

Software Engineering A Forty Years Research Retrospective

E. Kajan

Abstract: This paper presents author experience with software development during last forty years. It is divided into three eras: closed and open -systems and Internet of Everything, each covered by challenges, issues and solutions. The main stream is given from the interoperability perspective and flavored by security issues. Variety of applications domains, that author faced during his carrier, each with own challenges, is presented. Example include computer graphics systems, special-purpose solutions, business-to-business, business process management, e-government, and smart cities.

Keywords: IoT, interoperability, security, SE paradigms, social networking

1 Introduction

Almost a half of century ago, Donald E. Knuth stated *"We have seen that computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty. A programmer who subconsciously views himself as an artist will enjoy what he does and will do it better"* [37].

Since than, author of this paper enjoyed in making that art. Meanwhile, a bunch of pioneer visions have expressed, followed with the birth of many ICT technologies facing software engineering with new opportunities and challenges. Some of these revolutionary shifts are given, for example, by J. C. R. Licklider who stated that computers will not serve as expensive apertures for complicated mathematical calculations only [41], or by Mark Waiser who predicted embedded world of invisible technologies [63]. In the former, Licklider defines man-machine symbiosis in which both members of the team may express their best talents, e.g. people are brilliant, slow and negligent, whilst machines are "stupid", but fast and rigorous. In the later, Waiser discuss 21st century computer and set the target *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it"*. Both have become reality.

Manuscript received May 15,2020 ; accepted November 12,2020.
E. Kajan is with the State University of Novi Pazar, Serbia

This paper gives an overview of authors' experience by facing software engineering opportunities and challenges during last four decades with emphasis on the latest period. The rest of paper is organized as follows. Section 2.1 is devoted to preliminary discussion of the software engineering evolution. It is followed by mutation of artifact processing, interoperability and security in section 2.2. Section 3.1 overviews some challenges during the era of closed systems, section 3.2 discusses interoperability in open systems era with emphasis on B2B, while section 3.3 is devoted to Internet-of-Everything (IoE) with emphasis on public administration, social enterprise, and smart cities. Results are summarized in section 4, whilst concluding remarks are given in section 5.

2 Software development evolution

2.1 Software engineering paradigms

Since early days of computing up to-day software has raised from simple isolated programs to weak-tied complex systems. Meanwhile, ICTs have entered home doors and enabled pervasive computing taking a part of everyday lives of more and more sophisticated users. Following push-pull principle between ICTs and applications requirements, software engineering went and going through evolutionary changes that are roughly summarized in Fig. 1.

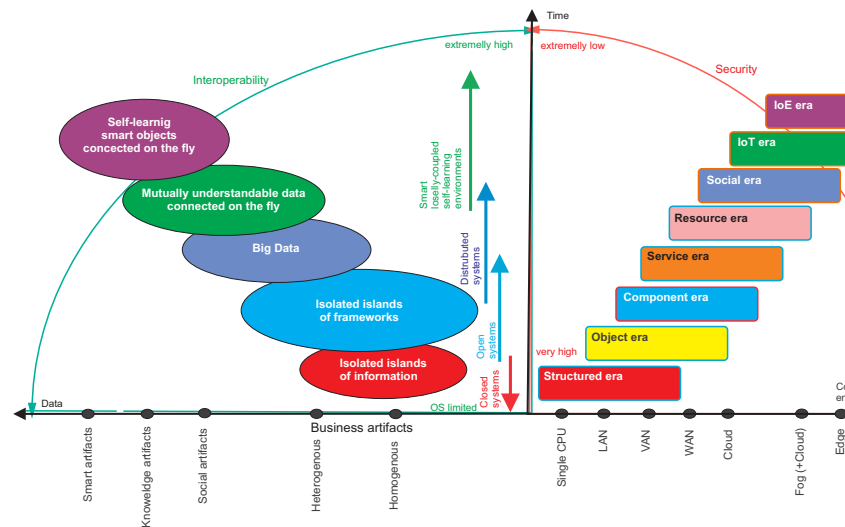


Fig. 1. Time line of software engineering

Structured era brought a clear distinction between functions and data. Concepts of information hiding [56] and abstract data types [42] are the major contributions to structured programming allowing large systems to be break into modules with clearly defined interfaces. This era is overlapping with the era of closed systems, when all programs and data have strongly limited to particular operating system. A step towards code reuse-ability

represents *object era* that promises improved productivity and software maintenance cost decrease. That era establishes object-oriented programming languages (e.g., Java and C++) and object-oriented design techniques (e.g., UML) [3]. Object era has been followed with Component-Based Software Engineering [39], a.k.a. *Component era* that improves software reuse and allows complex systems development by combining available modules in the manner of Rubik's cube to reach desired functionality [24]. Meanwhile, new technological improvements have taken place in the form of open systems and very soon enhancing openness with distributed components and resources.

Whilst the openness of objects and components has usually limited within institutional firewalls, natural evolution of component-based technologies is the *service era*, led by idea to make a programmable Web and emerged with the impressive raise of Web itself. Service era allows providers to expose their services on the Web and consumers to discover needed services, invoke and bind these to own applications. Popularity of Web services (WS), and appearance of cloud computing, where platform, infrastructure and software offer to consumers at large scale introducing new paradigm: Everything-as-a-Service (*aaS) [13].

However, WS stack is complex, thus some providers decided to expose their services as simple lightweight APIs using standard HTTP methods (e.g. GET, POST) in order to execute operations over resources. A resource is any entity on the Web accessible through URI. This new era is known as *resource era*. On the wave of Representational State Transfer (REST) [15], Baker et al. discuss Everything-as-as-Resource (*aaR) as a paradigm for designing collaborative systems on the Web.

