Empir Software Eng
DOI 10.1007/s10664-013-9280-x

An experimental investigation comparing individual and collaborative work productivity when using desktop and cloud modeling tools

Gregor Polančič · Gregor Jošt · Marjan Heričko

Abstract Successful modeling tools need to effectively support individual as well as team-based work (collaboration) within colocated and virtual environments. In the past, achieving this has been challenging, since traditional modeling tools are desktop-based and thus suitable for individual and colocated work only. However, with the rise of web-based architectures and the cloud paradigm, desktop modeling tools now have rivals in their web-based counterparts that are especially suited for online collaboration (e-collaboration). The objective of our research was to probe the question as to 'which type of modeling tools (desktop or cloud-based) performs better in cases of individual work and e-collaboration', and to obtain insights into the sources of the strengths and weaknesses regarding both types of modeling tools. A controlled experiment was performed in which we addressed two types of modeling tools—desktop and cloud-based, in respect to two types of work—individual and e-collaboration. Within these treatments, we observed the productivity of 129 undergraduate IT students, who performed different types of modeling activities. The experimental participants reported no statistical significant differences between self-reported expertise with the investigated tools as well as their overall characteristics. However, they did finish individual and e-collaborative activities faster when using cloud modeling tool, where significant differences in favor of the cloud modeling tool were detected during e-collaboration. If we aggregate the results, we can argue that cloud modeling tools are comparable with desktop modeling tools during individual activities and outperform them during e-collaboration. Our findings also correlate with the related research, stating that with the use of state-of-the-art Web technologies, cloud-based applications can achieve the user experience of desktop applications.

Communicated by: Natalia Juristo

Electronic supplementary material The online version of this article (doi:10.1007/s10664-013-9280-x) contains supplementary material, which is available to authorized users.

G. Polančič (✉) · G. Jošt · M. Heričko
Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia
e-mail: gregor.polancic@uni-mb.si

G. Jošt
e-mail: gregor.jost@uni-mb.si

M. Heričko
e-mail: marjan.hericko@uni-mb.si

Published online: 27 September 2013

Keywords Modeling tools · CASE · Desktop computing · Cloud computing · Productivity · Collaboration

1 Introduction

The increasing complexity of software products and projects requires effective use of support tools that help to manage the assets involved within the software development lifecycle. These tools are, in their more general form, defined as Computer-Aided Software Engineering tools—CASE (García-Magariño et al. 2010). Kuhn (1989) defined CASE as *"the scientific application of a set of tools and methods to a software system which is meant to result in high-quality, defect-free, and maintainable software products"*. This term is mainly applied to those tools, concerned with the modeling of artifacts, also known as modeling tools (García-Magariño et al. 2010). Modeling tools represent an essential part of modern software development processes by supporting the creation of different types of conceptual models or diagrams, e.g.: data models, static-structure models, behavioral models or user interface models.

With advances in information and communication technologies (ICT), modeling tools began to support not only individuals involved in software development processes but also entire software project teams. Support for project teams means support for collaboration, which is defined as *"a mutually beneficial relationship between at least two people, groups or organizations, who jointly design ways to work together to achieve related or common goals and who learn with and from each other, sharing responsibility, authority and accountability for achieving results"* (Schuman 2006). Collaboration is an ubiquitous presence in our lives and a constant feature of modern society (Schmidt 1991). It is important because it has a critical impact on the success of any type of community (Patel et al. 2012). From a conceptual point of view, collaborating participants alter a collaboration entity (i.e. a common tangible or intangible asset), which is in a relatively unstable form and changes according to participants' interactions (Mamčenko 2004). In the context of this article, modeling tools' users represent collaborating participants, where the conceptual models represent those collaboration entities that are changing due to the usage of modeling tools.

As modeling tools began to support collaboration, they enabled computer supported collaborative work (CSCW) and became a type of groupware, which stands for *"computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment"* (Ellis et al. 1991). In the early stages, groupware mainly supported collaborative activities within colocated environments. However, nowadays, many teams or entire organizations require tools which enable effective and efficient collaboration within distributive environments. Instead of meeting face-to-face with colleagues in offices, these tools support collaborating activities in dislocated environments, which is commonly defined as e-collaboration (Serçe et al. 2011).

To summarize the above, despite different software architectures, successful modeling tools need to effectively support individual as well as team-based work in colocated and virtual environments. The most common are desktop modeling tools (hereinafter referred to as DMT). These tools are installed on a personal computer and so provide primary support for individual work. In the case of DMT, CSCW is commonly realized by the use of local servers that manage data exchange between computers. Besides DMT, innovations in Web technologies have introduced alternatives to the traditional desktop-based software model (Keller and Hüsig 2009). What modern Web applications' architectures have in common is that they require only a Web browser on the client's side. This facilitates availability, portability, and interactivity (Keller and Hüsig 2009), making them suitable for e-collaboration. With the innovation of cloud computing, the concept of Software-as-a-Service (SaaS) has been introduced. Out of these Web and cloud paradigms, cloud modeling tools (hereinafter referred to as CMT) have evolved.

In order to be successful, modeling tools have to leverage modelers' activities, allowing them to create valid models in a productive way. It can be presumed that valid models can be developed independently of the type of work and type of modeling tool in use. However, the question remains as to whether the architectural differences between DMT and CMT actually affect the productivity of a modeler. According to the related work (see Section 3), several researches already partially addressed this question, however none of them conducted an empirical research which would quantitatively investigate the benefits of CMT against DMT. The lack and the need for empirical evidence in cloud computing was also reported by other researchers, e.g. Chebrolu (2012), Opitz et al. (2012) and Alharbi (2012). So, our primary objective was to overcome this gap (i.e. lack of empirical evidence) by performing experimental research in which we investigated the productivity of individual and collaborative work (hereinafter referred to as individual and collaborative productivity) when using DMT and CMT. The collaborative work was limited to e-collaboration (collaborative work in distributed environments) since people increasingly interact through the internet on a professional and personal level (de Valck et al. 2009) (Wang 2006). Within experimental treatments, we observed the 'task times' and 'modeling corrections' of 129 undergraduate IT students, who conducted different types of modeling activities. Based on the primary objective, the research question was defined as:

> **RQ:** Is there any difference between the individual and collaborative work productivity in the cases of using DMT and CMT?

The secondary objective was to investigate the roots of potential differences between modelers' productivity when using both types of modeling tools. For this purpose: (1) we identified and divided individual and collaborative modeling activities into their constituent parts, and (2) evaluated the perceptual characteristics of both types of modeling tools, according to modeling tools users' personal opinions.

In order to achieve these objectives, we organized our research, and consequentially this article, as follows. The second section reviews individual work and e-collaboration, with the focus on the latter. In addition, productivity, the observed construct of our investigation, is defined, analyzed and operationalized. The third section reviews previous work related to the comparative analyses of desktop and cloud computing. The fourth section presents the details of the performed experimental research. The fifth section presents the results, whereas the last section interprets the results in light of the related work, reviews the limitations and implications of the research in theory and in practice, as well as defines future planned activities within the article's topics.

2 Research Foundations

Corresponding to the main elements of the article, we organized this section into following sub-sections: (1) introduction of desktop and cloud computing, (2) definition of e-collaboration based on the 3C-model and (3) definition of modeler's productivity.

2.1 Desktop and Cloud Computing

As already discussed in the introduction, e-collaboration requires technologies that help users work together with each other and also remotely (Mann 2011). In this light, traditional desktop computing, that is often viewed as *"an end-user environment, defined by a profile consisting of applications, documents and configuration data"* (VMware 2009), lacks collaboration capabilities (Marston et al. 2011). On the other hand, it is still assumed that desktop-based applications have richer functionality than their web-based counterparts (Marston et al. 2011). However, despite the introduction of standards and technologies that provide client-side programmability (e.g. JavaScript DHTML, Flash, and Silverlight), web applications have started to approach the experience of their desktop counterparts (van Ommeren et al. 2009).

Cloud computing emerged in 2007 as a new computing paradigm that represents an alternative deployment strategy for web applications (Wang et al. 2008). Besides changing the way that web applications need to be designed, it also has a lot in common with collaboration. Both cloud computing and collaboration cross the boundaries of an organization, are relevant to the relation between business and IT, and can have a major impact on the efficiency of organizations and IT (van Ommeren et al. 2009). In addition, both have been identified as a key business technology trends that will reshape enterprises worldwide (Xu 2012). Thus, adopting cloud computing, enterprise collaboration can happen at a much broader scale, since the cloud provides a collaborative environment (Xu 2012).

Although there is no formal definition of cloud computing as yet, The National Institute of Standards and Technology (NIST) defines it as "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (*e.g.*, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*" (Mell and Grance 2011). The key advantages of cloud computing, therefore, include (Marston et al. 2011): (1) lowering the cost of entry, (2) providing almost immediate access to hardware resources, (3) lowering IT barriers to innovations, (4) scalability of services, and (5) delivering services that were impossible before (e.g. mobile interactive applications that are location, environment or context aware).

One of the main characteristics of cloud computing is its treating of everything as a service (commonly addressed as XaaS) (Xu 2012). Cloud computing architecture is usually divided into three layers (Marston et al. 2011): Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS defines the processing, storage, networks, and other computing resources as standardized services over the network, where operating systems and software can be deployed. The services for developing, testing, deploying, hosting, and maintaining applications are provided by PaaS, whereas the SaaS represents a complete set of applications. Since our article focuses on CMT, it is limited to SaaS.

Garther defines SaaS as "*software that is owned, delivered and managed remotely by one or more providers*" (Desisto and Pring 2011), so it does not require installation on the client's computer (Marston et al. 2011). It is usually based on web-service technology and is run, hosted, and provided via the internet (Katzmarzik 2011). SaaS is already widely adopted within the fields of: (1) enterprise markets, such as customer relationship management and human capital management (Desisto and Pring 2011), (2) enterprise-level applications, such as Salesforce or Netsuite, and (3) personal applications such as Google Apps, TurboTax Online, Facebook, or Twitter (Marston et al. 2011).

Table 1 represents a comparison between the quality characteristics of desktop applications and SaaS, as reported by different researchers.

Table 1 Comparison of desktop applications and SaaS

Quality characteristics	Desktop applications	SaaS
Functionality	It is assumed that desktop-based applications have richer functionality than their web-based counterparts (Marston et al. 2011).	The "Rich Internet Applications – RIA" are able to approach the experience of their desktop counterparts in terms of functionality (van Ommeren et al. 2009).
Reliability	Some of the reliability issues can be associated with different operating systems, which may or may not support a specific application. Also, users have to be in direct contact with the computer to use the installed application (Quinn 2010).	Potential system outages or Internet network instability are considered as major risks in the field of SaaS (Benlian and Hess 2011). SaaS users are also dependent on the reliability of the vendor, to have the application online and running. Usually, if the application goes offline, users cannot proceed with their work until the service is restored (Quinn 2010).
Usability	Desktop applications allow user customization with the use of customizable toolbars and menu bars, which enable the most common functions to be just a mouse click away. Many desktop applications also have a selection of themes, which are sets of coordinated backgrounds, button and cursor styles (Dale 2012).	SaaS should be easy to use, capable of providing faster and reliable services. User Experience Driven Design aims to maximize the usability, desirability and productivity of the application (Xu 2012).
Efficiency	Performance is supposedly quicker on a desktop, because the screen is drawn only once and data is usually not transferred from the server, which additionally increases the time to display data (Sheriff 2002).	The computing resources are used more efficiently. The computers can also be physically located in geographical areas that have access to cheap electricity whilst their computing power can be accessed over the Internet (Marston et al. 2011).
Maintainability	According to Benlian and Hess (2011), two-thirds of the average corporate IT staffing budget goes towards routine support and maintenance activities. Also, user application has to be usually deployed manually to hundreds or thousands of users (Sheriff 2002).	SaaS shifts the responsibility for developing, testing, and maintaining the software application to the vendor (Benlian and Hess 2011).
Portability	Software is licensed on company's own information technology (IT) infrastructure (Katzmarzik 2011). When moving from one workstation to another, we have to worry about whether or not the application is installed on each workstation (Sheriff 2002).	With the lack of standards in the field of cloud computing, a customer might risk the possibility of vendor locking them into using their technology (vendor lock-in) (Marston et al. 2011).

