theoretical and practical perspective. The many influences on artificial intelligence (e.g. philosophy, psychology and linguistics) bring to it a rich heritage of ideas and a sound foundation in applied field. Over the past 25 years, knowledge engineers have developed a number of principles, methods and tools that have made knowledge acquisition an efficient and effective activity. Interestingly, many of principles, methods and tools of knowledge engineering may have relevance to knowledge management.

9 Transdisciplinary approaches

Exploring the needs of transdisciplinary approaches in engineering solutions is not new at all. Even, the International Center for Transdisciplinary Research (CIRET) is a non-profit organization, located in Paris and founded in 1987. The aim of that organization is to develop research in a new scientific and cultural approach - the transdisciplinarity - whose aim is to lay bare the nature and characteristics of the flow of information circulating between the various branches of knowledge.

European Universities Association (EUA) is a European organization that tries to bring all the European universities for the creation of Knowledge Europe as a part of Knowledge Society of Europe. The most important topics of EUA in the recent years have been interdisciplinary and/or transdisciplinary research and education in European Universities. The Society for Design and Process Science (SPDS) especially for transdisciplinary topics has been founded already nearly two decades ago in the USA, the researchers and educators, our colleagues, therefore deserve also a special attention and congratulation referring to Ertas (2006).

Noting the constantly growing complexity of problems in various fields such as environmental sciences, life sciences, ICT, technology and society, materials science, mobility and transport etc., and also realizing the fact that it is almost impossible to generate comprehensive solutions from a single discipline only, then inter- and transdisciplinary work becomes more and more inevitable. Referring to Wikipedia, one may define transdisciplinarity as a principle of scientific research and intradisciplinary practice that describes the application of scientific approaches to problems that transcend the boundaries of conventional academic disciplines. Any phenomena, such as the natural environment, energy, and health, may be referred to as transdisciplinary or approached and better understood through a process of transdisciplinary modeling. Transdisciplinarity can also be found in the arts and humanities.

A similar concept is interdisciplinarity which usually refers to collaborative projects in which scientists from several fields work together. Interdisciplinarity or intradisciplinarity is considered to be different than transdisciplinarity. Interdisciplinarity concerns the transfer of methods from one discipline to another. Like pluridisciplinarity, interdisciplinarity overflows the disciplines but its goal still remains within the framework of disciplinary research.

As the prefix "trans" indicates, transdisciplinarity concerns that across the different disciplines, and beyond each individual discipline. Its goal is the understanding of the present world, of which one of the imperatives is the overarching unity of knowledge. We strongly believe that such an approach will have a particular contribution to the knowledge society development in the whole world. As in the case of disciplinarity, transdisciplinary research is not antagonistic but complementary to multidisciplinarity and interdisciplinarity research. Transdisciplinarity is nevertheless radically distinct from multidisciplinarity and interdisciplinarity because of its goal, the understanding of the present world, which cannot be accomplished in the framework of disciplinary research. The goal of multidisciplinarity and interdisciplinarity always remains within the framework of disciplinary research as noted by Ertas et al. (2003).

Finally, thinking aloud, the need for earthquake engineering among the disciplines of civil and geology engineering, geological sciences, even, health sciences, social sciences, media and communications; a very new topic with a not- new name, knowledge engineering, among the disciplines of computer sciences/engineering, expert systems (knowledge-based systems or early 'knowledge engineering'), management, business, business engineering, knowledge management, artificial intelligence etc., one would feel how timely and proper will then be a transdisciplinary approach for such topics, fields and new disciplines.

10 A proposal: Knowledge Engineering 'Reborn'

As noted by Cummins (2002), businesses have been becoming more complex and decisions require greater analysis and expertise. Therefore, much of the responsibility for managing an enterprise has been more and more being delegated to specialists, or "knowledge workers". We propose in this paper that these workers are named "knowledge engineers".

Knowledge Management is the distribution, access and retrieval of unstructured information about human experiences between interdependent individuals or among members of a workgroup. It involves identifying a group of people who have a need to share knowledge, developing technological support that enables knowledge sharing as well as creating a process for transferring and disseminating knowledge. (e.g. Wikipedia, Knowledge Management)

The knowledge is increasing at an astronomical rate. The management of such a fabulous level of knowledge is not a simple task today and requires systematic engineering techniques. Consequently, the early definition of "Knowledge Engineering" of AI and relatively recently proposed Knowledge Management alone are no longer applicable to the situation. Knowledge Engineering should now involve acquiring, formalizing and refining the knowledge. Knowledge engineers are therefore, concerned with building and presenting knowledge as the knowledge system responds to the needs of the user (whether human or machine) and allows a connected understanding of the topic rather than a fragmented understanding of pieces of knowledge in isolation. This also implies the logical and intelligent navigation of information / knowledge.