"Social fever" started in 2005 by many platforms that allowed people-to-people communications over the Web, e.g. Facebook, Twitter, LinkedIn, etc. This made new era in software engineering, *social era*. The social wave takes the great influence how today's ecosystems are designed and also brought new information phenomenon known as Big Data [7]. Unlike all previous experiences with feeding computers with data, Big Data is characterized by so-called four Vs: Volume, Variety, Velocity, and Value. In [5], we argue that software engineering should be revisited to capture all features that social era brings.

In 2005, ITU-T released the vision underlying the Internet of things (IoT) and states "*The next logical step in this technological revolution (connecting people anytime, anywhere) is to connect inanimate objects a communication network.*". Since then we are witnessing enormous flooding of things into our everyday lives and entering the door of fourth industrial revolution (a.k.a. Industry 4.0), etc. According to Gartner, connected things will reach 20.8 billion by 2020. Thus we enter *IoT era* in software engineering. In 2020, IoT has been precisely defined by ITU "as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies". Despite high potential of IoT, there is a lack of software engineering techniques that may deal with that emerging world [65]. But, that is not the end. Recently, new visionary statements are become reality. E.g. Snider and Byrd announce Internet of Everything (IoE) [60] as the next evolutionary stage of IoT, and put new challenges to software engineering in *IoE era*. These are discussed later in section 3.3

2.2 Artifacts mutation and processing

Evolution of data, that we deal with, is shown on the left horizontal axis in Fig. 1, whilst the right axis emphasizes the evolution of computational environments upon these data.

Nigam and Caswell define *business artifacts (BAs)* as concrete, identifiable, self-describing chunk of information that can be used by a business person to actually run a business [54]. Unlike era of closed systems when BAs were homogeneous, today they are heterogeneous by default. *Social artifacts (SA)* might be seen as any form of online activity (e.g., tweet, tag, share, comment, and endorse) that occurs over an open Web 2.0 application, but also to any form of online activity over ubiquitous Web (e.g., data emitted and actions triggered by captured sensor's signals, etc.). *Knowledge artifacts (KA)* may be seen as "objects that convey and hold usable representation of knowledge" as given in [18]. Recently, the world of *smart artifacts (objects) (SO)* raises on daily basis consisting numerous "autonomous physical/digital objects augmented with sensing, processing, and network capabilities" [38].

Computational environments have also revolutionary changed. Sixty years ago, or nearby, most of data processing were performed over a single CPU embedded into robust mainframes. This era has known as *batch processing era*, when programs and data were put into processing in the form of so-called jobs coded on punched cards or tapes. Networking, first over LAN, brought *time sharing era* allowing computer supported cooperative work and better interaction with computers, as well. However, before Internet era such cooperative work has been limited to closed environments inside organizational firewalls. And then we got a new industrial revolution. Along the fallen of Berlin wall, when Internet starting its spreading over the globe, the invention of Web by Tim Berners-Lee, and the Marc Anderson's MOSAIC browser we have jumped in the *Internet era*. This jump has changed a lot and triggered new software engineering paradigms and made now computational environments, like clouds, fogs and edges. Today, we could say that processing is taking place anywhere, any time, anyway, on the move, etc.

All the time two important ICT issues, interoperability and security, stay actual, theoretically and practically, as well. *Interoperability* is the ability of two or more systems or components to exchange information and use the information that has been exchanged in a useful way. Interoperability can come in three types at least: communication, syntactic, and semantic [25]. Communication interoperability is the capacity to bring the data from one place to another on time with the required quality. This depends on the existence of a solid infrastructure and standardized protocols. Syntactic interoperability is the capacity to provide data in a format that peers understand. It should allow content exchange among multiple software components regardless of their implementation languages, run-time environments, and other technological differences. Semantic interoperability is the capacity to assign meaning to data so that semantic conflicts are overcome due to differences in implicit meanings, perspectives, and assumptions in the data, business processes and so on. *Security* may be defined by possibility degree of interception, stilling and changing information. It is a complex umbrella term that covers, among others, policies, tools, guidelines, actions, best

practices, and technologies, used to set up security on the desired level. These are discussed further through this paper in the context of presented topics.

3 Swimming through eras

This section covers authors' experience with software development since closed systems up-to-day. At the end it also includes a short overview of ongoing research.

3.1 Closed systems

My experience with software development in closed systems environment relates to Ei-Honeywell DPS6 mainframes based on General Comprehensive Operating System 6 (GCOS6). During 1980s we faced with customer' special requirements that may be classified as: (1) a customer has legacy peripheral equipment from another vendor that would like to keep under GCOS6, (2) a customer wish a new device that should be developed, produced and enabled to run on GCOS6, (3) customers need equipment that is available on the market, but, by default, not works on DPS6 and ultimately ask to buy all from us to protect their investments in advance, and (4) we would like to penetrate into market with unique advanced software.

These market driven circumstances trigger establishment of Special projects department that we run since 1984 up to economic disaster of our society in 1991. The four aforementioned special needs in terms of software engineering required, but not limited to new GCO6 commands, device drivers and protocols under GCOS6, and complete integrated solutions (systems) design and development. Examples include, but not limited to, punched paper tape fast reader case [20], variety of OEM peripherals that we connect to GCO6 mainframes, such as Tektronix graphics terminals and XEROX laser printers, AS-PIPLOT graphical kernel system developed to satisfy customer needs for graphics [21], etc.