By considering the quality characteristics of desktop applications and SaaS (Table 1), we specialized the defined research question (RQ) as follows:

RQ_1: Is the individual productivity in the case of using DMT better than the individual productivity in the case of using CMT?
RQ_2: Is the collaborative productivity in the case of using CMT better than the collaborative productivity in the case of using DMT?

2.2 3C-Model of Collaboration

Collaborative work and collaborative software are commonly represented in 3C-model of collaboration, as originally proposed by Ellis and Wainer (1994). The 3C-model defines collaboration as the combination of communication, coordination, and cooperation activities, which are defined as follows. Communication is related to the exchange of messages and information amongst people. Coordination is related to the management of people, their activities, and resources. Cooperation is the production that takes place within a shared workspace. As evident from Fig. 1, the 3C-model's activities are interrelated and organized into a cycle, starting with communicating activities during which team members negotiate, discuss, and make decisions.

The out-coming commitments are defined within the coordinating type of activities, where team members organize tasks, focusing on effectiveness and productivity. These planned tasks are performed via cooperative type of activities in which common artifacts are evolved. New challenges and ideas that arrive when cooperating are discussed in a new collaborative cycle, starting with new communicating activities. This reveals the iterative nature of the collaboration. The participants obtain feedbacks from their actions and 'feed-through' from the actions of their companions by means of 'awareness' information (shared workspace) related to the interaction amongst participants (Gerosa et al. 2003).

As evident from the 3C-model, collaborative work is a composite of different activities, performed by at least two participants. Besides, collaborative work can be hierarchically divided into their sub-tasks until all the subtasks at leaf nodes can be assigned to an individual (Daihwan et al. 1999). This also means that within a shared workspace, the individual and collaborative actions become entwined, and that common artifacts always result from a combination of individual and collaborative actions (Ferreira and Antunes 2007). In line with the 3C-model, the research question $\mathbf{RQ_2}$ has been further specialized into the following ones:

> $\mathbf{RQ_{2.1}}$**:** Is the productivity of communicating in the case of using CMT better than the productivity of communicating in the case of using DMT?
> $\mathbf{RQ_{2.2}}$**:** Is the productivity of cooperating in the case of using CMT better than the productivity of cooperating in the case of using DMT?

The above-stated research questions implicitly assume that communicating and cooperating activities can only be performed in the case of collaborative work. The research question, which would investigate the coordinating type of activities, was omitted from our research, since

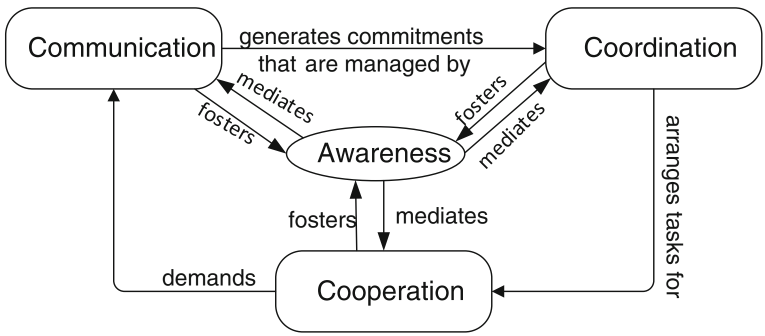


Fig. 1 3C model of collaborative work (Fuks et al. 2005)

coordinating activities impact the way team members communicate and cooperate. This means that coordination needs to be predefined in order to investigate the effects of DMT and CMT on communication and cooperation.

2.3 Productivity

Productivity is generally defined as the relationship between the quantity of output and the quantity of input used to generate that output. The output is commonly measured in physical quantity (e.g. number of goods produced) or financial value (e.g. value added), where the input can be measured according to the involved labor (e.g. number of hours worked or number of workers engaged) or capital spent to produce the output (SPRING Singapore 2011).

In our research, we focused on IT productivity, which investigates the contribution of IT to labor productivity. It can be measured as an increase in the output produced to the amount of labor spent on the production of this output due to procurement of IT (Oz 2005). This is also aligned with the 'in-use' productivity of a software product, which is defined as *"the capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use"* (ISO 9126–4 2004).

By considering IT productivity and the 'goal-question-metric' (GQM) approach (Basili et al. 1994; Briand et al. 1996), we defined GQM's measurement goal as to '**Analyze** individual and collaborative work when using DMT or CMT **for the purpose of** comparative evaluation **with respect to** IT productivity **from the viewpoint of** a modeler **in the context of** redesigning a pre-defined model'. The redesign of a pre-defined model (as defined in Section 4.3) represents an invariant work output, meaning that the resulting IT productivity depends on the input variable only (e.g. effort spent on the production of the output). This was considered in the GQM's questions, which were defined as: (1) 'How much effort do modelers spend to produce a pre-defined model in the cases of individual and collaborative work when using DMT or CMT?' and (2) 'How many corrections do modelers perform when producing a pre-defined model in the cases of individual and collaborative work when using DMT or CMT?'. Examples for 'corrections' include: undoing or deleting a part of the model, reshaping the elements of the model, and rearranging a connection between the elements of the model.

The first question's concept ('amount of effort') was operationalized with measuring 'task times' as defined in ISO/IEC 9126–4 (2004) where the second question's concept ('corrections') was operationalized by counting the 'corrections', which occurred during modeling activities (Table 2).

Table 2 Productivity metrics 'task time' and 'number of corrections' defined on ISO/IEC 9126–4 (2004) basis

Metric name	Task time	Number of corrections
Purpose of the metric	How long does it take to complete a task?	How many corrections are performed in a task?
Method of application	User test	User test
Measurement, formula, and data element computation	$X=T_a$; T_a=task time	$X=C_a$; C_a=number of corrections
Interpretation of measured value	0<= X, the smaller is better	0<= X, the smaller is better
Metric scale	Interval	Interval
Measure type	*T*=time	*C*=count
Input to measurement	User monitoring record	User monitoring record

3 Overview of the Related Work

In line with the research objectives and stated research questions, we searched for the related work. The search was performed by the following steps (Kitchenham and Charters 2007):

1. According to the research questions, we wanted to investigate if there had been any empirical research analyzing the differences between Web and desktop applications from the aspect of their productivity.
2. We defined the keywords to include any analysis that would compare any kind of Web application with its desktop counterparts. The search string included the terms ‘empirical research’ and ‘analysis’ in conjunction with ‘SaaS’, ‘Software as a Service’, ‘on the premise’, ‘desktop applications’ and ‘web applications’.
3. We identified those literature sources that were included in the overview of the related work. Our literature review included formal information sources as well as the grey literature. The former included ScienceDirect, Engineering Village, ProQuest, IEEE and ACM whilst the latter was limited to Google scholar and basic Google search.
4. The inclusion and exclusion criteria were formed. Inclusion criteria were as follows: (1) research, conducted between 2008 and 2013, (2) comparing Web and desktop applications from the aspect of usability, (3) analysis of Web and desktop applications in the light of their pros and cons, (4) empirical investigations, regarding the productivity of Web and desktop applications and (5) acceptance of SaaS. Exclusion criteria were as follows: (1) TCO analysis of SaaS, (2) addressing the development of SaaS applications, and (3) addressing Infrastructure as a Service (IaaS) or Platform as a Service (PaaS).
5. Finally, the search was performed using the defined search string upon the selected literature sources. The study selection was based on reading the titles, abstracts, and the full texts of papers.

After applying inclusion and exclusion criteria, we identified 16 relevant studies regarding the comparison between Web or SaaS and desktop applications.

Holzinger et al. (2010) discussed the pros and cons of using AJAX, a web development technique used to create asynchronous web applications. Their research showed that using AJAX can increase the usability of Web applications, since it allows them to look and feel like their desktop counterparts. Such applications function asynchronously in the background and are not interrupted or reloaded in any way. They found that some problems were still present (e.g. browser not supporting JavaScript, and loss of internet connection) but overall, the authors concluded that AJAX increases the usability of Web applications.

Even though the Web applications can have the look and feel of the desktop applications, an experimental investigation regarding the distribution of bugs in Web and desktop applications was performed by Torchiano et al. (2010). The experiment included 10 Web and 10 desktop applications where each pair belonged to the same type of software, with analogous features. The results of the research showed that Web applications are more defect-prone in the presentation layer than desktop applications (50 % vs. 35 %), based upon 1,472 identified bugs in the applications, altogether. The authors stated that the obtained empirical results were a consequence of the fact that the user interfaces of Web applications are more complex than those of desktop applications, the Web testing tools are immature and new technologies emerge at a higher rate. As a consequence of the latter, the Web applications are supposedly less stable than their desktop counterparts.

Regarding the cloud computing paradigm specifically, Chieu et al. (2009) addressed one of its main benefits, the ability to expand and add resources in a dynamic fashion, also known as 'scalability'. Scalability is critical to the success of many Web applications that provide services, which can suddenly become heavily demanded. Cloud computing can provide different resources on-demand including servers, storage or networking. The authors presented a dynamic scaling scenario and validated that the scaling capabilities of cloud computing are essential for providing higher resource utilization.

Iosup et al. (2011) also recognized cloud computing as an emerging commercial infrastructure paradigm that eliminates the need for maintaining computing facilities. However, these authors noted that scientific computer workloads differ from those of the web and small databases' workloads that are usually supported by cloud computing. Their research included an empirical evaluation of the performances of four commercial cloud computing services (Amazon EC2, GoGrid, ElesticHosts, and Mosso) in the light of scientific computing. The authors concluded that the computing performances of the tested clouds were low.

As the trend towards cloud computing increases, it is said that SaaS vendors are investing more on research and development (hereinafter referred to as R&D). Yang et al. (2010) explored the stated notion by empirically comparing the vendors of pure SaaS (6 cases) with on-premises/hybrid software (14 cases). The results of the analysis showed, that pure SaaS vendors spent less on R&D and more on marketing and sales than on-premises/hybrid software vendors. Also, the authors claimed that the traditional software model has competitive advantages when it comes to customization. Furthermore, even though the authors recognized that SaaS solutions will become more popular, they stated that this does not mean that SaaS will stimulate innovation.

Most of the identified related work analyzed the migration of applications to the cloud, e.g. (Bibi et al. 2012), (Godse and Mulik 2009), (Ju et al. 2010) and (Dillon et al. 2010). The strengths of SaaS include lower cost of entry (SaaS is delivered as a subscription model), easier implementation, configuration and maintainability, freedom of choice (users can switch from one SaaS to another), cost-effective scalability, easier accessibility, and platform independence. However, security and reliability are still considered as a drawback when discussing cloud computing. As the authors noted, putting the data and running the software on the hard disk of a third party is considered daunting to many. Besides the existing threats (losing the data, phishing, etc.), additional security issues have been introduced with cloud computing and its pooled computing resources. One of the main issues regarding the SaaS is also its interoperability, since there are no widely-adopted standards and the standardization process has just begun (e.g. The Cloud Computing Interoperability Forum, CCIF). Currently, each vendor has his own way regarding how applications interact with the cloud. As stated, many vendors impose the lock-in, which could outweigh the benefits that SaaS brings.

A study conducted by Wu (2011a) was set to explore the factors, which affect the adoption of SaaS. An empirical study was carried out, based on Technology Acceptance Model (TAM) and Rough Set Theory (RST). In addition to TAM, a novel framework was used that combines TAM with Diffusion Theory Model (TAM-DTM). The following constructs were observed: Social Influence (SI), Perceived Benefits (PB), Attitude toward Technology Innovations (ATI), Security and Trust (S&T), Perceived Usefulness (PU), Perceived Ease of Use (PEOU) and Behavioral Intention (BI). A questioner was mailed to 405 companies, with a 61 % response rate. The results demonstrated, that majority of the participants agreed with the statements "*Overall I intend to use the SaaS solutions in the future*" and "*Expert opinions affect me for using the SaaS solutions*". However, the participants were neutral when asked whether the security of data backups is determinant factor in SaaS adaption. The three key factors that significantly affect the use of SaaS solutions demonstrated to be SI, PU and S&T. On the other hand, no key factors belonged to PB, ATI and PEOU, presumably due to the fact that users are already aware of possible benefits of SaaS.

