

The Grand Challenges in Information Systems

Chandra Amaravadi

School of Computer Sciences, Western Illinois University, Macomb, USA

Email: c-amaravadi@wiu.edu

How to cite this paper: Amaravadi, C. (2022) The Grand Challenges in Information Systems. *Journal of Software Engineering and Applications*, 15, 103-115. <https://doi.org/10.4236/jsea.2022.154005>

Received: February 28, 2022

Accepted: April 8, 2022

Published: April 11, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The field of information systems has historically suffered from a pre-dominance of behavioral approaches. As a result, it is not surprising that in spite of decades of research, dozens of conferences and journals and thousands of researchers, very few breakthroughs have been achieved. This paper advocates a technical approach that focuses on improving existing systems in organizations by viewing them from the point of roles *i.e.* functions fulfilled by IS in organizations. These are understood as: 1) supporting operations, 2) providing information, 3) supporting decision making, 4) providing knowledge, 5) supporting knowledge and clerical work, and 6) supporting organizational design. It is generally agreed that the field's understanding of fulfilling roles "1", "2", "3" and "6" is mature while understanding of areas "3" (for unstructured decisions), "4", and "5" are still incomplete. The challenges in these areas are difficult problems that we refer to as grand challenges and discuss in this paper.

Keywords

Challenges in Information Systems, Information Systems Challenges, Research in IS, IS Research, Technology Support

1. Introduction

The field of information systems was established in 1968 when the University of Minnesota initiated its master's and Ph.D. programs in information systems. With an excess of 3200 scholars, dozens of journals, and thousands of papers, the field has progressed tremendously since its inception. In spite of a large number of scholars and exuberant participation in conferences, there have been few technical breakthroughs that inspire confidence in the discipline. Groundbreaking innovations such as HTML, social media, data analytics, and deep learning have originated from related areas outside the academic field and frequently from industry participants. The debate on "research that matters"

highlights the fact that there is a paucity of IS research that has made an impact on the world [1]. As scholars, we need to collectively acknowledge that we have not produced much useable research during the five decades that the field has been in existence.

Part of the reason is the behavioral approach that predominates in the field [1]. For example, whenever a new technology emerges, rather than study ways of improving it, there are studies that identify critical success factors in its implementation and conclude among other things that top management support is critical to its success. The utilization of information systems in organizations does not present the difficulties that it previously did due to the prevalence of visual interfaces (GUI) and the high level of technological proficiency among the general public. The case study of Siemens Sharenet illustrates this. The ICN division of Siemens was able to develop a very successful knowledge management system that was used by 19,000 employees at its peak [2]. The implementation did not use any of the academic theories pertaining to system usage. In the software industry, user acceptance testing (UAT) is a mandatory part of any release, so there is little value in researching acceptance issues or similar behavioral issues involving technology.

On the other hand, system approaches are far and few in between. One study found that only 15% of research articles in journals used the Software Engineering Approach [3], which is often referred to as the Design Science Approach. Real progress in the field is hard to achieve unless it is directly focused on improving information systems used in organizations. To achieve breakthroughs at the system levels, information systems research should focus on the technologies and how they can support organizations. Roles are a convenient way of focusing this effort. These are widely understood as [4]: 1) support operations, 2) provide information, 3) provide knowledge, 4) support structured decision making, 5) support unstructured decision making, 6) support knowledge and clerical work, 7) support organizational design and strategy. The field's technical/research understanding of "1", "2", "4" and "7" is mature but still incomplete, requiring mopping up operations. For example, the technology to support company operations across organizational boundaries is poorly understood. On the other hand, the IS field is lagging in its understanding of "3", "5", and "6" [5] [6] [7]. Coupled with these are improvements to the software development process, particularly where analyzing and developing specialized systems are concerned. Achieving success in these requires breakthroughs in certain fundamental areas which I will refer to as the "Grand Challenges of IS." These are based on personal experience and represent, it is hoped, the holy grail of information systems research [8]. These are called as such because resolving them would improve the fulfillment of information systems roles. Save for one or two, the majority are recognized problems in related disciplines, so the contribution here is in identifying the agenda for IS researchers.

2. The Grand Challenges

The grand challenges range across a number of different areas from supporting

decision making to file formats to knowledge representation. When these are addressed it will be possible to better support organizations and their employees.