Era of closed systems is characterized by very high security. However, some potential vulnerabilities were possible by abandoning Orange Book level D of security, or by organizational weakness allowing leaking of data via transient electromagnetic pulse emanation. While the former was caused by human negligence, the later was usually limited to government and military information systems whose parts were exposed outside Faraday's cages. On the other hand, interoperability during closed systems era was limited to organizational boundaries and resulted by isolated islands of information.

3.2 Era of open systems

Kajan defined *open systems* as a set of open components each satisfying portability, scalability and interoperability and covered by international standards [22]. In the late 1980s we got open systems, first by ISO OSI standard (Open Systems Interconnection), whose concepts is presented in [23] through the term itself and cross references, as well. Soon Internet become a main focus due to its fast growing, Web and GUI-enabled browsers. Since

than, I have shifted my research interest to an emerging area: electronic commerce(EC). The crash of so-called dot com era opened new research challenges, especially in terms of interoperability and security of EC applications.

It required many heterogeneous applications to be integrated within and outside an enterprise into a single coherent environment [36]. The integration within an enterprise requires all applications, for instance, Enterprise Resource Planning (ERP), Supply Chain Management (SCM), Customer Relationship Management (CRM), etc., to communicate with no data latency. This approach is known as ZLE (Zero Latency Enterprise), the term first coined by Compaq and Gartner, and defines a business-technical strategy applied to data exchange across technical and organizational boundaries within an enterprise in order to make an almost perfect match between strategic goals and information available from technical resources (data bases, data warehouses, etc.). It is being done over proprietary value added networks (VANs) or over the Internet.

Special emphasis has given to Business-to-Business (B2B) interoperability. In a business scenario, business processes (here and after BPs), are usually loosely coupled; that means BPs require ad hoc integration from time-to-time. Such on-the-fly relationship may be faced with many conflicts, due to large heterogeneity [12]. The problem is multifaceted. A variety of solutions have been proposed based on different technological enablers and approaches [27] and [30]. The most common complexity of interoperability issues includes, but not limited to, standards, semantics, ontologies, and services. The potential of semantic technologies in B2B has given by Tim Berners-Lee by his vision: "*Now we can imagine the world of people with active machines forming part of the infrastructure. We have only to express a request for bids, or make a bid, and machines will turn a small profit matching the two*" [16].

On the line on aforementioned vision we have developed the Business-to-Business Ontology-Oriented Middleware (B2BOOM) architectural B2B framework based on semantic mediators and ontologies [36]. This solution enables *business entities (be)* to participate in an electronic marketplace with assumption that commitments by the clients about the data to be shared are exist. The core architectural B2BOOM components are brokers and ontologies. In essence brokers are mediators [64] that allows two *bes* to exchange their business documents and transaction on demand automatically. Each *be* contains a B2B application, a local ontology, and a corresponding database. For each data source there is a translator (or wrapper) that logically converts basic data objects to the common information model. B2B server holds meta-knowledge about marketplace and a global top-level ontology. Unified method for data access allows simply chaining the translators and semantic mediators, as well. From the viewpoint of semantic interoperability across virtual enterprises (VE), in [34] we propose the use of goal based models for VEs to avoid ambiguities during enactment.

Open systems era in B2B interoperability resulted by variety of B2B frameworks, each relied on a particular technology stack [52], and in essence limited open B2B to these frameworks. The security requirements for B2B transactions are very rigorous. There is a need to protect business data before, after and during the transactions on the transmission

line. In [55], seven key parameters which influence EC security has identified. These include attack analysis, access control, authentication, data confidentiality and integrity, non repudiation, and system reliability and availability.

A question that arises here is *interoperability vs. security*. Medjahed et al. [52] identify several dimensions of B2B interoperability, autonomy, manageability, coupling, heterogeneity and adaptability. In [26], Kajan gives an analysis of potential vulnerabilities in B2B transactions due to these features. The only dimension that is good for security is the autonomy. Autonomy refers to the degree of compliance of a *be* to the global business rules. If the degree of autonomy is very high, the potential risks are low because *bes* communicate via well-defined interfaces allowing local control over operations. On the other side, external manageability seems to be an extremely dangerous feature of B2B interoperability because it facilitates the supervision and control by another *be*. Coupling among business partners may be tight or loose. In tightly coupled systems business entities strongly depends on each other and usually communicate via well-defined interfaces using business scenario known in advance. But, there is more likely that loose coupling will take place and will appear as a potential hole for attackers. Heterogeneity is the key feature of B2B environments and by default causes problems in security. And finally, adaptability refers to the degree of quickly adapt to changes, which is, in general, good feature.

3.3 Distributed systems and IoE era

This part is illustrated by ten years research on the public administration in digital age , social enterprise, smart cities, and IoT, as well.

The goal of the first aforementioned project was to analyze people observations and opinions given in different online forms to facilitate decision making by public authorities in three applications domains: social, biological and economic. The architectural concept of the project is shown in [59]. The process starts with data mining into public repositories and/or social networks. In order to analyze fetched data we have developed a stop words dictionary for the Serbian language and normalize available [35]. We have made several experiments and found that n-gram analysis makes it faster [51]. Normalized inputs are processed by rule based reasoning that separates business logic from heterogeneous data and the similarity measuring from decision making, as well. This approach we use to measure local economic environment, i.e. the ability of state and local government's bodies' to deal with specific problems in creating stable and favorable business environment. at the city of Niš [29]. Questions are given in a form of crowdsourcing, whilst answers are obtained from entrepreneurs.