A related research was once again conducted by Wu (2011b). The author employed the TAM to SaaS adaptation, observing similar constructs as in aforementioned study: PB, ATI, S&T, PU, PEOU, SI and BI. In addition, Marketing Efforts (ME) was introduced into the model. The hypotheses of the research were formed regarding the influences between the defined constructs. In a survey, which was mailed to 120 participants of the Taiwan Style Competency Study Group, 42 participants responded. Based on the findings the author concluded that SaaS providers must put their efforts to shape beneficial SI, which affects most of the BI. On the other hand, the users should be focused on PU, PEOU and S&T assurances in order to maximize the advantage of SaaS. The author also noted that PEOU had no positive effects on PU, suggesting that merely improving PEOU will not necessarily increase PU.

Acceptance of cloud computing was also empirically validated in scope of German IT Departments by Opitz et al. (2012). The authors used TAM and a modified version, TAM2. A Likert items-based questioner was sent to 567 German IT personnel and in addition, an online version was also available. Ultimately, 100 responses were collected, where the results demonstrated the following. Firstly, authors noted TAM and TAM2 are appropriate to describe the acceptance of cloud computing. Secondly, cloud service providers should focus on raising prestige and image of their cloud services. Thirdly, the providers should empirically demonstrate the effectiveness of cloud computing relative to user's existing solutions. And fourthly, the users should evaluate the job relevance, output quality and usefulness of cloud services in order to decide whether or not the cloud service is beneficial to them.

A modified version of TAM2 was once again used by Du et al. (2013) when trying to establish the user acceptance of SaaS in the context of customers of China's e-commerce company, Alibaba. The proposed analytical framework had four major constructs, namely e-service quality, PU, SI and BI. Four rounds of questioners were conducted on six groups of samples. First four groups of samples helped to develop a scale to evaluate the e-service quality by using an open questionnaire. Based on 311 valid responses, four factors were emphasized, namely ease of use (EOU), security, reliability and responsiveness. The last two groups of samples were used to test the SaaS user acceptance model. A final version of closed questionnaire was sent to participants and 1,532 valid responses were collected. After analyzing the responses, PU was once again recognized as having the key impact on the usage of SaaS. SaaS providers should therefore identify the users' needs and develop a practical online software, which should be updated in accordance with the users' preferences. Another important factor demonstrated to be EOU, which has a significant positive impact on PU and BI. As authors suggest, cloud service providers should incorporate functions, such as operation guide setting, additional module mouse drag function, automatic data running and more visual and humanized interface design in order to increase the PEOU. In accordance with the aforementioned studies, SI had a significant direct impact on PU and BI as well. Authors thus suggest, that a word of mouth marketing and viral promotion should be carried out.

Acceptance of cloud computing was also examined in Saudi Arabia. As Alharbi (2012) points out, cloud computing literature lacks empirical studies regarding the users' acceptance of emerging technology. Similar to aforementioned related work, TAM was again utilized as a baseline, with five additional constructs, which are believed to affect the users' acceptance: gender, age, education level, job domain and nationality. The results of the study demonstrated that age, education, job domain and nationality significantly affect users' attitude towards use, whilst gender demonstrated to be of no significance.

Regarding the adaptation process, another study was conducted by Chebrolu (2012). The author stated that there is a lack of empirical evidence about how cloud adoption impacts individual aspects of IT effectiveness. In this light, the research was designed to study how cloud capabilities correlate with aspects of IT effectiveness. A survey was conducted and out of 4,075 eligible participants, 143 responded with at least one question answered. The results showed the prioritization of cloud compatibility over cloud connectivity and modularity. The implication of this is that IT executives should allocate more financial resources towards software and systems compatibility within cloud in order to improve their IT effectiveness. More specifically, the focal point should be on supporting multiple interfaces for external users to access cloud services and portability across multiple cloud providers.

On the other hand, Sun (2013) recognized the collaborative advantages that could-computing applications possess. As stated, collaborative SaaS can fundamentally change the way that the decisions are made, which was represented in the case of environmental decision support systems (EDSS). Author has migrated an EDSS module from a traditional client–server-based architecture to Google cloud-computing services. Scalability, session management and server backup were all delegated to the cloud. The results were as follows. Firstly, an increase in the collaborative decision making experience was noted. Secondly, the cost of small-scale EDSS was drastically reduced. Thirdly, author suggests that in the future, developers will be able to develop EDSS on Google App engine or other cloud services.

As can be seen from the overview of the related work, 16 studies were identified. The majority of the studies (6) address the acceptance of the cloud computing paradigm. TAM was proven once again as the predominant IT acceptance theory, since it was incorporated in most of the acceptance studies, with Chebrolu (2012) being a sole exception. Cloud computing paradigm is further covered either by analyzing the migration of applications to the cloud (4 of the relevant studies) or addressing benefits of cloud computing (4 of the relevant studies as well). The remaining researches deal with the trends in Web technologies, which strive to make the Web applications comparable to their desktop counterparts (2 of the relevant studies).

4 Experiment

While we were unable to find the answers to the stated research questions within the related studies, an experiment was performed within a laboratory environment. The experiment's participants had to perform modeling activities in an individual and collaborative manner by using two specific modeling tools—desktop and cloud-based. The experiment was designed and performed as explained in the following subsections.

4.1 Experiment Context

Different modeling tools offer support for different modeling domains and notations. In our experiment, we focused on the process-modeling domain. This domain was chosen because: (1) it is common in software engineering (Mili et al. 2010), (2) it is useful for different professions (engineers, analytics, business-domain experts), and (3) it is critical for successful business process management (Melao and Pidd 2000) (Schmietendorf 2008). Within the process-modeling domain, we investigated Business Process Model and Notation (BPMN)-based tools and models, because BPMN is already acknowledged as the leader and de-facto standard for business process-modeling (Shapiro et al. 2011).

At the time of performing the experiment, the BPMN's official website reported more than 70 modeling tools. Whilst we were unable to empirically investigate all of the available BPMN tools, we decided to select two comparable representatives based on the following criteria: (1) common characteristics of modeling tools, (2) high popularity of modeling tools, and (3) the common level of BPMN support. Based on these criteria, the latest versions of the following tools were selected: Bizagi process modeler as a representative of DMT, and Signavio process editor as a representative of CMT.

According to the first criteria, both modeling tools offer an advanced and user-friendly interface with following common capabilities: grouping of BPMN elements, quick competition, pie menu, syntax validation, generating of reports, birds-eye view, automatic layout and zoom in/out. In the light of popularity, Bizagi was evaluated as the preferred BPMN editor according to Chinosi and Trombetta (2011), and regarded as a popular industrial BPMN modeling tool by Yan et al. (2010). Our analysis of Bizagi showed that it offers valid and full support for BPMN 1.2. Signavio is based on the Oryx web editor, which was regarded as the best open source tool for BPMN 2.0 modeling (Chinosi and Trombetta 2011). Signavio is also regarded as a major academic modeling tool that focuses on BPMN (Yan et al. 2010). Our analysis of Signavio showed that it offers valid and full support for BPMN 1.2 and BPMN 2.0.

4.2 Experimental Design

According to stated research questions, a 'top-down' approach (from generic to specific) was used for formulating the null and alternative hypotheses. They were used to test the effects of the type of modeling tool (DMT and CMT) and type of work (individual and collaborative) on modelers' productivity (Table 3).

Based on the hypotheses, we identified two independent latent variables: 'type of work' and 'type of modeling tool'. In accordance with Section 2 of this article, we defined two 'type of work' levels: individual work and collaborative work. As already-mentioned in the introduction, the latter was limited to e-collaboration. The 'type of modeling tool' was also divided into two levels: DMT and CMT. The dependent latent variable of the research was productivity (P). Two independent latent variables, each with two levels implicated a 2×2 factorial design of the experiment with four treatments in total.

Productivity was operationalized with two empirical indicators, which were defined in Section 2.3. The primary indicator was 'task time' (T_n), which was used to test the stated hypotheses. The secondary indicator was 'number of corrections' (C_n). Since C_n was inadequate in the case of communicative activities, we were unable to use C_n to test all of the stated hypotheses (e.g. $H_{0-2.1}$). We implicitly presumed that lower 'task times' would positively correlate with lower 'number of corrections'.

By considering four treatments, and the fact that we were able to apply each treatment to every subject, we decided to perform a within-subjects experiment. However, with this kind of design, two experimental areas became challenging: (1) the length of the experimental process and (2) the controlling of order and sequence during treatments. These two issues were proactively addressed and resolved, as described below.

Table 3 Formal definition of the experiment hypotheses

RQ	Null hypothesis	Alternative hypothesis
RQ	H_0: Productivity (DMT) = Productivity (CMT)	H_a: Productivity (DMT) ≠ Productivity (CMT)
RQ_1	H_{0-1}: Individual productivity (DMT) = Individual productivity (CMT)	$H_{a\text{-}1}$: Individual productivity (DMT) > Individual productivity (CMT)
RQ_2	H_{0-2}: Collaborative productivity (DMT) = Collaborative productivity (CMT)	$H_{a\text{-}2}$: Collaborative productivity (DMT) < Collaborative productivity (CMT)
$RQ_{2.1}$	$H_{0-2.1}$: Communicating productivity (DMT) = Communicating productivity (CMT)	$H_{a\text{-}2.1}$: Communicating productivity (DMT) < Communicating productivity (CMT)
$RQ_{2.2}$	$H_{0-2.2}$: Cooperative productivity (DMT) = Cooperative productivity (CMT)	$H_{a\text{-}2.2}$: Cooperative productivity (DMT) < Cooperative productivity (CMT)

The length of the experimental process became challenging because during each of the four experimental treatments, real-world modeling activities had to be performed in order to ensure the effects were valid and reliable. In order to minimize the length of the experimental process, we decided to perform only collaborative modeling activities and to afterwards decompose them into individual ones (as explained in Section 2.2).

The second issue, the controlling of order and sequence during treatments, was addressed by counterbalancing, so that during each experimental treatment both, the individual and collaborative work, activities appeared on different locations. In addition, the sequence of using DMT and CMT was randomized (see R in Table 4), thus forming two groups of subjects (see G1 and G2 in Table 4) each starting the experiment with another 'type of modeling tool'.

Table 4 presents the overall experimental design. The independent variable X represents the 'type of work', whereas the variable Z represents the 'type of modeling tool'. The variables O present two main observations of two experimental treatments (see 'Total work' in Fig. 4).

Each treatment consisted of several individual and collaborative tasks (see 'X=iw+X=cw' in Table 4), which are described in the following subsections.

4.3 Structure of the Experimental Treatments

The main objective of the two experimental treatments (using DMT and CMT) was to replicate a real-word scenario of individual and collaborative process modeling in a controlled environment. To achieve this, experimental participants have to overcome modeling, communicating and collaborating challenges when designing a BPMN model in accordance to the referenced one. The reference BPMN model (hereinafter referred to as the 'reference model') substantively represented a Help-desk process and was divided into three main parts (BPMN pools): Customer process, Help-desk center process and Expert process. The reference model started within the 'Customer' pool as presented in Fig. 2. In case of a 'help desk call event', the 'Help-desk center' pool was activated in order to solve customer's problem. If no solution for the problem was found, the third pool ('Expert') was activated. These three pools corresponded to three experimental subjects who formed a modeling team (hereinafter referred to as 'the team').

The subjects, who formed a team, had to redraw the reference model, where the modeling activities were divided into individual and collaborative ones. Individual activities corresponded to individual process modeling. The collaborative activities consisted of overcoming common collaborative modeling challenges (e.g. solving problems when merging two partial models), communicating between team participants about common activities and sharing of common artifacts (e.g. sending a specific version or part of a model to other team members). Before and after performing the specified tasks, the times were recorded, where the differences between these times (T_{n_end} - T_{n_start}) represented the 'task times' (T_n) as defined in Table 2. Besides recording the 'task times', the 'number of corrections' (C_n) was recorded in the cases of individual and cooperative activities.