2.1. Supporting Decision Making

DSS is a mature technology in organizations, providing support for structured and semi-structured decisions, mostly in production, operations, marketing and transportation [9]. Support for unstructured decisions is still primitive. In a pathological case, an executive at a Japanese electronics company used a version of “rock, paper or scissors” to make a \$20 m decision [10]. The key to supporting these types of decisions still lies in understanding how they are made. The phases of decision making are widely understood as Intelligence, Design and Choice although these are not necessarily sequential [11]. The Intelligence phase concerns the identification of the problem, an area that has received much attention in the organizational literature as problem formulation. According to [12], managers become aware of problems through informal channels of communication. Then they attempt to pigeonhole the problem into “categories” such as “mission and goals,” “organizational structure,” “resources” etc. [12] [13]. Cognitive scientists believe that categorization is followed by a mental representation of the problem that includes details such as constraints, objects, procedures and operators which enables them to solve it [14]. Both the problem formulation and categorization stages involve retrieving information from stored schemata, an area that technology could support. This stage overlaps with the Design phase of the decision-making process. Choices are made to maximize decision-maker utility, as in the classical literature with experienced managers giving extra attention to see that decision constraints are fulfilled [14]. Understanding these stages will certainly contribute to the state-of-the-art of DSS.

In the meantime, specific contributions are required in three areas. The first is Decision Representation—to develop a graphical method of representing a decision, similar to the representation of a process (see also challenge#9). A universal methodology, like UML for representing alternatives, constraints, information, rationale, micro-decisions, actions and impacts of a decision is required. Along with this, a tool that provides an unconstrained interface to formulate the problem and additionally to find a solution is required. With the tool, a decision-maker should be able to “sketch” decision alternatives, sub-problems, assumptions, weights, etc. and freely hop among these (see [15] for a similar idea). Drawing a diagram with the Microsoft Visio tool, using different symbol sets, is a comparable design metaphor. Such a “doodling” tool is one of the great challenges in IS. The second challenge is modeling subparts of the decision with existing DSS models such as L.P., Probabilistic, or simple algebraic models. For example, can we model the impacts of a decision as a mathematical relationship between the decision objective and a possible action [16]? Is such an approach feasible? What information will the subparts require? Is it available from the decision situation as depicted by the model? How can we aggregate results from these

sub-decisions? If it is not possible to model subparts with DSS models, is it at least possible to have a schemata for each category of problem, in a manner somewhat analogous to word/email templates? This is the third area where research is required.

2.2. Designing Natural Language Interfaces

Natural language interfaces (NLI) will be the ultimate interface to information systems since no menus or user training is required. NLIs have been developed for databases [17], visualization software, medical and tourist information systems [18] [19]. But not all domains lend themselves to NLI applications, and graphical software being one example. So the first question to address is, for what types of software are NLIs best suited to? Perhaps it is not the software as much as certain actions that are suited to NLI. What are those? Assuming this is addressed, other issues arise during the process of interacting with the software [20]:

- 1) Understanding the User's Request—there are two approaches to understanding the user's request. The first is to break the request into parts of speech to understand the subject and object of the request. This presents the usual problems of processing NL which will be addressed in time. The second and more promising approach is to map the request into a pre-defined pattern. The request and its intent have to be understood in the context of the user's previous actions as well his/her current software context in order to select a sequence of actions.

- 2) Mapping the Request into a Sequence of Actions—once the request is understood, the next step is to identify a sequence of actions that will fulfill the objective. Classical issues in means-ends analysis and hierarchical planning arise (see for example [21]). A compromise approach where the complexity is limited should be taken to ensure that these problems do not arise. What then is the right level of complexity that can be handled by the system without running into planning problems? Additional questions that arise are, are there canonical forms for task-execution sequences? Can execution sequences be somehow auto-generated from the software itself to avoid coding them explicitly?

- 3) Determining the User's Context—the most important problem here is identifying the user's present context. Issues arising here are how can context be determined within an application in relation to the task that the user is carrying out? What knowledge should the system have about previous and current actions? About his/her computing environment? If the user is asking for a heading change in a file, how does the system recognize the file the user referring to?