We also focus on sentiment analysis [50] tailored for Serbian language. This ongoing field of research "aims at determining opinions, emotions, and attitudes reported in source materials like documents, short texts, sentences from reviews, blogs, and news, among other sources" [9]. *Sentiment analysis* can be applied on a variety of textual sources on different granularity levels (an entire document, phrases, separate words) [17] and may include several steps, like tokenization, POS tagging and lemmatization, that are usual in any *NLP*

task, but also may include word disambiguation [10]. Problems that may occur in sentiment analysis are the ambiguity of words, non-standard writing, slang, neologisms, segmentation problems, irony, metaphor, and negation, just to mention a few.

Social enterprise is a paradigm shift that represents a fundamental change in how today's enterprises should operate. That paradigm shift replaces Fordism (Organizational philosophy introduced in Ford automobile company inspired with scientific management theory [40].) by *egalitarianism* that significantly changes rights and roles of business workers. Instead of dividing them to those who direct what to do and how and to those who do it, all of them are getting equal and with equal rights. Unlike traditional top-down approach to responsibilities, i.e. knowing in advance who is in charge for what, idea of Wisdom of crowds [19] promotes loose pancake model of hierarchy where problem solving, decision making and supervision may be assigned to unknown crowd targeting human resources from everywhere in cyberspace.

Our research agenda has focused around three directions: social design, coordination, and monitoring of BPs, each of which is described by details in [14]. The first direction consists of integrating smoothly Web 2.0 applications in BP design and execution in order to make social enterprise more practical. The second direction attempts to capitalize on these applications during BP completion and conflict resolution. Last but not least the third direction studies different opportunities to capture and analyze different flows in BPs, as well as their interactions in order to evaluate BP improvement. Questions include who are the social worlds' stakeholders, what actions do Web 2.0 applications allow stakeholders to perform, what is the impact of executing BPs on these actions and *vice-versa*, and what interactions take place between the business and social worlds?

Different relations with a "social flavor" have been proposed such as interchange, substitution, delegation, and peering. They permitted to connect the constituents of a BP that are task, person, and machine together. For instance, tasks can be substituted when their requirements are not met. Persons delegate their tasks when unexpected changes occur in their schedules. Last but not least machines act as backups when their peers fail. We have reported results of this part of the project in different venues. In [31], we present approach and introduce an extension of BPMN standard implemented in open source Yaoqiang BPMN editor. Complete framework overview has given in [44] and [46]. We have also shown a variety of use cases, like handling software engineering [33], hospital treatment [44], running purchase order [31], etc.

Interoperability in Enterprise 2.0 requires interactivity between BAs i SAs that is proposed by the concept of Social Machines (SMs) [4]. It allows establishment of the *Connection-Open-Reachable-Engagement (CORE)* characteristics that allow to "control" the social fever advocate for converting ad hoc relations into long-lasting ones, creating new conversation channels with stakeholders, facilitating ubiquitous accessibility to the organization, and creating a culture of community.

Many definitions of *smart cities* exist. From a system's point of view, Mitchel [53] defines smart city as the combination of the nerves, brain, sensory organs, and knowledge, i.e., the combination of digital telecommunications networks, ubiquitously embedded in-

telligence, sensors, and software. From a usability's point of view, it may be seen as “*a developed urban area that creates sustainable economic development and high quality of life by excelling in multiple key areas: economy, mobility, environment, people, living, and e-government*” [6]. Last but not least, Chourabi et al. [8] describe smart cities from a conceptual's point of view as large organic systems connecting many subsystems and components.

Regardless different views, a smart city is a place where IoE allows such a complex ecosystem to accomplish its goals by listening its sensors, connect different inputs, and adjust own operations accordingly. The main sensors in a smart city are citizens that act as prosumers, i.e. IoE entities that produce data for smart city applications, but also consume information smart city initiatives [57].

In that sense we develop a framework that allows the transparent co-existence of citizens and IoT-compliant things in smart cities [61]. The framework is based on social machines which are responsible for data collection from city sensors (citizen and IoT) and high-level artifacts used to measure citizens engagement and willingness in smart city initiatives. Social machines act as a meet-in-the middle environment between subscribers (citizens and things) and smart city authorities (actually, applications they used and data on which these applications act upon) [4]. It facilitates the buy-in of both citizens and things in these initiatives. The framework is supported by temporary tunnel closure due to car accident use case.

Another use case that support citizen and things cooperation is presented in [62]. In this approach we use smart city sensors for BP re-engineering to avoid traffic jams, find free parking, or to find the best route four tourists that want visit some interesting places without losing their time in crowded facilities, etc. We discuss a BP re-engineering, also, by a SocialMiner approach [32]. In this paper we analyze how actions over social media (in this case Facebook). like comment, share, like, endorse, etc., may be used to track and measure feedback of the followers to some smart city initiative.

High quality of life is an imperative smart city goal. In this important research we also use social machine to bring together inputs of patients and things to achieve that goal [2]. In essence, IOT sensors measure air quality by capturing values of the main pollutants (O_3 , PM_{10} , SO_2 , NO_2 , NO_3 , N_2O_4 , N_2O_5 , and CO), while people participates as sensors, either by reporting some problems in breathing or asking for a medical help. The whole platform consists with three layers: sensors, social machine, and smart city cloud services.

We have also present few other original approaches for applying things into BP execution, by storytelling and by interaction between so-called Thing-as-a-Process (TaP) and process-of-Things (PoT). In the former, every thing (non-living, e.g. temperature sensor in the milk tank and living-thing, e.g. pasteurizer, in a milk-supply chain) has a role described by appropriate script, like for a role in a movie being filmed [48]. The latter approach overcomes IoT-related (in particular, silo restriction, no-reasoning and absence of proper modeling techniques) and BP-related limitations (in particular, physical surrounding, data input, and context insensitivity.) when they meet each other [49].