Table 4 Experimental design

		First treatment			Second treatment		
R	G_1	X=iw+X=cw	Z=dmt	O	X=iw+X=cw	Z=cmt	O
	G_2	X=iw+X=cw	Z=cmt	O	X=iw+X=cw	Z=dmt	O

R … random sampling; G_x … sequence of treatments according to the 'type of modeling tool'; X=iw … treatment 'individual work'; X=cw … treatment 'collaborative work'; Z=dmt … treatment 'DMT'; Z=cmt … treatment 'CMT'; O … observation

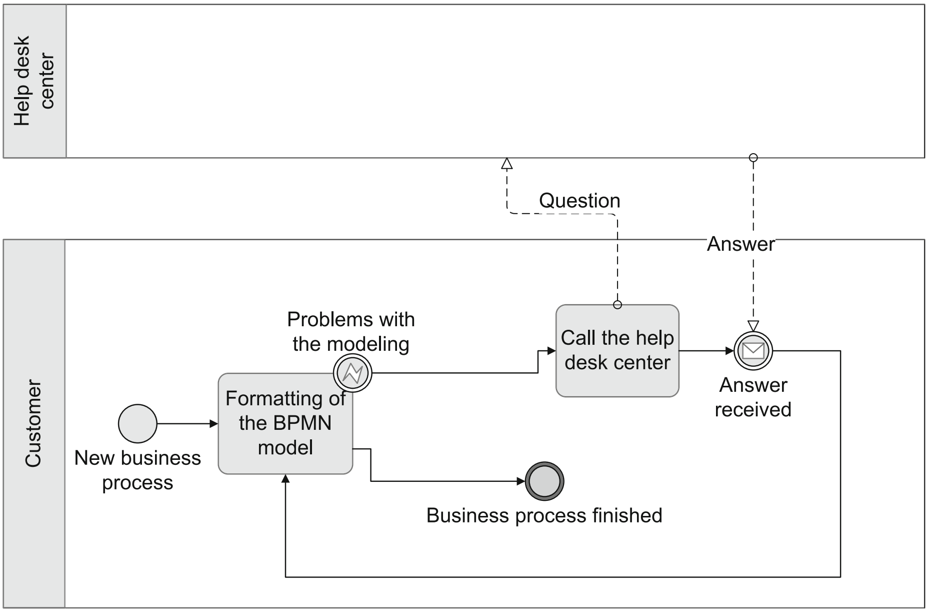


Fig. 2 Example of a partial BPMN model that had to be redrawn by an individual participant

4.3.1 Individual Activities

Individual activities were defined as activities that involved only one team member (one experimental subject). Within each individual activity, the subjects had to independently redraw different parts of the reference model. Each member received a different task, representing different parts of the aforementioned model. An example showing part of the BPMN model that had to be remodeled by a team member, is presented in Fig. 2.

4.3.2 Collaborative Activities

The collaborative activities were defined as those activities that involved more than one team member. In accordance with the 3C-model, they were divided into communicating, coordinating, and cooperating activities. Whilst coordinated activities impact the way team members communicate and cooperate, they were predefined in the experimental process, thus providing a constant for the experiment. As already stated, this was reasonable because otherwise the communicating and cooperating activities would differ between the teams, thus making any assumptions about the hypotheses invalid.

The communicative activities were represented by sharing of the partial BPMN models between team members, as well as the supporting e-communication. Depending of the type of modeling tool, sharing of partial models was performed in the following manner. In case of DMT, a modeler had to save a partial BPMN model and send it to two other modelers in the same team, by using a predefined asynchronous communication tool (an e-mail client). In case of CMT, the modelers worked on a common BPMN model and communicated with two other modelers of the same team by using a predefined communication tool. The conceptual difference between those two sharing approaches is presented in Fig. 3.

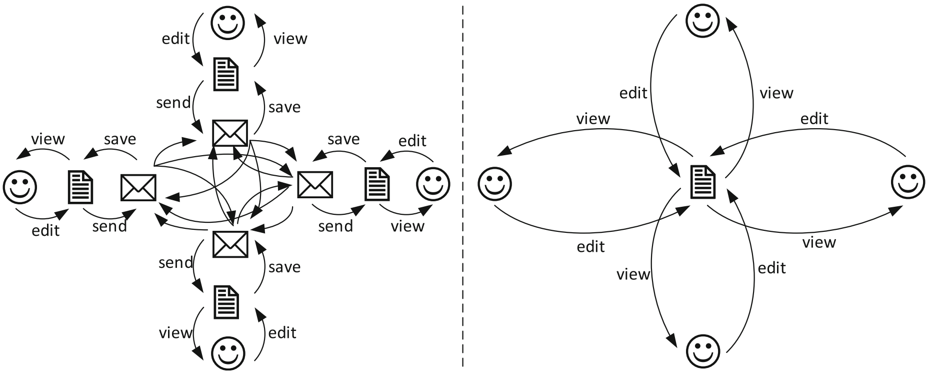


Fig. 3 Conceptual differences by using DMT (*left*) and CMT (*right*) collaboration, based on Maider (2007)

The cooperative modeling activities were actually performed by a single team member. However, they differed from the individual modeling activities because they appeared only within a collaborative environment. For example, one activity was related to the merging of partial BPMN models into a complete BPMN model. Despite the fact that the merging was performed by an individual, it required partial models of other team members. Thus, this activity was defined as a collaborative one.

4.3.3 Semantic Alignments

Although subjects received precise instructions regarding the modeling and collaborating activities, we became aware that each deviation from the given instructions might impact the resulting BPMN model. So, the last task in each experimental treatment was to validate the resulting BPMN model against the reference BPMN model, identify any potential deviations and remove them by using additional modeling activities. This task was finally validated by supervisors who ensured that all the modeling teams produced semantically equal BPMN models.

Semantic alignment represents a modeling activity. However, we classified the semantic alignment tasks neither as individual nor collaborative ones, for the following reasons: (1) this type of activity is absent within an actual working environment, where modelers do not have an opportunity to check their resulting models against a reference model; (2) the number of activities (semantic alignments) that had to be performed by a modeling team, were not predefined in the experimental design.

4.4 Measurement Model

The tasks that had to be performed by an individual or team of modelers during each experimental treatment are presented in Table 5, which provides following information (in columns):

1. Sequence of the experimental task (#),
2. Team member, responsible for performing the specified task,
3. Version of the BPMN model, before the task was performed (Input model version),
4. Description of the task, which was performed by the responsible team member,
5. Team members that were informed (affected) of the task outcomes,
6. Type of modeling activity (type of work),
7. Corresponding 'task time' measure,
8. 'Number of corrections' in the corresponding activity (Num. of corr.).

Table 5 Tasks and measures of the experimental treatments

#	Team member	Input model version	Description of the task	Informed team member	Output model version	Type of work	Task's time	Num. of corr.
1	A	–	Create the model as displayed in the picture.		1A	Individual modeling.	T_{i1}	C_{i1}
2	B	–	Create the model as displayed in the picture.		1B	Individual modeling.	T_{i2}	C_{i2}
3	C	–	Create the model as displayed in the picture.		1C	Individual modeling.	T_{i3}	C_{i3}
4	A	1A	Share the model with modeler C and receive feedbacks.	C	1A	Communication and sharing.	T_{S1}	/
5	B	1B	Share the model with modeler C and receive feedbacks.	C	1B	Communication and sharing.	T_{S2}	/
6	C	1A,1B, 1C	Merge the partial team members' models into a consistent model.		2	Cooperative modeling.	T_{CO}	C_{CO}
7	A	2	Improve the model as it is displayed in the picture.		2A	Individual modeling.	T_{i4}	C_{i4}
8	B	2A	Validate the collaboratively developed model against the reference model, and apply corrections if necessary.		3	Semantic alignment.	T_{SA}	C_{SA}
9	B	3	Share the final model with the rest of the team.	A,C	3	Communication and sharing.	T_{S3}	/

Based on Table 5, the aggregated values for individual and collaborative activities were calculated, as presented in Fig. 4.

As evident from the Fig. 4, the 'semantic alignment' type of activities was excluded from the measures of total work. This was in line with the definition of the semantic alignment (see Section 4.3.3).

4.5 Subjects, Groups, and Randomization

The ideal candidate for our experiment would be a random candidate from the population of modeling tools' users. For practical reasons, as well as for considering the context of the experiment (e.g. business process modeling), we searched for subjects within the subset of the population

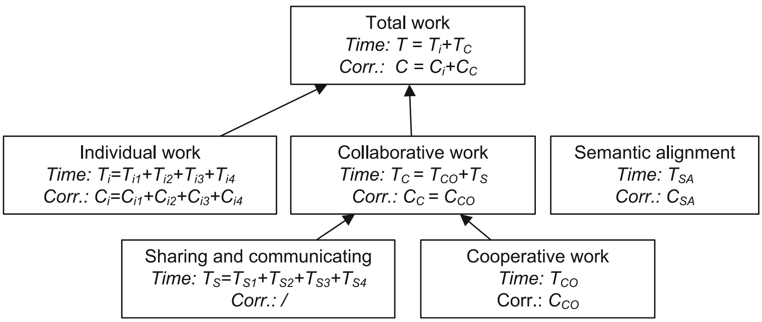


Fig. 4 Aggregated measures for different types of work

(sample frame). Our sample consisted of undergraduate IT students which participated on business process modeling-related course at the local university. According to the experimental design (4.2), the IT students were randomly assigned into teams consisting of three subjects. 43 teams were randomly formed from 129 subjects.

According to the experimental design, two treatments, each with individual and collaborative work, were performed by each team. The sequence of using DMT and CMT was randomly defined. This minimized any effects that might occur due to the sequence of the treatments.

4.6 Experimental Instrumentation

The experimental instrumentation consisted of subjects' guidelines, measuring instruments and modeling tools.

4.6.1 Subjects' Guidelines

Each subject (modeler) received paper-based instructions with predefined steps for the experimental treatment (Fig. 5). Whilst each team consisted of three members, each of them received unique instructions, with a predefined sequence of tasks (see the tasks and responsible team members, as presented in Fig. 5). The sum of all tasks within a team resulted in a consistent model, which was finally semantically aligned to the reference model.

All three types of instructions started with a common introductory speech addressing the participants and informing them about the nature and rules of the experiment. The following parts of the instructions were organized into a table, where each row of the table corresponded to a task, which had to be performed by a subject (see Table 5). The task was either related to experimental treatment (Table 5) or experimental observation (e.g. time recording or marking a correction). An application based on synchronized PC's system clock, which was permanently available on the screen, was used for the time recording.

4.6.2 Measurement Instruments and Questionnaire

Experimental observations were acquired using two instruments. The primary instruments were the previously described subjects' guidelines, which included blank table cells used for time and corrections recording. As presented in Fig. 5 the time recordings (T_x) were

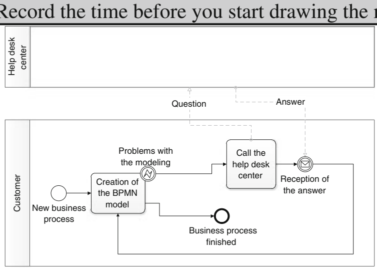
<table>
<tr><th>Task</th><th colspan="2">Description</th><th>Notes</th></tr>
<tr><td>3.1</td><td colspan="2">Record the time before you start drawing the model.</td><td>T_{3_start}= ___:___:___</td></tr>
<tr><td>3.2</td><td>
</td><td>Complete the pool "Customer", as it is shown in the picture on the left.
WARNING. In case of a Web-based tool, only one (1) person can work at once (simultaneously). Use synchronous communication tools to define the sequence of the modeling with the rest of the group.</td><td>Marking corrections (undo, delete), example (C_3):</td></tr>
<tr><td>3.3</td><td colspan="2">Record the time when you finished with modeling.</td><td>T_{3_end}= ___:___:___</td></tr>
</table>

Fig. 5 Example of part of a subject's guidelines

performed before and after each individual and collaborative tasks, where the correction recording (C_x) was performed instantly, when a correction occurred.