Nowadays, information systems are projected to include and manage the knowledge, thus information system development should be more or less a sort of knowledge engineering practice as well. However, the classical separation of concerns for business development and information system development of the past is not capable today for generating the elegant and successful information systems. At every step of information system development life cycle, business experts and information technology experts should work together in a synergy to avoid tangling relationships, which amplifies the severe communication and lack of understanding. In that sense, existing process maturity models, pure OO methods, agile approaches and even supporting tools are not adequate enough. Definitely, there is a need for a new approach.

The separation of concerns at this conceptual level is the hot topic of software industry today. It may be performed dividing into three as: “business services”, “business processes” and “business rules”. Subsequently, business services will be implemented with an Enterprise Service Bus (ESB – a sort of SOA engine), business processes with a Business Process Model (BPM) engine, and finally business rules with a Business Rules Model (BRM) engine, respectively. The only missing thing in this picture is the “Business to Business (B2B)” integration, which will be implemented with Business Process Orchestration (BPO) engines. Such an imaginary model will provide a meta-model for business experts of business departments to define their own model and the information technology departments are expected to design and implement these four aforementioned engines and the others as expected to build the needed bridge on the gap between IT and Business Units.

We strongly envisage that this imaginary model will constitute the foundation of contemporary “Knowledge Engineering”.

David Garlan and Mary Shaw from Carnegie Mellon University have used another statement in their lecture notes: engineering enables ordinary people to do things that formerly required virtuosos. In addition to classical engineering disciplines, there are some new engineering fields relevant to knowledge such as:

1. Business Engineering
2. e-Business Engineering
3. Enterprise Engineering
4. Biomedical Engineering
5. Data Engineering
6. Knowledge Engineering (BUT ...)

Nearly twenty years ago, the term “Knowledge Engineering” started to be used as a part of AI concerned with the principles, methods and tools for acquiring knowledge and developing knowledge-based systems. Relatively recently, Preece et al. (2001) stated that “In the 1990s, knowledge engineering emerged as a mature field, distinct from but closely related to software engineering. Among its distinct aspects are a range of techniques for knowledge elicitation and modeling, a collection of formalisms for representing knowledge, and a toolkit of mechanisms for implementing automated reasoning.”

More recently we have been talking about "knowledge workers" in various occasions. We, therefore, believe that it will be very timely to propose a new context for "Knowledge Engineering" to include not only AI but also business engineering, e-business engineering, computer science/engineering, software engineering, ICT and knowledge management. A similar claim had already been put by Solvberg & Kung (1993) by introducing the term "information systems engineering".

An interesting discussion is given by Kendal & Creen (2007) as follows: "The terms 'knowledge management' and 'knowledge engineering' seem to be used as interchangeably as the terms data and information used to be. The term 'manage' relates to exercising executive, administrative and supervisory direction, whereas, to engineer is to layout, construct or contrive or plan out, usually with more or less subtle skill and craft. The main difference seems to be that the (knowledge) manager establishes the direction the process should take, whereas the (knowledge) engineer develops the means to accomplish that direction. We should therefore find knowledge managers concerned with the knowledge needs of the enterprise, e.g. discovering what knowledge is needed to make decisions and enable actions. They should be taking a key role in the design of the enterprise and from the needs of the enterprise they should be establishing the enterprise level knowledge management policies. On the other hand, if we were to look in on the knowledge engineers we should find them concerned with data and information (and knowledge) representation and encoding methodologies, data repositories, etc. The knowledge engineers would be interested in what technologies are needed to meet the enterprise's knowledge management needs. The knowledge engineer is most likely a computer scientist specializing in the development of knowledge bases but a knowledge manager may be the chief information officer or the person in charge of the information resource management." One may read that part considering the more general concept that we propose for knowledge engineering term and we feel that there is a very close parallelism.

Though there are dozens of terms ascribed to knowledge leaders by consulting firms, the five main categories of knowledge leadership in the corporations are introduced by Bergeron (2003):

1. Chief Knowledge Officer (CKO)
2. Knowledge Analyst
3. Knowledge Engineer
4. Knowledge Manager
5. Knowledge Steward

Thus, it will be right time to open a discussion for the content of a Knowledge Engineering program especially to include the following specialization topics among others:

1. Knowledge Management and Engineering
2. Artificial Intelligence
3. Software Engineering
4. Domain Engineering
5. Data Bases and Data Mining
6. Knowledge-based Systems
7. Knowledge Elicitation

The way we redefine “Knowledge Engineering” is demonstrated in Figure 4.