Interoperability in the IoE era is taking place by real-time collaboration between hu-

mans, software, things, and many other IoE entities that cooperate in different domains and environments [28]. A big challenge here is to ensure visibility, common understanding and interoperability on all collaboration levels. Hidden obstacles are already identified in the lack of software engineering techniques to deal with an unpredictable behavior of social world [5], and it could be raised with the expansion of that world by socialization of other IoE entities, like Web Services [43] and IoT [1], etc. A step forward to IoT interoperability is presented in [47] by extending OWL-S into OWL-T. That approach supports semantic description of things from three dimensions: interaction, consumption and operation. In essence, OWL-T allows to describe interaction peers, which resource things consume and how they may be invoked.

Because of high heterogeneity, connectivity and emerging socialization of everything, the restricted ability to control such complex interactions may become critical and new risks may arise. It is especially valid for IoT where bunches of mutually connected heterogeneous devices exist as well applications that relay on them [58]. Most of current research on security is devoted to vetting either mobile applications or IoT devices. However, it usually performs after their deployment, that therefore leads to the fact that we potentially use vulnerable devices and applications. In our approach we have clear statement that things should be vet before the first use [45]. We focus on thing duties: sensing, actuating and communicating and identify security holes that should be check before putting things into operation.

4 Retrospective summary

This section gives a summary based on published papers over all eras, their citations, and some overview of international cooperation.

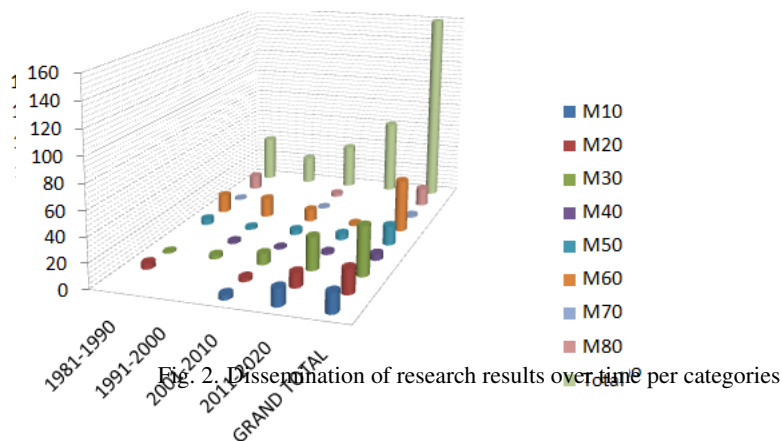


Figure 2 overviews author research results over time divided by decades and organized

by M categories. During the last ten years author has established decent international cooperation with a number of researches. Figure 3 shows a part of his DBLP records since 2012 (<https://dblp.org/pers/hd/k/Kajan:Ejub>) with emphasis on M20 results achieved as a result of that cooperation.

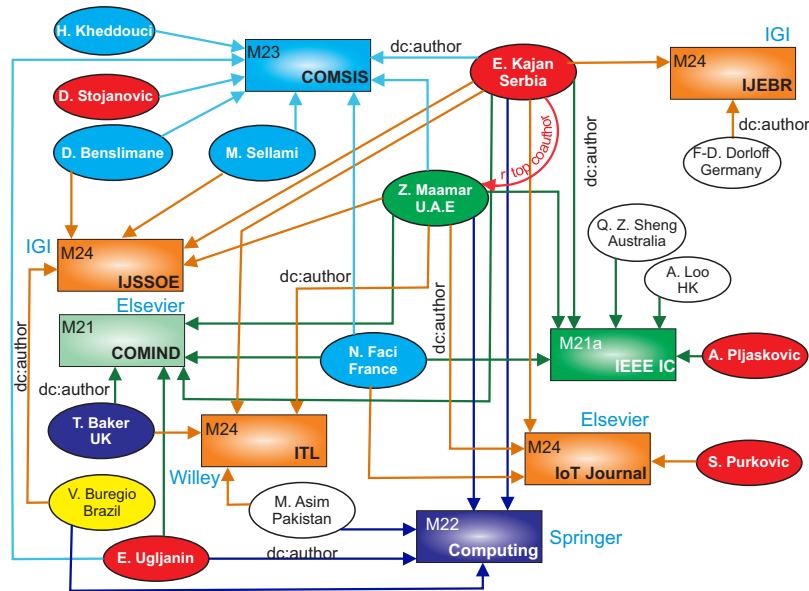


Fig. 3. International cooperation by DBLP shown in RDFs

International cooperation has also raised by editing three reference books, two of which are devoted to electronic business interoperability [27] and [30]. In the former, the seventy-one authors from fifteen countries and forty research organizations contributes to. In the later, the thirty-eight chapters are written by eighty authors who also define more than 250 key terms in the area of e-business standards. The third book is devoted to e-government research [11]. This effort was awarded by nineteen chapters contributed by fifty-three authors from all around of world that, in summary, present a solid foundation for understanding e-government challenges, adoption experiences, importance, and benefits.

Figure 4 gives an overview of citations per some well-known index bases. Most of them has been established after 2000, thus, data are based on period 2000-2020 by the time this paper has written.