At the beginning and end of each experimental treatment, the subjects were instructed to complete part of an online questionnaire that was divided into the following sections:

- section 1: demographic questions (e.g. questions about the participant's gender, age, and experience with BPMN modeling),
- section 2: basic questions regarding any experience the subjects had had with the tools selected for the experiment,
- sections 3 and 6: questions relating to the individual activities performed regarding the first and second modeling tools. These sections also included fields for rewriting the times and numbers of corrections, as stated in the instructions,
- sections 4 and 7: questions relating to those collaborative activities performed using the first and second modeling tools. These sections also included spaces for recording the times and numbers of corrections, as stated in the instructions,
- sections 5 and 8: questions relating to the overall perceived characteristics about the first and second modeling tools,
- section 9: the last section included an optional open question soliciting any comments regarding the experiment.

Most of the items were evaluated on a seven-point Likert's scale, meaning that the participants were asked to indicate their level of agreement with a particular statement on a seven-point scale, with the endpoints 'strongly agree' and 'strongly disagree'. The time was recorded in the time format (hh:mm:ss), where the 'number of corrections' was recorded on a positive natural numbers interval.

4.7 Operation

The experiment was pre-tested, performed, and controlled in the following manner. As already described, the experiment was organized within business process modeling related lectures, where the practical part of the course was proactively organized in such a way that the students received equal amounts of training using both modeling tools. In this way, any effects regarding any differences in prior knowledge or experience that could have affected the experimental results were minimized. In addition, students became familiar with the modeling technique and domain, as was later used during the experiment. In parallel with the training of the students, we started to prepare the environment, procedures, and instruments for the experiment. The experiment's environment was set up in a computer classroom with 30 PCs all based on the same hardware and software configurations. This was later validated with a PC benchmark tool, which demonstrated that all the PCs performed equally.

Because of the complexity of the experimental treatments, several pre-tests were necessary to ensure that the experiments' subjects would understand and perform the experiments in a consistent manner. So, the pre-test resulted in improvements in the subjects' guidelines (e.g. improved consistency of terms, simplified sequence of modeling steps, increased space for recording observations, and improved visibility of the models' elements), and the online questionnaire's design. In addition, the experiment's supervisors defined consistent procedures for performing the experiment. The latter was necessary since only ten teams (30 students) performed the experiment at a time. A maximum set of ten teams performing the experiment simultaneously, was defined as an 'experimental unit'.

Each experimental unit started with supervisors' instructions regarding the experimental procedure and explanations regarding personal data privacy. The experiment continued with the randomization, as well as testing the modeling and communication tools to see if they were working properly. Whilst no direct communication was allowed between team members, each modeling team shared e-mail and instant-messaging contacts when testing the corresponding tools. These allowed the sharing of artifacts, synchronous, and asynchronous communication. Those experimental supervisors recorded the execution of the experiment, focusing on any potential experimental process exceptions that could affect the results.

4.7.1 Data Validation

At the end of each experimental unit, the subjects' guidelines with records were collected and validated against unintentional errors. These might have occurred since the subjects recorded the times and corrections manually on the guidelines, and afterwards rewrote them into an online questionnaire. In order to minimize risks all the data recordings were rechecked and, if necessary, corrected by the experiment's supervisors. In addition, those models produced by modeling teams were validated against the reference model, thus ensuring that all the modeling teams completed their tasks equally.

5 Data Analysis and Interpretation

Data analysis was performed using SPSS Statistics 20. For the purpose of testing the differences between DMT and CMT, several unpaired *t*-tests were performed by considering the following assumptions: (1) each of the two populations being compared should follow a normal distribution, (2) the two populations being compared should have the same variance and (3) the data used to carry out the test should be sampled independently from the two populations being compared.

The data analysis section is organized into following parts: (1) descriptive statistics of the subjects, (2) descriptive statistics of the perceived characteristics regarding the investigated modeling tools and (3) descriptive and inferential statistics regarding the defined measurement models, which were used to test the stated hypotheses.

5.1 Subjects' Statistics

As already-mentioned, 129 participants were involved in the experiment, forming together 43 teams. Most of the participants were male (89.7 %) and reported to be in the majority (94.6 %) familiar with at least the basic scope of the BPMN elements (Fig. 6). This type of BPMN expertise was sufficient, since the reference model was only conducted on a basic set of BPMN elements.

Because the participants worked in teams, each team's self-reported expertise (calculated as an average of the individual ones) with the representative tools, was analyzed by using Mann-Whithey U test. The results demonstrated that there was no statistically significant difference between DMT and CMT groups' median self-reported expertise (U=769.5, p=0.176).

Figure 7 represents the boxplots showing self-reported expertise with the tools, measured on a seven point scale with higher values preferred.

Taking the above data into consideration, we can conclude that an average team consisted of male participants who were familiar with the basic scope of BPMN and had similar expertise with investigated modeling tools.

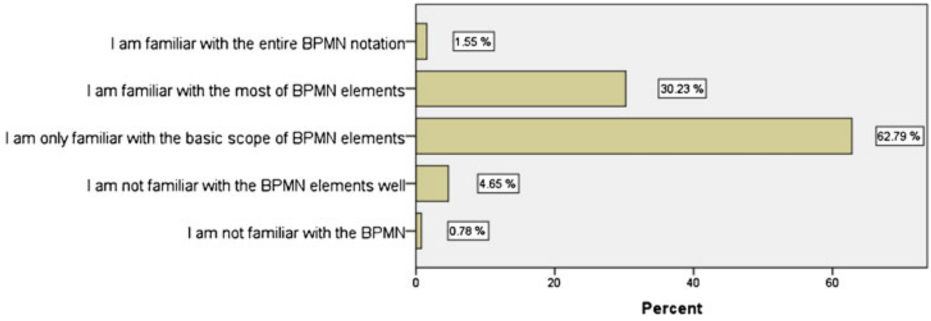


Fig. 6 Self-reported BPMN expertise

5.2 Modeling Tools' Statistics

During the experiment, the modeling tools were investigated according to the users' personal opinions about the characteristics of the investigated tools (hereinafter referred to as 'perceived tools characteristics'—PTC) (Table 6).

Figure 8 presents bar charts for the reported PTC regarding the investigated tools, with upper values preferred (7-strongly agree, 6-agree, 5-partially agree, 4-neutral, 3-partialy disagree, 2-disagree, 1-strongly disagree).

As evident from Fig. 8, the subjects' responses to the questions regarding PTC produced means ranging between partially agree (5.22) and agree (6.02). The total PTC score (calculated as an average of the individual ones) was 5.54 for DMT and 5.68 for CMT.

The subjects agreed that both modeling tools had understandable and easy to use icons of the elements (PTC1). Compared to CMT, DMT was reported as slightly more mature and predictable in the case of an error (PTC4). This is understandable, since CMT operates through the browser (Mozilla Firefox in our case). Still, the CMT was perceived as being as reliable as its desktop counterpart (PTC3) and even more powerful when considering the resources in use (PTC5). When regarding the modeling aspect of the tool, the connecting, naming and positioning of the elements was perceived as better in CMT (PTC6, PTC7 AND PTC8). The model itself looked more as expected in the case of CMT (PTC9). Though it might differ between vendors, it is obvious that the CMT have progressed to a point where

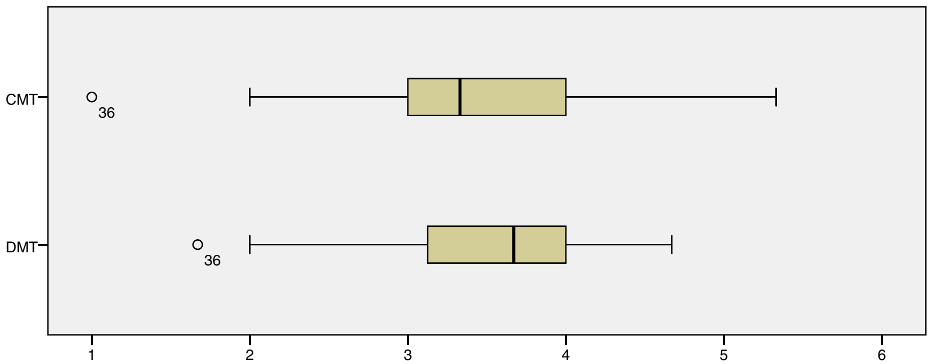


Fig. 7 Teams' self-reported expertise with DMT and CMT

Table 6 Definitions of perceived tools' characteristics—PTC

Abbreviation	Item description	Derived from
PTC1	Understandability and usage of the tool's elements/icons are simple.	(Legris et al. 2003)
PTC2	Availability, quantity and quality of the tool's documentation are good.	(Roca et al. 2006)
PTC3	I can rely on the tool.	
PTC4	The tool is mature and has a predictable response in case of an error.	(Mohagheghi 2010)
PTC5	The tool is powerful considering the computing resources it uses.	(Holden and Karsh 2010)
PTC6	Using the tool's BPMN connecting elements works well.	(Legris et al. 2003)
PTC7	The naming of the tool's BPMN elements works well.	
PTC8	Positioning of the elements of the model works well.	
PTC9	Model in the tool looks exactly like I expected it.	
PTC10	Visual elements are in accordance with the BPMN notation.	(Mohagheghi 2010)
PTC11	The user interface is clear.	(Thong et al. 2002)
PTC12	The user interface provides a good overview on more complex BPMN models as well.	
PTC13	The tool works smoothly (it never froze).	New item

they can even surpass their desktop counterparts in the light of functionality. However, the user interface still slightly favors the DMT (PTC1 and PTC12).

Due to the discrete nature of Likert items used for the measures of PTC, we were unable to determine the significances of the differences between the PTC's means (Clason and Dormody 1994). However, this precondition was met on the total scores, which are generally more normally distributed (Winter and Dodou 2010).

The following assumptions for performing the *t*-test on the total scores were met. Firstly, the sample size was relatively large, since 129 subjects evaluated PTC of both tools. Secondly, the normality of the total scores distribution was analyzed through the skewness and kurtosis divided by their standard error (z-scores). Both values were below 3.29 (Martin and Bridgmon 2012), meaning that the assumptions for performing *t*-test were met. The

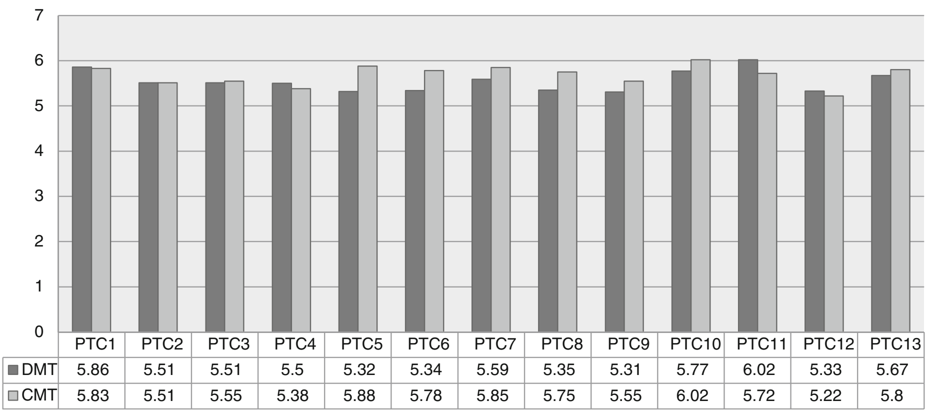


Fig. 8 Results of reported PTC

results from the performed t-test showed no significant differences between the means of the total PTC scores for the representative tools (t=−1.341; df=250; sig. (2-tailed)=0.181).

5.3 Data-Set Winsorization

Several outliers were detected despite the proactive activities performed to minimize unintentional errors in the cases when recording the 'task times' (see Section 4.6.1). Their impact on the population mean was reduced by data Winsorizing. Winsorization is a method for reducing the effects of extreme values in a sample. It is based on the transformation of statistics by limiting extreme values within the statistical data to reduce the effect of possibly spurious outliers. Winsorized estimators are usually more robust to outliers than their more standard forms such as trimming, where the outliers are simply excluded from the data set (Dixon and Yuen 1974).

We performed a typical Winsorizing strategy and set all outliers to a specified 90th percentile of the data. This meant that all data below the 5th percentile was set to the 5th percentile, and data above the 95th percentile, was set to the 95th percentile, thus representing an outlier. The resulting, winsorized data set, was afterwards used to evaluate the measurement model statistics, as presented next.