“Software Engineering” rising on the shoulders of “Computer Engineering” for the past 40 years has been enriched by the early birth of systematic classification and manipulation of computer-based knowledge using “Artificial Intelligence (AI)” and “Expert Systems” during the last two decades. Therefore, it used to be the common understanding that “Knowledge Engineering” has fully covered the structural knowledge on top of AI and Expert Systems. However, the recent advancements like Ontology-Based Software Engineering (OBSE) and Model Driven Development (MDD) in software engineering have led the “knowledge workers” to concentrate more and more on modeling the specific “domain know-how”.

These advancements have formed the six basic pillars of “Knowledge Engineering” given in Figure 4 to leverage epistemology more in business execution:

1. *Requirements engineering* deals with exploring the exact needs of systems and various stakeholders, where functional and non-functional (quality related) issues should be extracted, elicited, and associated correctly for a better knowledge management.
2. *Domain engineering*, as an emerging discipline, manages the reusability of knowledge through commonality and variability models in software product design. Using systematic techniques such as Feature-Oriented Domain Analysis (FODA) and Feature-Oriented Reuse Method (FORM), Domain Engineering aims to help industrialize the software production process.

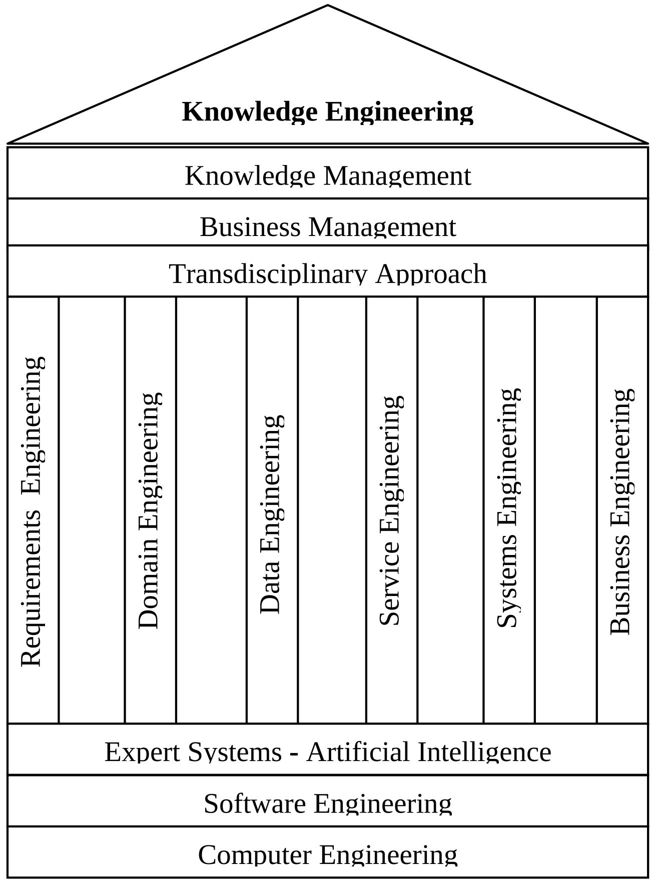

Figure 4.The Foundations and Six Pillars of Knowledge Engineering

3. *Data engineering* approaches to classification of raw data in a way that proper level of knowledge can be extracted and structured by suitable mining techniques, which is essential to transform the data into knowledge within an extremely growing size of information in the World Wide Web era.

4. *Service engineering* is required to structure the huge amount of data systematically in a cross- business environment for a loosely-coupled management of knowledge, where new trends such as "Web 2.0" and "Software as a Service (SaaS)" concepts are pushing enterprises every day to understand the need for epistemology in their daily businesses.

5. *Systems engineering* is an interdisciplinary way to manage the “enterprise knowledge” by means of integration of disciplines within a business or across businesses. It integrates other disciplines and specialty groups into a team effort, forming a structured information management process that proceeds from concept to production and later to operation and disposal.

6. *Business engineering* leverages the rest of five pillars to use knowledge for creating more business opportunities through Business Process Management possibly with the segregation of business rules, processes, services, persistence, and presentation models.