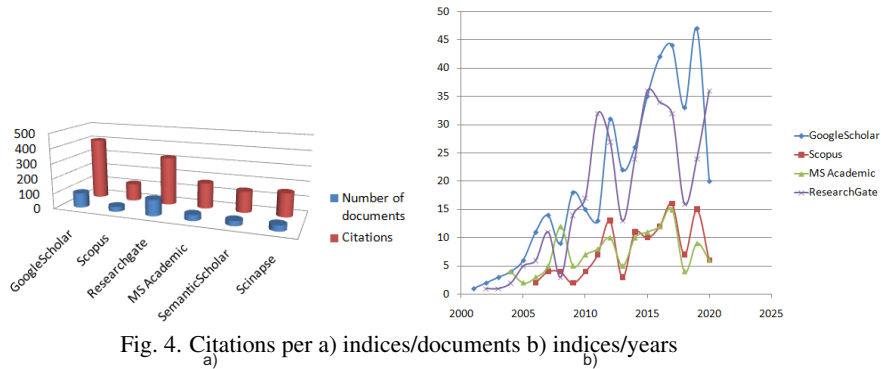


Fig. 4. Citations per a) indices/documents b) indices/years

5 Conclusion

If we take a look into the past of software engineering, and more generally, in computer science, we can conclude that, in essence, it is a subject of methodology that was used to overcome obstacles that mined either technologies or applications to be mutually fitted at given time. That gap, also known as push-pull principle, drive software engineering all the time targeting perfect alignment between applications and technologies upon which these are built and run.

Today we may resume that a bunch of applications have been matured meanwhile and doing their jobs successfully. But, on the other hand, a bunch of applications are in an infant stage and sooner or later will push new technological development. Furthermore, there will be new ideas, like upside-down business models, that could be born either in minds of people that have not been born yet, or in a distributed artificial mind.

If we take look into the future we can imagine a new world of mutually, on-the-fly, connected smart objects that will learn from each other, do something together and go further. That vision will require new software engineering methodologies that will be able to define new technologies, and trigger new unseen applications. These will be run by people or by artificial intelligence, who knows?

Finally, it is my pleasure that I have had the opportunity to work in the exciting environment of software engineering and from time-to-time made an art or contributed to it a little.

Acknowledgment

This paper is devoted for author official retirement after 40 years of work in computer industry and higher-education. Author would like to thanks to all researchers who worked with him during this period, 79 in total, from 20 countries by the time when this paper has been written.

References

- [1] ATZORI, L., IERA, A., MORABITO, G., AND NITTI, M. The Social Internet of Things (SIoT) - When Social Networks Meet the Internet of Things: Concept, Architecture and Network Characterization. *Computer Networks* 56, 16 (2012).
- [2] AVDIĆ, A., KAJAN, E., JANKOVIĆ, D., AND AVDIĆ, D. Towards context-aware smart healthcare platform. *International Journal of Electrical Engineering and Computing* 3, 1 (2019), 26–31.
- [3] BOOCH, G. Object-oriented development. *IEEE Transactions on Software Engineering SE-12*, 2 (1986), 211–221.
- [4] BURÉGIO, V., MAAMAR, Z., AND ROMERO DE LEMOS MEIRA, S. An Architecture and Guiding Framework for the Social Enterprise. *IEEE Internet Computing* 19, 1 (2015), 64–68.
- [5] BURÉGIO, V. A., KAJAN, E., SELLAMI, M., FACI, N., MAAMAR, Z., AND BENSLIMANE, D. Revisiting software engineering in the social era. *IJSSOE* 6, 4 (2016), 36–46.
- [6] CARLOS, E., FRANCISCO, F., AGUSTI, S., HECTOR, P., SALEEM, Z., AND FEDERICO, G. Smart Government: Opportunities and Challenges in Smart Cities Development. In *Democratic Strategies and Citizen-Centered E-Government Services*, Č. Dolićanin, E. Kajan, D. Randjelović, and B. Stojanović, Eds. IGI Global, 2014, pp. 1–19.
- [7] CHEN, M., MAO, S., AND LIU, Y. Big data: A survey. *Mobile networks and applications* 19, 2 (2014), 171–209.
- [8] CHOURABI, H., NAM, T., WALKER, S., GIL-GARCIA, J., MELLOULI, S., NAHON, K., PARDO, T., AND SCHOLL, H. Understanding Smart Cities: An Integrative Framework. In *Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS'2012)* (Jan 2012), pp. 2289–2297.
- [9] DA SILVA, N. F., HRUSCHKA, E. R., AND HRUSCHKA, E. R. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems* 66 (2014), 170 – 179.
- [10] DIAMANTINI, C., MIRCOLI, A., POTENA, D., AND STORTI, E. Social information discovery enhanced by sentiment analysis techniques. *Future Generation Computer Systems* (2018).
- [11] DOLIĆANIN, Č., KAJAN, E., RANDJELOVIC, D., AND STOJANOVIC, B. *Handbook of research on democratic strategies and citizen-centered E-government services*. Information Science Reference, 2015.
- [12] DORLOFF, F., AND KAJAN, E. Balancing of heterogeneity and interoperability in e-business networks: The role of standards and protocols. *IJEBR* 8, 4 (2012), 15–33.
- [13] DUAN, Y., FU, G., ZHOU, N., SUN, X., NARENDRA, N. C., AND HU, B. Everything as a service (xaas) on the cloud: origins, current and future trends. In *2015 IEEE 8th International Conference on Cloud Computing* (2015), IEEE, pp. 621–628.
- [14] FACI, N., MAAMAR, Z., KAJAN, E., AND BENSLIMANE, D. Research roadmap for the enterprise 2.0: Issues & solutions. *Scientific Publications of the State University of Novi Pazar Series A: Applied Mathematics, Informatics and mechanics* 6, 2 (2014), 81–89.
- [15] FENG, X., SHEN, J., AND FAN, Y. Rest: An alternative to rpc for web services architecture. In *2009 First International Conference on Future Information Networks* (2009), IEEE, pp. 7–10.