5.4 Measurement Model Statistics

The measurement model (as described in Section 4.4) represented the focal observations of our research in which we investigated two empirical indicators of modelers' productivity—'task time' (T_n) and 'number of corrections' (C_n). Table 7 shows the results of t-tests that were performed for each of the defined T_n during the experimental treatments.

Table 7 demonstrates that the T_n means of the CMT were lower as the T_n means of the DMT in all cases, whereas significant differences were detected in the cases of collaborative activities. These findings are graphically supported by corresponding boxplot diagrams, as presented in Fig. 9. The X-axis represents the T_n measures for both types of modeling tools in use (see also Table 7), and the Y-axis represents the corresponding results (measured in seconds, with the lower values preferred).

Table 7 Measurement model statistics—T_n

Seq.	Task's time measure (T_n)	Mean		Standard deviation		T-value	Sig. (2-tailed)
		DMT	CMT	DMT	CMT		
1	T_{i1}	434.628	387.326	157.629	151.917	−1.417	0.160
2	T_{i2}	653.837	609.605	254.519	265.717	−0.788	0.433
3	T_{i3}	401.395	393.280	135.498	130.905	−0.282	0.778
4	T_{S1}	547.302	227.326	393.305	222.355	−4.644	0.000[a]
5	T_{S2}	536.605	143.116	287.444	107.108	−8.412	0.000[a]
6	T_{CO}	339.023	202.930	244.457	157.738	−3.067	0.003[b]
7	T_{i4}	226.116	211.721	100.925	93.300	−0.687	0.494
8	T_{SA}	254.581	214.698	120.515	93.206	−1.717	0.090
9	T_{S3}	140.000	76.605	92.812	65.109	−3.667	0.000[a]

[a] T-test is significant at the 0.001 level (2-tailed)

[b] T-test is significant at the 0.01 level (2-tailed)

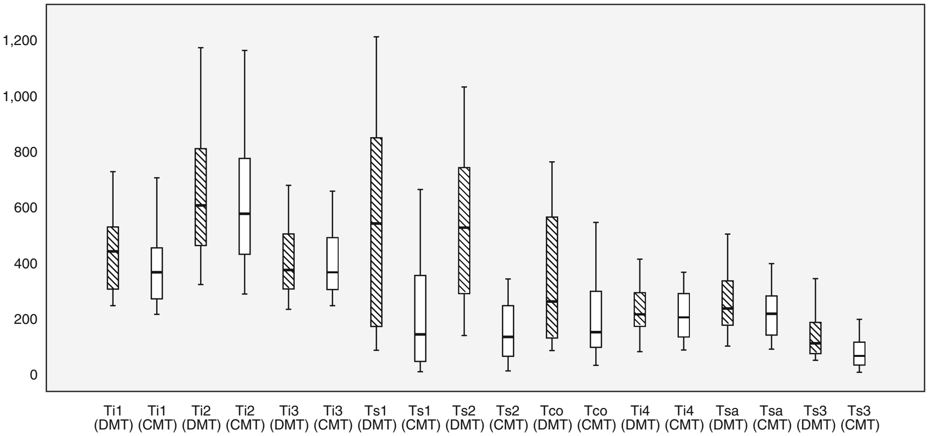


Fig. 9 Boxplots for measurement model statistics—T_n

The secondary productivity measure was related to the 'number of corrections' (C_n), which were performed during the individual and cooperative activities. While the corrections could not apply to communicating activities, the tasks' sequences 4, 5 and 9 (as defined in Table 5) were excluded from Table 8.

The analysis of the 'number of corrections' demonstrates that users performed fewer corrections when using CMT. Significant differences were detected in the cases of cooperative activities (C_{CO}) and semantic alignment (C_{SA}). In addition to the basic T_n and C_n measures, the aggregate measures for specific types of modeling activities were calculated (Table 9), as defined in Fig. 4.

Table 9 presents the results of the analysis regarding 'task times', where the means of all aggregated T_n were lower in the case of using CMT. The significant differences between aggregated T_n means were detected during collaborative (sig. 0.000) and total activities (sig. 0.000). Figure 10 presents the corresponding boxplot diagrams of the aggregated T_n. The X-axis presents the different types of T_n measures and the Y-axis the corresponding results (in seconds, with the lower values preferred).

Table 8 Measurement model statistics—C_n

Seq.	Corrections measure (C_n)	Mean		Standard deviation		*T*-value	Sig. (2-tailed)
		DMT	CMT	DMT	CMT		
1	C_{i1}	0.26	0.14	0.539	0.467	1.069	0.288
2	C_{i2}	8.84	7.67	3.709	4.502	1.307	0.195
3	C_{i3}	1.86	1.79	1.740	1.909	0.177	0.860
6	C_{CO}	2.95	0.65	4.287	1.173	3.397	0.001[a]
7	C_{i4}	1.91	1.30	2.136	1.551	1.502	0.137
8	C_{SA}	2.44	1.28	2.548	1.485	2.586	0.011[b]

[a] *T*-test is significant at the 0.001 level (2-tailed)

[b] *T*-test is significant at the 0.05 level (2-tailed)

Table 9 Measurement model statistics – aggregated T_n

Seq.	Task's time measure (T_n)	Mean		Standard deviation		*T*-value	Sig. (2-tailed)
		DMT	CMT	DMT	CMT		
1	$T_i=T_{i1}+T_{i2}+T_{i3}+T_{i4}$	1715.977	1601.930	381.0177	395.792	−1.361	0.177
2	$T_C=T_{CO}+T_S$	1562.930	649.977	809.653	347.759	−6.794	0.000[a]
2.1	T_{CO}	339.023	202.930	244.458	157.738	−3.067	0.003[b]
2.2	$T_S=T_{S1}+T_{S2}$	1223.907	447.047	638.614	282.863	−7.294	0.000[a]
3	T_{SA}	254.581	214.698	120.515	93.206	−1.717	0.090
4	$T=T_i+T_C$	3278.907	2251.907	977.191	561.743	−5.975	0.000[a]

[a] *T*-test is significant at the 0.001 level (2-tailed)

[b] *T*-test is significant at the 0.01 level (2-tailed)

Corresponding to aggregated T_n, the aggregated C_n were calculated (Table 10) as defined in Fig. 4. As evident from Table 10, the mean values of the aggregated C_n were lower when using CMT, where the significant differences were detected in the cases of cooperative activities (sig. 0.001), semantic alignments (sig. 0.011), and total modeling activities (sig. 0.006).

Finally, both types of productivity indicators (T_n and C_n) were compared for potential correlations. Since C_n could not be applied to the communicative (sharing) type of activities (as defined in Section 4.2), the total work was also calculated without this type of activity in the case of measuring 'task times'. The results of Pearson's correlation test are summarized in Table 11.

Based on the summary of Pearson's correlations results (Table 11), we can conclude that productivity indicators T_n and C_n are significantly and positively correlated.

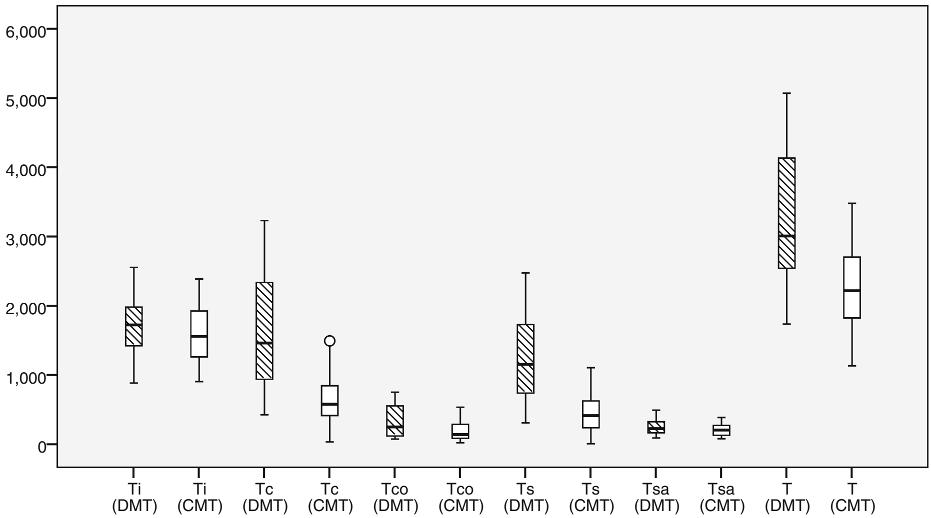


Fig. 10 Boxplots for measurement model statistics—aggregated T_n

Table 10 Measurement model statistics—aggregated C_n

Seq.	Corrections measure (C_n)	DMT		CMT		*T*-value	Sig.(2-tailed)
		Mean	Std. dev.	Mean	Std. dev.		
1	$C_i=C_{i1}+C_{i2}+C_{i3}+C_{i4}$	12.86	5.272	10.91	6.055	1.596	0.114
2	C_{CO}	2.95	4.287	0.65	1.173	3.397	0.001[a]
3	C_{SA}	2.44	2.548	1.28	1.485	2.586	0.011[c]
4	$C=C_i+C_{CO}$	15.81	7.654	11.56	6.265	2.821	0.006[b]

[a] *T*-test is significant at the 0.001 level (2-tailed)

[b] *T*-test is significant at the 0.01 level (2-tailed)

[c] *T*-test is significant at the 0.05 level (2-tailed)

5.5 Hypotheses Testing

The hypotheses were evaluated after the measurement model analysis using a 'bottom-up' approach (from specific to generic). Table 12 presents the hypotheses, the corresponding types of work, and our findings. Based on the aggregated T_n measures (Table 9), we either failed to reject or rejected the null hypothesis in favor of the corresponding alterative hypothesis (Table 3).

As evident from the table above, both DMT and CMT performed similarly in the case of individual work. Thus, we failed to reject the null-hypothesis H_{0-1} stating that there is no difference between the modeler's productivity of DMT and CMT in the case of individual work. However, the null-hypothesis regarding the collaborative work (H_{0-2}) was rejected in favor of the corresponding research hypothesis (H_{a-2}), since there were significant differences between DMT and CMT in the light of modeler's productivity when performing communicative and cooperative types of activities. Finally, we rejected the main null hypothesis (H_0) stating that there were no differences between the productivities of DMT and CMT, since the total productivity when using CMT was better compared to the total productivity in the case of using DMT.

In order to reduce the chances of obtaining false-positive results when multiple tests are performed on a single set of data, the Family wise error rate (FWER) was additionally considered by using a single step 'Bonferroni correction' method (Bland and Altman 1995). By considering this method, the experiment-wide type 1 error rate of 0.05 was divided by the number of investigated hypotheses (Table 3). While the highest significance level within the rejected null hypotheses was 0.003 ($H_{0-2.2}$) the corrected significance level still supports the conclusions, presented on Table 12.

Table 11 Pearson's correlations between productivity indicators (T_n and C_n)

Task type	Task time measure (T_n)	Corrections measure (C_n)	Pearson's correlation (*N*=86)	Sig. (2-tailed)
Individual work	$T_i=T_{i1}+T_{i2}+T_{i3}+T_{i4}$	$C_i=C_{i1}+C_{i2}+C_{i3}+C_{i4}$	0.378	0.000[a]
Cooperative work	T_{CO}	C_{CO}	0.560	0.000[a]
Semantic alignment	T_{SA}	C_{SA}	0.372	0.000[a]
Total work	$T=T_i+T_{CO}$	$C=C_i+C_{CO}$	0.556	0.000[a]

[a] *T*-test is significant at the 0.001 level (2-tailed)

Table 12 Hypothesis testing

Null hypothesis	Type of work	Findings
H_{0-1}	Individual	**H_{0-1} failed to reject.** Based on the *t*-test, we failed to reject the H_{0-1}. There were no significant differences (sig. 0.177) between the 'task time' means of DMT and CMT detected in the case of individual work (see also Table 9).
$H_{0-2.1}$	Collaborative/ communication	**$H_{0-2.1}$ rejected in favor of $H_{a-2.1}$.** Based on the *t*-tests, we rejected the $H_{0-2.1}$ in favor of the corresponding $H_{a\text{-}2.1}$. The communication work (sharing of models) was done significantly faster (sig 0.000) when using the CMT, compared to DMT.
$H_{0-2.2}$	Collaborative/ cooperation	**$H_{0-2.2}$ rejected in favor of $H_{a\text{-}2.2}$.** Based on the *t*-test we rejected the $H_{0-2.2}$ in favor of the $H_{a\text{-}2.2}$ hypothesis, since the cooperative work was done significantly faster (sig. 0.003) using the CMT, compared to its desktop counterpart.
H_{0-2}	Collaborative	**H_{0-2} rejected in favor of $H_{a\text{-}2}$.** Based on the *t*-test we rejected the H_{0-2} in favor of the corresponding $H_{a\text{-}2}$. The total 'task time' when performing communicating and coordinating activities was significantly lower (sig 0.000) in case of using CMT.
H_0	Total work	**H_0 rejected in favor of H_a.** Based on the performed *t*-test, we rejected the H_0 in favor of the corresponding H_a. As evident from Table 9 the total work was done significantly faster (sig 0.000) using the CMT when compared to DMT.