- 4) Error handling and help—the system needs to interact with the user when their request is not understood, incorrect or poorly phrased. What are effective ways of dealing with help and error handling? Should the system present alternative formulations of the request? Suggest a list of functions that the user can use? Or fill in missing parameters from what it knows?

2.3. Providing Task/Functional Support

Task or functional support is concerned with technological support for performing daily tasks. A simple example is a researcher preparing a research paper that has a list of references at the end. He/she ought to be able to ask the system to use IEEE referencing style for formatting them, but this is not possible in current systems without using specialized software. Despite the plethora of information technologies available, office workers are poorly supported in their work [6]. They cannot automatically send emails or collect information from an email message (some of this capability is appearing in contemporary systems). Technology support for professionals is even poorer. It has not evolved past the use of specialized software such as CAD/CAM, LexisNexis database or a business intelligence tool. So an engineer carrying out a design cannot ask the system a question such as “What is the yield point for a 2" thick 10' long I beam?”

Functional support requires four types of capabilities on the part of office systems [6]: 1) Answer requests for information from documents stored in the system, 2) Store and retrieve assorted information such as a vendor offering a particular type of discount, 3) Answer questions about the employee’s domain/work (essentially knowledge management) and 4) Carry out an action or a sequence of actions. The system should support the office worker like an assistant [6].

The first requirement is to answer requests for information that is in the system. The field of information retrieval focuses primarily on retrieving documents using keywords/terms [22], but in this realm, retrieval of parts of documents is an important problem [6]. It is up to IS researchers to identify different scenarios in which this requirement arises and then pursue a solution using techniques from the field of information extraction (see for example [23]). The second capability in the list requires suitable organization of information and perhaps the use of AI technologies [5]. The third requirement, question answering capability requires knowledge to be encoded using a suitable representation method (challenge#6) as well as natural language interfaces (challenge#2). Although there are some proposals [24] [25], research is still required to operationalize these schemes to develop technologies for a production implementation. Lastly, the requirement to provide clerical support for mundane tasks requires re-architecting office systems so that its functionality is available as services to end-users as well as other applications [6]. For example, it should be possible to send an email while editing a document or to schedule a meeting from an email client. Here also different usage scenarios need to be identified. This and NL capability (challenge#2) will allow requests such as “Email the Frankfurt report to all VPs of the company” to be fulfilled by the system [6].

2.4. Designing Universal Formats for Information Storage and Retrieval

The information and internet revolutions have spawned a number of different file formats for documents, images and data. For example, there are docx, html,

xml, pdf, txt, and rtf formats for documents and bmp, jpg, png, and gif formats for images. At present, we will confine our discussion to physical file formats for documents only. A variety of applications including workflows, patient records, maintenance records, web searches and IoT (Internet of Things) related communications require standardized archival file formats that have to withstand the test of time. Is such a universal document format possible? XML has been introduced as such a standard but it has several shortcomings [26]. The first is ambiguity in the meaning of tags used since these are user-generated and subject to linguistic problems. Secondly, it lacks mechanisms for encoding relationships between elements *i.e.* other than as a hierarchy of elements [26]. Thirdly XML is verbose and requires a DTD file for interpreting it correctly. Fourthly, it is possible to encode *only* the linear or hierarchical structure of document elements, e.g. “a”, “b”, “c” (linear) or “a” (“b” (“c”)) (hierarchical) where a, b and c are elements of the document. So a table structure is awkward to encode [27]. A similar problem occurs with office forms [6]. Modeling of form components as normalized tables makes it difficult to model operations on them in the same way that hierarchical structure in hierarchical databases made queries difficult [6]. An intermediate representation therefore presents a layer of complexity. A possible solution is to go beyond basic data types (e.g. integer, character, text, etc.), standardize more complex data types such as data ranges, min-max values, list of tasks, meeting minutes, etc. and incorporate them into the interpreter/browser. Conceivably these could be implemented as XML extensions. It will enable any document type to be modeled independently of the technology. For example, document A could be encoded as a + b, while document B could be a + b + c, and document C could be b + d where “a”, “b”, “c” and “d” are components of a document. As a concrete example, “a” could be user profile on a web site and “b” could be a list of patient medications, dosages, etc., so “a + b” would be a patient record. Only a comprehensive use-case analysis can reveal the complex types that are required. This addresses only part of the problem. The second issue is to have a more elegant solution to describe the appearance of the document, that would integrate well with the semantic model. Markup languages such as HTML may be thought of as a possible solution, but in these, data is stored without interpretation. For example, items such as graphs would be stored simply as images rather than with their semantic content, *i.e.* what is the graph about?