- [16] FENSEL, D. *Spinning the Semantic Web: bringing the World Wide Web to its full potential*. MIT press, 2005.
- [17] GIATSOGLOU, M., VOZALIS, M. G., DIAMANTARAS, K., VAKALI, A., SARIGIANNIDIS, G., AND CHATZISAVVAS, K. C. Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications* 69 (2017), 214 – 224.
- [18] HOLSAPPLE, C., AND JOSHI, K. Organizational knowledge resources. *Decision Support Systems* 31, 1 (2001), 39 – 54. Knowledge Management Support of Decision Making.
- [19] HOWE, J. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*, 1 ed. Crown Publishing Group, New York, NY, USA, 2008.
- [20] KAJAN, E. Some aspects of extending operating system while connecting new io devices. In *XXIX ETAN conference* (1985), ETAN, pp. VIII.139–VIII.146.
- [21] KAJAN, E. Proširenje gks sistema sa rasterop funkcijama. In *XXXVII Jugoslovenska konferencija ETANa* (1988), ETAN, pp. VIII.385–VIII.392.
- [22] KAJAN, E. *Otvoreni sistemi: koncepti, komponente i aplikacije za budućnost*. Prosveta, 1994.
- [23] KAJAN, E. *Information technology encyclopedia and acronyms*. Springer, 2002.
- [24] KAJAN, E. The b2b standard it framework. In *Proceedings of the IADIS International Conference e-society 2003* (June 2003), pp. 597–602.
- [25] KAJAN, E. The maturity of open systems for B2B. *SIGecom Exchanges* 5, 2 (2004), 34–44.
- [26] KAJAN, E. The security issues of b2b interactions. *Informatika* 6, 1 (2006), 105–112.
- [27] KAJAN, E. *Electronic Business Interoperability: Concepts, Opportunities and Challenges: Concepts, Opportunities and Challenges*. IGI Global, 2011.
- [28] KAJAN, E. Revisiting interoperability issues and challenges in the era of ubiquitous social web of everything. In *Proceedings of the International Conference on Computing for Engineering and Sciences* (2017), pp. 1–5.
- [29] KAJAN, E., AVDIĆ, A., MAROVAC, U., LJAJIĆ, A., ŠIMIĆ, G., AND STANKOVIĆ, J. Enhancing local economic development using collective intelligence. In *2015 23rd Telecommunications Forum Telfor (TELFOR)* (2015), IEEE, pp. 882–885.
- [30] KAJAN, E., DORLOFF, F.-D., AND BEDINI, I. *Handbook of Research on E-Business Standards and Protocols: Documents, Data and Advanced*. 2012.
- [31] KAJAN, E., FACI, N., MAAMAR, Z., LOO, A., PLJASKOVIC, A., AND SHENG, Q. Z. The Network-based Business Process. *IEEE Internet Computing* 18, 2 (March/April 2014).
- [32] KAJAN, E., FACI, N., MAAMAR, Z., SELLAMI, M., UGLJANIN, E., KHEDDOUCI, H., STOJANOVIĆ, D., AND BENSLIMANE, D. Real-time tracking and mining of social media. *Computer Science and Information Systems* 17, 1 (2020), 1–24.
- [33] KAJAN, E., MAAMAR, Z., AND FACI, N. Software Process Improvement by Enterprise 2.0 Technologies. In *Proceedings of the 2014 Contemporary Problems of Mathematics, Mechanical and Informatics- Selected Papers (CPMMI 2014)* (Novi Pazar, Serbia, 2014), pp. 56–64.
- [34] KAJAN, E., NARENDRA, N. C., AND MAAMAR, Z. Towards conflict-free virtual enterprises. In *Encyclopedia of E-Commerce Development, Implementation, and Management*. IGI Global, 2016, pp. 1116–1129.