6 Discussion

The performed research resulted in the following new insights. The experiment's participants reported similar personal opinions about the overall quality of the investigated modeling tools (total PTC score), as well as similar self-reported expertise with the BPMN. On the level of individual tool characteristics, no *t*-tests were performed due to the discrete nature of the measurement items. Nevertheless, some minor differences in the tool characteristics' means were detected. Despite slightly preferring desktop modeling tool's (DMT) user interface (PTC11, PTC12), the experiment's participants reported that the cloud modeling tool (CMT) slightly outperforms the desktop one in cases of naming, positioning, and connecting BPMN elements (PTC6, PTC7, PTC8). This finding is aligned with the 'number of corrections' which were performed within individual modeling activities (C_i). Despite the fact that individual modeling activities were performed equally in both modeling tools, the subjects reported less corrections (e.g. fewer rearrangements, less reshaping, and easier connecting of BPMN elements) when using the cloud modeling tool.

In the light of the productivity, the participants finished their tasks faster and with less required corrections when using the cloud modeling tool, where the significant differences between the 'task times' (T_n) and 'number of corrections' (C_n) means were detected in all the collaborative activities, as well as in the total work. In addition, it was demonstrated (Table 11) that the number of performed corrections (C_n) positively and significantly correlates with 'task times' (T_n). This might be explained as follows: since performing a correction requires time, it can be concluded that the modelers performed their collaborative activities faster when using cloud modeling tool, also due to less corrections they made to the models. Beside these new insights, several alignments to the related work (Section 3) have been identified (Table 13).

As evident from Table 13, the majority of related work was in alignment with our findings except of the Yang et al. (2010) study, which findings could not be reconfirmed. The following

Table 13 Relations between the related work and our findings

Authors	Main findings of the related work	Relations with our findings
(Holzinger et al. 2010)	AJAX allows Web applications to look like their desktop counterparts. It can also increase the usability of a Web application.	Our results are in alignment with the findings of Holzinger et al. (2010). The experiment participants reported no significant differences between the total 'perceived tool characteristics' of the investigated CMT, which is AJAX-based, and DMT (Fig. 8).
(Chieu et al. 2009)	The article focused on the scalability and demonstrated that the SaaS is capable of handling sudden load surges, delivering IT resources on-demands to users, and maintaining higher resource utilization,	Our results are in alignment with the findings of (2009). During our experiment, thirty participants were modeling at the same time (as described in Section 4.6) and reported that the CMT worked smoothly. Also, CMT was recognized as reliable and powerful, considering the computing resources it uses (see PTC5 on Fig. 8).
(Torchiano et al. 2010)	The results of the experiment showed that Web applications are more defect-prone in the presentation layer than desktop applications (50 % vs. 35 %).	Again, our results align with the findings of Torchiano et al. (2010), since DMT was reported as slightly more mature and predictable in the case of an error when compared to CMT (see PTC4 on Fig. 8).
(Yang et al. 2010)	SaaS vendors supposedly spent less on R&D and more in marketing and sales than desktop vendors. The authors also doubt that SaaS will bring innovation, even though its popularity will rise.	The results of our study could not confirm the presumptions made by (Yang et al. 2010). With the introduction of collaborative modeling, commenting capabilities and the concept of a dictionary, the investigated CMT appeared to be superior to its desktop counterpart. The participants also perceived the CMT to be superior to DMT in the light of its appearance and usability (see PTC6 to PCT10 on Fig. 8). So, we can claim that cloud computing vendors do not necessarily spend less on R&D as was stated in the corresponding article.
(Wu 2011a)	When exploring the significant factors affecting the adoption of SaaS, the authors noted that users strongly agreed with the statement "*using the SaaS solutions enables me to do things faster*".	The results of our experiment complement the findings of Wu's (2011a) article. In case of CMT, users agreed that connecting, naming and positioning of elements worked well. When compared to DMT, users found CMT to work slightly better (see PTC6, PTC7 and PTC8 on Fig. 8).
(Opitz et al. 2012)	As authors suggested, the users should evaluate output quality among other characteristics when deciding about the usage of cloud services.	Our study is complementary to the results of Opitz et al. (2012) study, since the participants noted that the model itself (representing the output) looked more as expected in the case of CMT (see PTC9 in Fig. 8), showing a higher output quality when compared to DMT.
(Du et al. 2013)	After analyzing the acceptance of SaaS, authors suggested that operation guide setting, additional module mouse drag function, automatic data running and more visual interface design should be the priority of the service provider.	The results of our study are in accordance to the findings of (2013). Several functions (e.g. positioning of the elements) worked well in CMT. The users also agreed that the user interface of CMT was clear. However, DMT had a slight advantage over CMT when comparing clarity of the user interface (see Fig. 8).

Table 13 (continued)

Authors	Main findings of the related work	Relations with our findings
(Sun 2013)	Author noted an increase in the collaborative decision making experience when migrating the software to the cloud.	The results of our study complement the results of Sun's (2013) article, since there were significantly differences in total collaborative work when comparing DMT and CMT in favor of the latter (see Section 5.5).

subsections define the validity evaluations and present the theoretical and practical implications of the results. Subsequently, we discuss the possibilities for future work and finally, summarize the article.

6.1 Validity Evaluation

Several concerns regarding the internal, construct and the external validity (discussed in the summary) were investigated according to Trochim and Donnelly (2006) and Neuman (2005). In respect to the internal validity, the following threats were considered and controlled (Table 14).

The constructs validity threats, which involve generalizing from the measures to the concepts of the study, referred mainly to the dependent variable of the research. As explained in Section 2.3, we measured productivity primary with 'task time.' The construct validity was additionally tested by the secondary productivity measure – 'number of corrections'. With Pearson's correlations test (Table 11), we demonstrated that both productivity indicators correlate significantly.

Table 14 Management of internal validity threats

Internal validity threats	Implications and countermeasures
Selection bias	Controlled with random assignment (4.2).
History	Controlled with simultaneous recording of the experimental observations (see Section 4.6.2).
Maturation	Psychological or emotional processes within the experimental subjects could impact the answers regarding the subjective evaluation of the modeling tools' characteristics (Table 6).
Testing	Due to the factorial design of the experiment, pre-testing did not impact the dependent variables of the experiment.
Instrumentation - PC's system clocks	The accuracy of PC's system clocks was controlled with pre-experiment clock synchronization.
Instrumentation - subjects guidelines	Manual recording of tasks' times and modeling corrections (see notes section in Fig. 5) and their rewriting into the online questionnaire could be prone to unintentional errors. We controlled this treat with a post-experimental cross-check of paper and online records.
Experiment mortality	All subjects finished the experiment according to the plan.
Diffusion of treatment	Controlled with the interdiction of verbal communication within the experiment and supervision of the e-communication.
Compensatory behavior	Irrelevant – all the experiment's subjects and modeling teams were treated equally.
Experimenter's expectancy	Experiment was supervised by assistants who were unaware about the stated hypotheses.

In respect to the external validity, the highest threats were to the DMT and CMT used in the experiment (Section 4.1). Whilst only two modeling tools were investigated, there is a risk of generalizing the tool characteristics results (PTC) to all modeling tools. In light of productivity, which represented the focal observation of our research, the threat of generalizing the results was minimized by defining equal criteria for the selection of both types of modeling tools (see Section 4.1), and limiting the used functions in both tools to their intersections. For example, despite the available communication capabilities of the representative CMT, the subjects used common synchronous and asynchronous communication tools in both cases. The similarities between the representative tools as perceived by experimental subject were also demonstrated using the *t*-test, which indicated no significant differences between the means of the total PTCs of both representative tools. Nevertheless, there were no significant differences found in the light of performed corrections during individual activities, which were performed equally for both types of modeling tools.

Second external validity threat could refer to the size of the defined reference BPMN model (Fig. 2), where the investigated modeling tools could behave differently in case of using a more complex BPMN model. We further investigated this issue by comparing the investigated tools according to modeling tools characteristics as defined by (2010). The results showed that both tools offer support for managing large BPMN models, such as: birds-eye view, automatic layout and zooming. Third external validity threat could refer to the selected communication tools and their impact on the modelers' productivity. Since CMT usually provide a built-in collaborative environment (Xu 2012), we presume that CMT could perform even better if our experimental investigation would use the integrated communication tools. Fourth, our investigation involved undergraduate IT students, who have grown up with web technologies. Considering this, we believe that an older population of modelers could perform inferior to students in case of using CMT, since they are more familiar with traditional, desktop-based solutions.

6.2 Implications and Future Work

We believe that the most important implication of this research is its attempts to quantitatively investigate the differences between end-users' productivity when using desktop and cloud-based solutions. Whilst the related research focused mainly on qualitative-based analysis of the security, economic, and legal aspects of desktop and cloud-based solutions, our research is complementary because it investigated quantitative aspects by performing an experimental investigation. Since there is a concern shared by some researchers about the lack of empirical research and experimentation in software engineering (Dyba et al. 2005; Kitchenham et al. 2002), especially in the domain of cloud computing (Chebrolu 2012; Opitz et al. 2012; Alharbi 2012), we believe that our research might also stimulate other researchers to perform analogue investigations in other software domains which apply two currently most common software architectures—desktop and cloud-based.

Besides the theoretical implications, we can see also the practical ones. Modeling tools' providers could use our experimental results and their relationships with the related work (Table 13) for defining their strategies when evolving their tools. For example, a summarized finding of our investigation is that CMT characteristics are already in line with those of DMT (Table 6), where the productivity of using CMT in the cases of individual and collaborative work outperforms the productivity of using DMT. These findings, combined with increased collaborating and networking capabilities of CMT (Xu 2012), could motivate providers to shift their modeling tools from desktop to cloud architectures.

On the other hand, our research could be useful for modelers, which are commonly investigating new approaches and tools in order to increase their productivity. In general, we demonstrated that

CMT can compete with DMT during individual activities and outperform it during collaborative ones. This means that, in light of productivity, modelers should consider to support their work with CMT.

Besides these recommendations, readers are advised to consider DMT and CMT in a holistic way—outside the scope of our research. For example, Benlian and Hess (2011) identified following weaknesses and threats relating to the use of CMT: (1) any problems relating to the non-availability of a CMT would interrupt modeling activities, (2) whilst companies treat their artifacts as an intellectual property, they might have security and privacy concerns about putting their diagrams into the "cloud", (3) the interchange of artifacts between cloud and desktop tools is still hard to implement and use (e.g. copy-paste between a cloud and a desktop tool is commonly unsupported), (4) users of CMT usually lack control over the upgrades, which might cause interoperability and compatibility problems and (5) if a provider vendor stops maintaining a CMT, users can lose the access to the tool as well as the related artifacts (diagrams) stored in the cloud. However, even though security and privacy are potential threats when trying to promote greater user acceptance of SaaS, Du et al. (2013) suggested that improving ease of use, reliability and responsiveness is more crucial than to improve security. This is in accordance with Wu's (2011a). As author noted, many cloud service surveys reported that security and trust are the primary concern when adopting cloud services, yet few studies have systematically explored such issues. The results of the author's study demonstrated that users were neutral when asked whether the security of data backups is determinant of using the SaaS solutions. Besides these weaknesses and threats, the quality characteristics of desktop applications and SaaS, as discussed in Table 1, should also be considered along with the findings of the related work (see Section 3).