2.5. Designing Executive Information Systems

Executive information systems (EIS) have been introduced in the eighties. Their main functions are to provide high-level summarized information (from operational databases) and to provide support for analyzing this information. The information can be internal or external but generally speaking, external information is more valuable than summarized data from operational databases. The data and analysis capability are currently provisioned by Business Intelligence (BI) tools [28] [29]. These can provide summarized information in a variety of

formats with an almost infinite variety of data couplings. BI tools appear to have replaced EIS completely, but some fundamental problems remain even with the addition of data/analysis capability.

The first is access to external information. Most EISs do not provide access to external information [30], presumably to avoid exposing sensitive company information. BI tools tend to be standalone tools with the ability to import data. Top management needs access to soft, non-financial data and strategic information that is currently a gap in EIS capabilities [29]. Regardless of the source and type of information, executives use it to make strategic decisions [31] and this is another bottleneck in the hyper-reactive 21st century business environment. To keep up with this velocity, it is necessary to rely on AI techniques to process and analyze trends; for example, to assess how a company's valuation is affected by a change in regulation. This has been missing in EIS research and is obviously the most important problem in this area. While there has been some research, there are no good models of how an organization or an organizational unit is affected by environmental forces (see also challenge#1). Secondly, there is no support for analyzing environmental information, such as creating scenarios or simulations [31]. The third issue is with respect to filtering the information. Critical success factors (CSFs) or strategic business objectives are often used as development methodologies for executive information [32] but in practice the volume of information is vast and dynamic, making CSFs an unsuitable paradigm. When executives are responding to a specific situation, they will need to obtain and analyze very specific information. For this reason, filtering information is an important research problem. Filtering requires the user's context which is typically ignored in Information Retrieval systems [33]. For executives, the user's context translates into markets, products, customers, etc. but these will not be simple encodings. So there is a need for user representations of context and at the same time, representations of document content are also needed to enable retrieval, spilling into other related challenges (challenges #3 and #6).

2.6. Knowledge Engineering for Professional Knowledge

Professional knowledge is deep knowledge associated with a particular domain such as stock trading, marketing, banking, engineering, etc. [24]. It describes objects, events, actions, situations, concepts, objectives or policies but consists mostly of abstract concepts [24]. In professional knowledge, there are complex relationships between such concepts including mathematical, axiomatic, logical, temporal, structural, etc. Traditionally rules have been used to model this knowledge but these are not useful for representing declarative knowledge such as "a 'trade' (an 'action') being one of 'options', 'stocks', 'bonds' or 'mutual funds' ". This is a typical class-subclass relationship (structural) that is better modeled with declarative schemes. Professional knowledge is filled with abstract concepts that are defined in terms of other abstract concepts, creating a knowledge engineering challenge [24]. In the previous example, "trade", "options", "stocks" etc.

are all abstract concepts with structural relationships among them. There could be elaborations or restrictions on both concepts and relationships such as conditions under which the relationship is valid or some sort of agreement that enforces the relationship. For example, the allowable condition of a trade is that there should be sufficient funds in the account. These funds could consist of “cash” or “unsettled trades” or both. Thus concepts can have elaborations on relationships, but there are additional requirements for the representational mechanism. Firstly, the knowledge stored in the system should be viewable/editable by employees, so it should be at the “conceptual level” rather than at the “implementation level” of Brachman’s knowledge levels. Graphical representations are easier to comprehend so this is a requirement. Secondly, the representation scheme should support the formation of abstractions so that concepts may be defined in terms of other concepts. Class-subclass relationships and concept definitions are common abstraction mechanisms. Thirdly, the scheme should support modularity which takes the form of partitioning. It should not be a monolithic graph that will present difficulties in comprehension, update or assertions of facts. In other words, it should be easy to divide the representation into parts and perhaps store these parts or make assertions independently. Fourthly, the scheme should be extensible, so that new concepts/constructs can be easily added. Lastly, since the amount of information in organizations is large, the techniques should scale up to accommodate volume [5].