We feel that we are lucky to have the job title "Knowledge Engineers" already being used relevant to knowledge-based systems or expert systems for nearly twenty years. What we propose now in short is to use it instead of “knowledge workers”, “information engineers” or “information systems engineers”. We also propose to expand the previous content of “Knowledge Engineering” to include in parts, computer engineering, software engineering, business management and knowledge management. Additionally, in addition to expert systems and the six pillars introduced by Figure 5, we define “Knowledge Engineering” as a new synthesis of “Transdisciplinary Approach” for better “Business Management” and “Knowledge Management”. This vision shifts the existing focus of “Knowledge Engineering” from “domain knowledge” to futuristically significant “knowledge domain” in which the “Knowledge Engineering” will try to model “ontology of ontologies” across transdisciplinary domains.

11 Conclusions

It has been a fascinating experience for both of us to overview especially the last three decades of software development process. Although there have been many attempts over and over during these years to improve the process it is dramatic that we are still talking about “software crisis” or “chaos in software development”. We therefore believe that it may be a proper time to initiate a debate on the curriculum content of a new engineering discipline named “Knowledge Engineering” as a result of a transdisciplinary approach that will be especially over Computer Science/Engineering or simply ICT, and Business Management and Engineering disciplines.

We strongly believe that it will be the Knowledge Engineering that will put sand-gravel and steel, the data - information -knowledge and wisdom into the construction of buildings, knowledge systems; using cement - software components. The rest are just basic process modeling issues and pure technology.

Topic is interesting yet quite extensive. It is almost impossible to include all the facets in a single paper. It is our only hope that this paper stirs an attention and provides a discussion platform for the future of our professions. We know and understand the difficulty to change the meaning of a term, say 'knowledge engineering' after nearly thirty years. But again all these years may become the reason to make the change as a proof of imminent need.

References

Addis, T. R. (1986): *Designing Knowledge-Based Systems*. Kogan Page Ltd. London.

Aktas, Z. A. (1987): *Structured Analysis and Design of Information Systems*. Chapters 1-4. Prentice- Hall.

Aktas, Z. A. (2006): *Keynote Panel: Transdisciplinary Collaboration in Advance Education & Research.* Integrated Design & Process Technology (IDPT) Conference. Society for Design & Process Science (SDPS), San Diego, California.

Aktas, Z. A. & Cetin, S. (2006): *We Envisage the Next Big Thing*. Pages 404-411.Integrated Design & Process Technology (IDPT) Conference. Society for Design & Process Science (SDPS), San Diego, California.

Aktas, Z. A. (2007): *Opening Speech.* Integrated Design & Process Technology (IDPT) Conference. Society for Design & Process Science (SDPS). Antalya, Turkey.

Alexander, I. (1987): *Designing Intelligent Systems*. Kogan Page Ltd. London.

Altintas, I. & Cetin S. (2005): *Integrating a Software Product Line with Rule-Based Business Process Modeling*. Trends in Enterprise Application Architecture. VLDB Conference 2005. LNCS 3888. Pages 15-28.

Awad, E. & Ghaziri, H. M. (2004): *Knowledge Management*. Pearson / Prentice Hall.

Bass, L., Clements, P. & Kazman, R. (2003): *Software Architecture in Practice*. Addison-Wesley.

Booch, G., Rumbaugh, J. and Jacobson, I. (1999): *The Unified Modeling Language User Guide.* Addison-Wesley.

Braude, E.J. and Bernstein, M.E. (2011): *Software Engineering.* 2nd Ed., J. Wiley.

Brown, A. (2007): *Delivering Successful Software Solutions: Lessons for Success*. Invited lecture. Third National Software Engineering Symposium, Ankara, Turkey.

Becerra-Fernandez, Gonzalez, I. A. & Sabharwal, R. (2004): *Knowledge Man-*

agement. Parson / Prentice Hall.

Bergeron, B. (2003): *Essentials of Knowledge Management*. J. Wiley.

Booch, G. (1991): *Object-Oriented Design*. Benjamin Cummings.

Booch, G., Rumbaugh, J. and Jacobson, I. (1999): The Unified Modeling Language User Guide. Addison-Wesley, USA.

Boggs, W. and Boggs, M. (2002): Mastering UML with Rational Rose 2002. SYBEX, USA.

Cetin, S. (1998): *An Information Systems Acquisition Method for Turkey and EUROMETHOD*, Ph. D. Thesis. Department of Computer Engineering. Middle East Technical University. Ankara, Turkey.

Cetin, S. , Altintas, N. I. & Solmaz, R. (2006): *Business Rules Segregation for Dynamic Process Management with an Aspect-Oriented Framework*. Dynamic Process Management Workshop at Business Process Management (2006): Conference, LNCS 4103. Vienna, Austria.