- [35] KAJAN, E., PLJASKOVIĆ, A., AND CRNIŠANIN, A. *Normalizacija tekstualnih dokumenata na sprskom jeziku u cilju efikasnijeg pretraživanja u sistemima e-uprave*. Etran, Zlatibor, jun (2012).
- [36] KAJAN, E., AND STOIMENOV, L. Toward an ontology-driven architectural framework for B2B. *Commun. ACM* 48, 12 (2005), 60–66.
- [37] KNUTH, D. E. Computer programming as an art. *Communications of the ACM* 17, 12 (1974), 667–673.
- [38] KORTUEM, G., KAWSAR, F., SUNDRAMOORTHY, V., AND FITTON, D. Smart objects as building blocks for the internet of things. *IEEE Internet Computing* 14, 1 (2010), 44–51.
- [39] KOZACZYNSKI, W., AND BOOCH, G. Component-based software engineering. *IEEE software* 15, 5 (1998), 34.
- [40] KURZ, M., AND FLEISCHMANN, A. BPM 2.0: Business process management meets empowerment. In *Subject-Oriented Business Process Management - Second International Conference, S-BPM ONE 2010, Karlsruhe, Germany, October 14, 2010. Selected Papers* (2010), pp. 54–83.
- [41] LICKLIDER, J. Man-Computer Symbiosis. *IRE Transactions on Human Factors in Electronics HFE-1*, 1 (March 1960), 4–11.
- [42] LISKOV, B., AND ZILLES, S. Programming with abstract data types. *ACM Sigplan Notices* 9, 4 (1974), 50–59.
- [43] MAAMAR, Z., BADR, Y., FACI, N., AND SHENG, Q. Z. Realizing a social ecosystem of web services. In *Advanced Web Services*, A. Bouguettaya, Q. Z. Sheng, and F. Daniel, Eds. Springer, 2014, pp. 455–473.
- [44] MAAMAR, Z., FACI, N., KAJAN, E., SAKR, S., BOUKHEBOUZE, M., AND BARNAWI, A. How to Make Business Processes "Socialize"? *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* 2, 5 (2015).
- [45] MAAMAR, Z., KAJAN, E., ASIM, M., AND SHAMSA, T. B. Open challenges in vetting the internet-of-things. *Internet Technol. Lett.* 2, 5 (2019).
- [46] MAAMAR, Z., SAKR, S., FACI, N., BOUKHEBOUZE, N., AND BARNAWI, A. SUPER: Social-based Business Process Management Framework. In *Proceedings of the 12th International Conference on Service Oriented Computing (ICSOC'2014)* (Paris, France, 2014).
- [47] MAAMAR, Z., FACI, N., KAJAN, E., ASIM, M., AND QAMAR, A. OWL-T for a semantic description of iot. In *New Trends in Databases and Information Systems - ADBIS 2020 Short Papers, Lyon, France, August 25-27, 2020, Proceedings* (2020), J. Darmont, B. Novikov, and R. Wrembel, Eds., vol. 1259 of *Communications in Computer and Information Science*, Springer, pp. 108–117.
- [48] MAAMAR, Z., FACI, N., KAJAN, E., PURKOVIC, S., AND UGLJANIN, E. Process-of-things: Weaving film industry's practices into the internet-of-things. *Internet Things* 11 (2020), 100248.
- [49] MAAMAR, Z., KAJAN, E., GUIDARA, I., MOCTAR-M'BABA, L., AND SELLAMI, M. Bridging the gap between business processes and iot. In *IDEAS 2020: 24th International Database Engineering & Applications Symposium, Seoul, Republic of Korea, August 12-14, 2020* (2020), B. C. Desai and W. Cho, Eds., ACM, pp. 2:1–2:10.

- [50] MAROVAC, U., A.LJAJIC, KAJAN, E., AND A.AVDIC. Towards the lexical resources for sentiment-reach informal texts - the serbian language case. In *V CPMMI conference* (Novi Pazar, Serbia, 2018), pp. 1–10.
- [51] MAROVAC, U., PLJASKOVIC, A., CRNISANIN, A., AND KAJAN, E. N-gram analysis of text documents in serbian language. In *In Proceedings of the 20th Telecommunications Forum (TELFOR)* (Belgrade, Serbia, 2012), pp. 1385–1388.
- [52] MEDJAHED, B., BENATALLAH, B., BOUGUETTAYA, A., NGU, A. H., AND ELMAGARMID, A. K. Business-to-business interactions: issues and enabling technologies. *The VLDB journal* 12, 1 (2003), 59–85.
- [53] MITCHELL, W. Intelligent Cities. <http://www.uoc.edu/uocpapers/5/dt/eng/mitchell.pdf>, 2007.
- [54] NIGAM, A., AND CASWELL, N. S. Business artifacts: An approach to operational specification. *IBM Syst. J.* 42, 3 (July 2003), 428–445.
- [55] NUGENT, J. H., AND RAISINGHANI, M. S. The information technology and telecommunications security imperative: Important issues and drivers. *Journal of Electronic Commerce Research* 3, 1 (2002), 1–14.
- [56] PARNAS, D. L., SHORE, J. E., AND WEISS, D. Abstract types defined as classes of variables. *ACM SIGPLAN Notices* 11, SI (1976), 149–154.
- [57] PEDRINACI, C., AND DOMINGUE, J. Toward the Next Wave of Services: Linked Services for the Web Data. *Journal of Universal Computer Science* 16, 13 (2010), 1694–1719.
- [58] QUIROLGICO, S., VOAS, J., AND KUHN, R. Vetting mobile apps. *IT Professional* 13, 4 (2011), 9–11.
- [59] ŠIMIĆ, G., JEREMIĆ, Z., KAJAN, E., RANDJELOVIĆ, D., AND PRESNALL, A. A framework for delivering e-government support. *Acta Polytechnica Hungarica* 11, 1 (2014), 79–96.
- [60] SNYDER, T., AND BYRD, G. The internet of everything. *Computer*, 6 (2017), 8–9.
- [61] UGLJANIN, E., KAJAN, E., MAAMAR, Z., ASIM, M., AND BURÉGIO, V. Immersing citizens and things into smart cities: a social machine-based and data artifact-driven approach. *Computing* (2020), 1–20.
- [62] UGLJANIN, E., STOJANOVIĆ, D., KAJAN, E., AND MAAMAR, Z. Re-engineering of smart city’s business processes based on social networks and internet of things. *Facta Universitatis, Series: Automatic Control and Robotics* 16, 3 (2018), 275–286.
- [63] WEISER, M. The Computer for the 21st Century. *ACM SIGMOBILE Mobile Computing and Communications Review* 3, 3 (July 1999).
- [64] WIEDERHOLD, G. Mediators in the architecture of future information systems. *Computer* 25, 3 (1992), 38–49.
- [65] ZAMBONELLI, F. Key Abstractions for IoT-Oriented Software Engineering. *IEEE Software* 34, 1 (January-February 2017).