2.7. Knowledge Management Support with Technology

Knowledge management (KM) is the explicit management of organizational knowledge, including tools and processes to create, store, access and disseminate organizational knowledge [34]. KM ideas have been in existence for at least two decades now, yet very little progress has been achieved in terms of technology. The prevailing approach in IT is to use knowledge repositories [35] with little consideration being given to the problem of retrieval or the need for keeping thousands of knowledge items up to date [5]. The trend toward virtual organizations and exploding knowledge in high-tech industries exceeds the bounded rationality of organizations, if not their budgets. The leader in addressing KM problems was not academia but Siemens with its Sharenet [2]. The repository approach is not practical if the use case requires answers to fine-grained queries (see also Challenge #2 and #3) such as “What should be the blade angle for a jet engine, if it needs 16,000 lbs of thrust at an altitude of 25,000 feet, facing a head wind of 15 knots?” Such queries can be answered only through AI approaches [5]. But these are poorly developed at best [24]. Progress in representing complex knowledge is needed as outlined in Challenge #6. Since knowledge in a repository has to be updated, it should be both human and machine processable. For this reason, graphical representations should be used [24]. Further, the knowledge should be checked for consistency, so maintaining knowledge integrity is another challenge for researchers. A philosophical issue here is when

knowledge is updated, should it replace existing knowledge or should it be another version of the same knowledge? Would having ten versions of the same knowledge item defeat the purpose of a KM system? The details and functionality of the KM engine should be worked out as well as the best way to enhance the user experience (UX). These issues are definitely within the scope of IS research.

2.8. Achieving Large-Scale Distributed AI within a Smart-Device Ecosystem

The internet of things (IoT) is upon us with billions of smart devices that communicate with other smart devices. Designing applications that rely on IoT and communications with other systems to achieve distributed intelligence will be a major challenge for the future [36]. A car that has been sold in a dealership should be able to communicate to a Dealer Inventory database as well as to the Department of Motor Vehicles to update its registration [3]. How to achieve this without causing chaos or conflicts while preserving security and privacy is the “mother of all problems” and can serve as a gateway for many useful applications.

The basic IoT architecture is a wireless sensor network that collects data from sensors (e.g. a utility meter). This data could be sent to a local data store or a cloud-based server [37]. In some cases, actions can be taken based on the data, such as turning on a pump. For some applications, the data is cleaned and processed while in others such as meter readings, it is collected for other purposes. Data can also be aggregated and mined for patterns/conclusions [38]. There are networking/architectural challenges in having common transport and application protocols given the heterogeneity of sensor/actuator devices and networks [37]. Maintaining security and privacy in a ubiquitous data rich environment is also a major hurdle [39]. For IS researchers, the challenge lies in developing a framework that can facilitate data sharing (at the user level) since this will be a crucial component of distributed AI. The necessary protocols for handling this need to be developed. Issues that arise are: What data is public vs private? What checks have to be carried out before allowing information access from a device such as a health monitor? If a building detects a person with a weapon entering it, can it identify the person and check his/her firearm permit? Can it alert law enforcement if there is no permit? Philosophically, where should the intelligence that carries this out be located? Who should update it?

2.9. Universal Method for Specifying System Requirements

Software engineering is a massive but mature industry. Potential developers are confronted by a bewildering array of tools and methodologies for specifying requirements. Existing methodologies can be classified into two main branches based on whether the development approach is traditional or agile, using object-oriented development [40]. There are a number of methodologies in each of these for specifying the development. In the traditional approaches, there are tools such as DFDs, structure charts, data dictionaries [41], while for the latter

approach there are tools such as UML diagrams, sequence and interface diagrams. Most research approaches have focused on specialized systems or on individual tools. See for example, [42] [43]. There is a compelling need for a uniform and integrated specification method (that is not unwieldy) to document or define: 1) objects and their relationships, 2) processes, 3) process calling sequences, 4) data relationships, 5) decisions 6) interfaces and interface relationships, 7) system features, 8) business rules and 9) organizational knowledge. Such a universal method ensures that the same methodology can be used for different types of systems and additionally, for creating a system's entire documentation in an integrated fashion. UML has been proposed as one such standard, but it has several limitations [44]. Among these is the inability to capture: 1) system-actor interactions (for example, a system informing an actor of its unavailability), 2) organizational context (for example, interactions between actors), 3) use case relationships (for example, part/subpart), 4) state-dependent system behavior (for example, a use case depends on a system's state). Modeling of information flows is also awkward in UML, but is a basic requirement of a specification methodology [44]. A universal methodology that overcomes these limitations and adapted to specifying/documenting different types of systems is required.