Cetin, S., Tufekci, O., Buyukkagnici, B. & Karakoc, E. (2006): *Lighthouse: An Experimental Hyperframe for Multi-Model Software Process Improvement*. EuroSPI2-2006, European Software Process Improvement and Innovation Conference. Joensuu - Finland.

Chrissis, M. B., Konrad, M. & Shrum, S. (2003): *CMMI: Guidelines for Process Integration and Product Improvement*. The SEI Series in Software Engineering series. Addison-Wesley.

Clements, P. & Northrop, L. (2002): *Software Product Lines*. Addison-Wesley.

Coad, P. & Yourdon, E. (1991): *Object-Oriented Analysis*. Prentice Hall. 2nd Edition.

Cummins, F. (2002): *Enterprise Integration*. J. Wiley.

Davenport, T.H. (2005): *Thinking for a Living: How to Get Better Performances and Results from Knowledge Workers.* Harvard Business School Press, Boston, Massachusetts.

Dijkstra, E. W. (1972): *The Humble Programmer*. Communications of ACM 15, 10. pp. 859-866.

Drucker, P. (1993): *Post-Capitalist Society*. Harper Collins, New York.

Emam, K., Drouin, J. N. & Melo, W. (1997): *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*. Wiley-IEEE Computer Society Pr.

Erl, T. (2004): *Service-Oriented Architecture*. Prentice-Hall.

Ertas, A. (2006): *Personal Story: A Long and Narrow Road to Success*. A private communication.

Ertas, A., Maxwell, T., Rainey, V.P. and Tanik, M. M. (2003): *Transformation of Higher Education: The Transdisciplinary Approach in Engineering*. IEEE Transactions on Education, Vol.46, No.2, May 2003. Pages 289-295.

Gamma, E. et al. (1994): *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Giarratano, J. & Riley, G. (2005): *Expert Systems*. Thomson, Canada.

Harel, D. (1992): *Biting the Silver Bullet*. Computer, Vol.25, No.1, January. Pages 8-20.

ISPL. (2001): *Information Services Procurement Library*. http://projekte.fast.de/ISPL/.

Kendal, S. & Creen, M. (2007): *An Introduction to Knowledge Engineering. Springer, 2007.*

Knuth, D. E. (1968–1973): *The Art of Computer Programming*. Addison-Wesley.

Laddad, R. (2003): *AspectJ in Action*. Manning, USA.

Maruta, R. (2011): *Transforming knowledge workers into innovation workers to improve corporate productivity.* Knowledge Based Systems. doi:10.1016/j.knosys.2011.06.017.

Medeni, T. (2008): *Knowledge Engineering and a Proposed Curriculum.* MSc Thesis, Cankaya University, Department of Computer Engineering.

NATO, Science Committee. (1968): *NATO Software Engineering Conference*. Garmisch, Germany.

Nonaka, I. & Takeuchi, H. (1995): *The Knowledge-Creating Company*. New York: Oxford University Press, Inc.

OECD. (2006): *Fact Book – 2006*. Science & Technology / Investment in Knowledge.

Parnas, D. L. (1972): *On the criteria to be used in decomposing systems into modules*. Communications of the ACM, v.15 n.12. Pages 1053-1058.

Preece, A., Sleeman, D., Curry, D., Meany, N. & Perry, P. (2001): *Better knowledge management through knowledge engineering*. IEEE Intelligent Systems, Vol. 16, No. 1, January/February, Pages 36-43.

Pressman, S. (2005): *Software Engineering*. Mc-Graw Hill, 6th Edition.

Roknuzzaman, Md. and Umemoto, K. (2008): *Knowledge Management's Relevance to Library and Information Science: an Interdisciplinary Approach*. Journal of Information & Knowledge Management, Vol. 7, No. 4, 279–290.

Shaw, M. (1990): *Prospects for an engineering discipline of software*. IEEE Software, Volume 7, Issue Nov 1990. Pages 15 - 24.

Solvberg, A. & Kung, D. C. H. (1993): *Information Systems Engineering*. UNIT/NTH, 7034. Trondheim, Norway.

Sommerville, I. (2007): *Software Engineering*. Addison-Wesley, 8th Edition.

Standish Group. (2003): *2003 Chaos Chronicles Report*. http://www.standishgroup.com.

Tiwana, A. (2002): *The Knowledge Management Toolkit*. Prentice-Hall.

Yang, X. (2009): *To Facilitate Knowledge Management Using Basic Principles of Knowledge Engineering.* Proceedings of 2009 Pacific-Asia Conference on Knowledge Engineering and Software Engineering (KESE 2009), Shenzhen, China, Dec. 19 - 20.