3. Conclusion

The field has suffered a behavioral emphasis for a long time as manifested in the design science debate [2]. It can only progress with application to business needs as defined at the outset. Behavioral work was relevant to the industry in the "initiation stage" of information systems in the "70's and "80's, but with the widespread adoption of information technologies by the masses (adults and infants alike!), this assumption is rapidly losing currency. For example, product manufacturers handle their own usability studies rather than utilize findings from academia. Previous works have viewed research issues in a broad sense, such as research being required in Databases, GDSS, etc. [45]. This paper advocates a technical approach to designing systems for organizations. Some of the topics discussed include decision modeling systems, designing natural language interfaces based on understanding simple requests, systems supporting employees' daily tasks, such as filling time sheets, universal data formats based on document algebra, and a universal method of specifying systems among others. Ultimately, all of these will boil down to models of various situations/objects, such as models of decision making, knowledge representation, semantic models of files, etc. It is the task of IS researchers to design these. There are other candidates for the grand challenges such as security and large-scale architectures for organizing data/processes on the cloud, but they appear surmountable with existing methods and cannot be considered grand challenges.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H. and Loos, P. (2010) Memorandum on Design-Oriented Information Systems Research. *European Journal of Information Systems*, **20**, 7-10.
<http://dx.doi.org/10.1057/ejis.2010.55>
- [2] MacCormack, A.D., Volpel, S. and Herman, K. (2002) Siemens ShareNet: Building a Knowledge Network. Harvard Business School Case 603-036.
- [3] Morrison, J. and George, J.F. (1995) Exploring the Software Engineering Component in MIS Research. *Communications of the ACM*, **38**, 80-91.
<http://dx.doi.org/10.1145/213859.214802>
- [4] Valacich, J. and Schneider, C. (2013) Information Systems Today: Managing in the Digital World. 6th Edition, Prentice Hall, Upper Saddle River.
- [5] Amaravadi, C.S. (2005) Knowledge Management for Administrative Knowledge. *Expert Systems*, **22**, 53-61. <http://dx.doi.org/10.1111/j.1468-0394.2005.00294.x>
- [6] Amaravadi, C.S. (2014) Office Information Systems: A Retrospective and Call to Arms. *Journal of Software Engineering and Applications*, **7**, 700-714.
<http://dx.doi.org/10.4236/jsea.2014.78065>
- [7] Huttenegger, G. (2004) Knowledge Management and Information Technology: Goals/Problems, Practical Approaches and Proposal Solution. Doctoral Dissertation, University of Vienna, Vienna.
- [8] Amaravadi, C.S. (2016) Ten Grand Challenges in Information Systems. Australian Conference on Information Systems, Workshop on Grand Challenges, Wollongong.
- [9] Eom, S. and Kim, E. (2006) A Survey of Decision Support System Applications (1995–2001). *Journal of the Operational Research Society*, **57**, 1264-1278.
<https://doi.org/10.1057/palgrave.jors.2602140>
- [10] Vogel, C. (2005) Rock, Paper, Payoff: Child's Play Wins Auction House an Art Sale. New York Times.
- [11] Mintzberg, H., Raisinghani, D. and Théorêt, A. (1976) The Structure of Unstructured Decision Processes. *Administrative Science Quarterly*, **21**, 246-275.
<http://dx.doi.org/10.2307/2392045>
- [12] Lyles, M. and Mitroff, I. (1980) Organizational Problem Formulation: An Empirical Study. *Administrative Science Quarterly*, **25**, 102-119.
<http://dx.doi.org/10.2307/2392229>
- [13] Smith, G.F. (1995) Classifying Managerial Problems: An Empirical Study of Definitional Content. *Journal of Management Studies*, **32**, 679-706.
<http://dx.doi.org/10.1111/j.1467-6486.1995.tb00794.x>
- [14] Nokes, T.J., Schunn, C.D. and Chi, M.T.H. (2010) Problem Solving and Human Expertise. *International Encyclopedia of Education*, **5**, 265-272.
<http://dx.doi.org/10.1016/B978-0-08-044894-7.00486-3>
- [15] Acar, W. and Druckenmiller, D. (2006) Endowing Cognitive Mapping with Computational Properties for Strategic Analysis. *Futures*, **38**, 993-1009.
<http://dx.doi.org/10.1016/j.futures.2005.12.018>
- [16] Saaty, T.L. (1978) Modeling Unstructured Decision Problems—The Theory of Analytical Hierarchies. *Mathematics and Computers in Simulation*, **20**, 147-158.
[http://dx.doi.org/10.1016/0378-4754\(78\)90064-2](http://dx.doi.org/10.1016/0378-4754(78)90064-2)
- [17] Androutopoulos, I., Ritchie, G.D. and Thanisch, P. (1995) Natural Language Interfaces to Databases—An Introduction. *Natural Language Engineering*, **1**, 29-81.

- <http://dx.doi.org/10.1017/S135132490000005X>
- [18] Rector, A.L., Solomon, W.D., Nowlan, W.A, Rush, T.W., Zanstra, P.E. and Claassen, W.M. (1995) A Terminology Server for Medical Language and Medical Information Systems. *Methods of Information in Medicine*, **34**, 147-157. <https://doi.org/10.1055/s-0038-1634569>
- [19] Malaka, R. and Zipf, A. (2000) Deep Map: Challenging IT Research in the Framework of a Tourist Information System. In: Fesenmaier, D.R., Klein, S. and Buhalis, D., Eds., *Information and Communication Technologies in Tourism 2000*, Springer, Vienna, 15-27. https://doi.org/10.1007/978-3-7091-6291-0_2
- [20] Srinivasan, A. and Stasko, J. (2017) Natural Language Interfaces for Data Analysis with Visualization: Considering What Has and Could Be Asked. *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers*, Barcelona, 12-16 June 2017, 55-59.
- [21] Yang, Q. (1997) Hierarchical Planning. In: Amarel, S. Biermann, A., et al, Eds., *Intelligent Planning, Artificial Intelligence*, Springer, Heidelberg, 163-188. https://doi.org/10.1007/978-3-642-60618-2_10
- [22] Manning, C.D., Raghavan P. and Schütze, H. (2008) Introduction to Information Retrieval. Cambridge University Press, Cambridge. <http://dx.doi.org/10.1017/CBO9780511809071>
- [23] Muslea, I. (1999) Extraction Patterns for Information Extraction Tasks: A Survey. In: The AAAI-99 Workshop on Machine Learning for Information Extraction.
- [24] Amaravadi, C.S. (2020) A Representation Scheme for Managing Complex Professional Knowledge. *Advanced Aspects of Engineering Research*, **1**, 13-37. <http://dx.doi.org/10.9734/bpi/aaer/v1/1506F>
- [25] Rosner, D., Grote, B., Hartman, K, Hofling, B. and Guericke, O. (1998) From Natural language Documents to Sharable Product Knowledge: A Knowledge Engineering Approach. In: Borghoff, U.M. and Pareschi, R., Eds., *Information Technology for Knowledge Management*, Springer, Berlin, 35-51. http://dx.doi.org/10.1007/978-3-662-03723-2_8
- [26] Renear, A., Dubin, D., McQueen, C.M. and Huitfeldt, C. (2002) Towards a Semantics for XML Markup. *Proceedings of the 2002 ACM symposium on Document Engineering*, McLean, 8-9 November 2002, 119-126. <http://dx.doi.org/10.1145/585058.585081>
- [27] Walsh, N. (1998) A Technical Introduction to XML. <https://www.xml.com/pub/a/98/10/guide0.html>.
- [28] MicroStrategy (2017) Basic Reporting Guide, version 10.9. <https://www2.microstrategy.com/producthelp/10.9/manuals/en/BasicReporting.pdf>
- [29] Marx, F., Mayer, J.H. and Winter, R. (2012) Six Principles for Redesigning Executive Information Systems—Findings of a Survey and Evaluation of a Prototype. *ACM Transactions on Management Information Systems*, **2**, 1-19. <http://dx.doi.org/10.1145/2070710.2070717>
- [30] Salmeron, J.L., Luna, P.L. and Martinez, F.J. (2001) Executive Information Systems in Major Companies: Spanish Case Study. *Computer Standards and Interfaces*, **23**, 195-207. [http://dx.doi.org/10.1016/S0920-5489\(01\)00071-X](http://dx.doi.org/10.1016/S0920-5489(01)00071-X)
- [31] Papageorgiou, E. and De Bruyn, H. (2011) Challenges of Executive Information Systems in Listed Johannesburg Stock Exchange Companies. *SA Journal of Information Management*, **13**, Article No. a448. <https://doi.org/10.4102/sajim.v13i1.448>

- <https://sajim.co.za/index.php/sajim>
- [32] Boynton, A. and Zmud, R. (1984) An Assessment of Critical Success Factors. *Sloan Management Review*, **25**, 17-27.
- [33] Pasi, G. (2010) Issues in Personalizing Information Retrieval. *IEEE Intelligent Informatics Bulletin*, **11**, 3-7.
- [34] Prusak, L. (1997) Introduction to Knowledge in Organizations. Butterworth-Heinemann, Boston, ix-xv.
<http://dx.doi.org/10.1016/B978-0-7506-9718-7.50003-2>
- [35] Kankanhalli, A., Tanudidjaja, F., Sutanto, J. and Tan, B.C.Y. (2003) The Role of Information Technology in Successful Knowledge Management Initiatives. *Communications of the ACM*, **46**, 69-73. <https://doi.org/10.1145/903893.903896>
- [36] Amaravadi, C.S. (2003) World and Business Computing in 2051. *The Journal of Strategic Information Systems*, **12**, 373-386.
<http://dx.doi.org/10.1016/j.jsis.2001.11.012>
- [37] Farhan, L., Kharel, R., Quiroz, M., Alissa, A. and Abdulsalam, M. (2018) A Concise Review on Internet of Things (IoT)-Problems, Challenges and Opportunities. *11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, Budapest, 18-20 July 2018, 1-6.
<https://doi.org/10.1109/CSNDSP.2018.8471762>
- [38] Stolpe, M. (2016) The Internet of Things: Opportunities and Challenges for Distributed Data Analysis. *ACM SIGKDD Explorations Newsletter*, **18**, 15-34.
<http://dx.doi.org/10.1145/2980765.2980768>
- [39] Aggarwal, C.C., Ashish, N. and Sheth, A. (2013) The Internet of Things: A Survey from the Data-Centric Perspective. In: Aggarwal, C., Ed., *Managing and Mining Sensor Data*, Springer, Boston, 383-428.
https://doi.org/10.1007/978-1-4614-6309-2_12
- [40] France, R.B., Wu, J., Petrie, M.L. and Bruel, J.M. (1996) A Tale of Two Case Studies: Using Integrated Methods to Support Rigorous Requirements Specification. *Proceedings of the Methods Integration Workshop*, Leeds, 25-26 March 1996, 1-18.
<https://doi.org/10.14236/ewic/MI1996.6>
- [41] Page-Jones, M. (1980) *The Practical Guide to Structured Systems Design*. Yourdon Press, New York.
- [42] Gulla, J.A. (2004) Understanding Requirements in Enterprise Systems Projects. *12th IEEE International Requirements Engineering Conference*, Kyoto, 10 September 2004, 176-185. <http://dx.doi.org/10.1109/ICRE.2004.1335675>
- [43] Escalona, M.J. and Koch, N. (2003) Requirements Engineering for Web Applications—A Comparative Study. *Journal of Web Engineering*, **2**, 193-212.
- [44] Glinz, M. (2000) Problems and Deficiencies of UML as a Requirements Specification Language. *10th International Workshop on Software Specification and Design*, San Diego, 7 November 2000, 11-22.
<http://dx.doi.org/10.1109/IWSSD.2000.891122>
- [45] Galliers, R.D. (1993) Research Issues in Information Systems. *Journal of Information Technology*, **8**, 92-98. <http://dx.doi.org/10.1057/jit.1993.13>