We believe in the importance of our research investigation's approach and domain, so we will continue our work in two main directions. Firstly, we plan to increase the validity of the research addressing and minimizing the threats present during this investigation. In this manner, we plan to replace paper-based experimental instruments and those requiring manual recordings of observations, with automatic ones. For example, we plan to automatically record tasks' times and supervise information flows within communication and cooperation activities. Secondly, we plan to perform triangulation research, which will increase the cumulative validity of the results. This means that our future investigations will include: (1) other modeling domains or modeling tools (e.g. UML tools), (2) other experimental tasks and scenarios and (3) other experimental participants (e.g. IT professionals).

7 Conclusion

In this article we performed a quantitative investigation of modelers' productivity when using desktop and cloud modeling tools in the cases of individual work and e-collaboration. The experiment included two popular representatives of both types of modeling tools within the BPMN domain. The main results from the experimental activities performed by 129 IT students are as follows. Regarding the defined tools' characteristics, we found no significant differences between the aggregated mean values of the investigated desktop and cloud modeling tools, meaning that the participants perceived both types of tools similarly. However, an analysis of individual tool characteristics showed that, on the one hand, the subjects slightly preferred the desktop tool's user interface, but on the other hand, they preferred the cloud modeling tool's modeling capabilities. In respect to individual work, we found that the subjects finished individual activities faster and made less corrections to the diagrams when using cloud modeling tool, however with no significant improvements against the desktop one. In respect to e-collaborative work, we demonstrated that the subjects finished collaborative activities significantly faster and made significantly less corrections to the diagrams when using cloud modeling tool. While both productivity indicators correlated, it can be presumed that the modelers also performed their collaborative activities faster due to less repetition of work when using the cloud modeling tool.

If we aggregate all the partial results, we can argue that cloud modeling tools are comparable with desktop modeling tools during individual activities and outperform them during e-collaborative ones. These findings correlate with some of the related research, stating that with the use of state-of-the-art Web technologies, cloud-based applications can achieve the user experience and performance of desktop applications. Nevertheless, readers should apply our findings only by considering the qualities of desktop and cloud modeling tools, which were not investigated by us (e.g. the capability of performing offline work, interoperability between tools and privacy concerns) and with the limitations of the research in mind.

References

Alharbi ST (2012) Users' acceptance of cloud computing in Saudi Arabia: an extension of technology acceptance model. Int J Cloud Appl Comput 2:1–11. doi:10.4018/ijcac.2012040101

Basili VR, Caldiera G, Rombach HD (1994) Goal question metric paradigm. Encyclopedia of software engineering. Wiley-Interscience, pp 528–532

Benlian A, Hess T (2011) Opportunities and risks of software-as-a-service: findings from a survey of IT executives. Decis Support Syst 52:232–246. doi:10.1016/j.dss.2011.07.007

Bibi S, Katsaros D, Bozanis P (2012) Business application acquisition: on-premise or SaaS-based solutions? IEEE Softw 29:86–93. doi:10.1109/MS.2011.119

Bland JM, Altman DG (1995) Multiple significance tests: the Bonferroni method. BMJ 310:170

Briand LC, Differding CM, Rombach HD (1996) Practical guidelines for measurement-based process improvement

Chebrolu SB (2012) How do cloud capabilities impact various aspects of IT effectiveness? 2012 I.E. 5th International Conference on Cloud Computing (CLOUD). pp 932–940

Chieu TC, Mohindra A, Karve AA, Segal A (2009) Dynamic scaling of web applications in a virtualized cloud computing environment. IEEE 281–286

Chinosi M, Trombetta A (2011) BPMN: an introduction to the standard. Comput Stand Interfaces. doi:10.1016/j.csi.2011.06.002

Clason D, Dormody T (1994) Analyzing data measured by individual Likert-type items. J Agric Educ 35:31–35

Dale V (2012) Usability for desktop apps. http://help.utest.com/testers/crash-courses/usability/usability-for-desktop-apps. Accessed 25 Jul 2012

De Valck K, van Bruggen GH, Wierenga B (2009) Virtual communities: a marketing perspective. Decis Support Syst 47:185–203. doi:10.1016/j.dss.2009.02.008

Desisto RP, Pring B (2011) Essential SaaS overview and 2011 guide to SaaS research. 15

Dillon T, Wu C, Chang E (2010) Cloud computing: issues and challenges. IEEE 27–33

Dixon W, Yuen K (1974) Trimming and winsorization: a review. Stat Pap 15:157–170. doi:10.1007/BF02922904

Du J, Lu J, Wu D et al (2013) User acceptance of software as a service: evidence from customers of China's leading e-commerce company Alibaba. J Syst Softw. doi:10.1016/j.jss.2013.03.012

Dyba T, Kitchenham BA, Jorgensen M (2005) Evidence-based software engineering for practitioners. IEEE Softw 22:58–65. doi:10.1109/MS.2005.6

Ellis C, Wainer J (1994) A Conceptual model of groupware. Proceedings of the 1994 ACM conference on Computer supported cooperative work. pp 79–88

Ellis CA, Gibbs SJ, Rein G (1991) Groupware: some issues and experiences. Commun ACM 34:39–58. doi:10.1145/99977.99987

Ferreira A, Antunes P (2007) A technique for evaluating shared workspaces efficiency. Proceedings of the 10th international conference on Computer supported cooperative work in design III. Springer, Berlin, pp 82–91

Fuks H, Raposo AB, Gerosa MA, Lucena CJP (2005) Applying the 3C model to groupware development. Int J Cooperative Inf Syst 299–328

García-Magariño I, Fuentes-Fernández R, Gómez-Sanz JJ (2010) A framework for the definition of metamodels for computer-aided software engineering tools. Inf Softw Technol 52:422–435. doi:10.1016/j.infsof.2009.10.008

Gerosa MA, Fuks H, Lucena C (2003) Analysis and design of awareness elements in collaborative digital environments: a case study in the AulaNet learning environment. J Interact Learn Res 14:315–332

Godse M, Mulik S (2009) An approach for selecting Software-as-a-Service (SaaS) product. IEEE 155–158

Holden RJ, Karsh B-T (2010) The technology acceptance model: its past and its future in health care. J Biomed Inform 43:159–172. doi:10.1016/j.jbi.2009.07.002

Holzinger A, Mayr S, Slany W, Debevc M (2010) The influence of AJAX on Web usability. e-Business (ICE-B), Proceedings of the 2010 International Conference on. Athens, pp 124–127

Iosup A, Ostermann S, Yigitbasi MN et al (2011) Performance analysis of cloud computing services for many-tasks scientific computing. IEEE Trans Parallel and Distrib Syst 22:931–945. doi:10.1109/TPDS.2011.66

ISO 9126–4 (2004) ISO/IEC TR 9126-4-Software engineering, product quality, quality in use metrics. International Organization for Standardization

Ju J, Wang Y, Fu J et al (2010) Research on key technology in SaaS. IEEE 384–387

Katzmarzik A (2011) Product differentiation for Software-as-a-Service providers. Bus Inf Syst Eng 3:19–31. doi:10.1007/s12599-010-0142-4

Keller A, Hüsig S (2009) Ex ante identification of disruptive innovations in the software industry applied to web applications: the case of Microsoft's vs. Google's office applications. Technol Forecast Soc Chang 76:1044–1054. doi:10.1016/j.techfore.2009.03.005

Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. 65

Kitchenham BA, Pfleeger SL, Pickard LM et al (2002) Preliminary guidelines for empirical research in software engineering. IEEE Trans Softw Eng 28:721–734. doi:10.1109/TSE.2002.1027796

Kuhn DL (1989) Selecting and effectively using a computer aided software engineering tool. Annual Westinghouse computer symposium, Pittsburgh, PA (USA)

Legris P, Ingham J, Collerette P (2003) Why do people use information technology? A critical review of the technology acceptance model. Inf Manag 40:191–204

Mader S (2007) Wikipatterns, 1st ed. Wiley

Mamčenko J (2004) Introduction to lotus notes collaborative software

Mann J (2011) Cloud computing means new opportunities and decisions for collaboration. 4

Marston S, Li Z, Bandyopadhyay S et al (2011) Cloud computing—the business perspective. Decis Support Syst 51:176–189. doi:10.1016/j.dss.2010.12.006

Martin WE, Bridgmon KD (2012) Quantitative and statistical research methods: from hypothesis to results, 1st ed. Jossey-Bass

Melao N, Pidd M (2000) A conceptual framework for understanding business processes and business process modelling. Inform Syst J 10:105–129. doi:10.1046/j.1365-2575.2000.00075.x

Mell P, Grance T (2011) The NIST definition of cloud computing (Draft). 7

Mili H, Tremblay G, Jaoude GB et al (2010) Business process modeling languages: sorting through the alphabet soup. ACM Comput Surv 43:4:1–4:56. doi:10.1145/1824795.1824799

Min D, Koo S, Chung Y-H, Kim B (1999) Distributed GOMS: an extension of GOMS to group task. 1999 I.E. International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. IEEE, pp 720–725 vol.5

Mohagheghi P (2010) An approach for empirical evaluation of model-driven engineering in multiple dimensions—SINTEF. Paris, France, pp 6–17

Neuman WL (2005) Social research methods: qualitative and quantitative approaches, 5th ed. Allyn & Bacon

Opitz N, Langkau TF, Schmidt NH, Kolbe LM (2012) Technology acceptance of cloud computing: empirical evidence from German IT Departments. 2012 45th Hawaii International Conference on System Science (HICSS). pp 1593–1602

Oz E (2005) Information technology productivity: in search of a definite observation. Inf Manag 42:789–798. doi:10.1016/j.im.2004.08.003

Patel H, Pettitt M, Wilson JR (2012) Factors of collaborative working: a framework for a collaboration model. Appl Ergon 43:1–26. doi:10.1016/j.apergo.2011.04.009

Quinn LS (2010) Comparing online vs. traditional office software. In: TechSoup. http://www.techsoup.org/learningcenter/software/page11852.cfm. Accessed 27 Jul 2012

Roca JC, Chiu C-M, Martínez FJ (2006) Understanding e-learning continuance intention: an extension of the technology acceptance model. Int J Hum Comput Stud 64:683–696. doi:10.1016/j.ijhcs.2006.01.003

Schmidt K (1991) Riding a tiger, or computer supported cooperative work. ECSCW'91 Proceedings of The Second European Conference on Computer-Supported Cooperative Work 1–16

Schmietendorf A (2008) Assessment of business process modeling tools under consideration of business process management activities. IWSM/Metrikon/Mensura. pp 141–154

Schuman S (2006) Creating a culture of collaboration: The International Association of Facilitators Handbook, 1st ed. Jossey-Bass

Serçe FC, Swigger K, Alpaslan FN et al (2011) Online collaboration: collaborative behavior patterns and factors affecting globally distributed team performance. Comput Hum Behav 27:490–503. doi:10.1016/j.chb.2010.09.017

Shapiro R, White SA, Bock C, et al. (2011) BPMN 2.0 handbook second edition: methods, concepts, case studies and standards in business process modeling notation. Future strategies, incorporated

Sheriff PD (2002) Designing for web or desktop? In: http://msdn.microsoft.com. http://msdn.microsoft.com/en-us/library/ms973831.aspx. Accessed 29 Mar 2012

SPRING Singapore (2011) A guide to productivity measurement. SPRING Singapore, Singapore

Sun A (2013) Enabling collaborative decision-making in watershed management using cloud-computing services. Environ Model Softw 41:93–97. doi:10.1016/j.envsoft.2012.11.008

Thong JYL, Hong W, Tam K-Y (2002) Understanding user acceptance of digital libraries: what are the roles of interface characteristics, organizational context, and individual differences? Int J Hum Comput Stud 57:215–242. doi:10.1016/S1071-5819(02)91024-4

Torchiano M, Ricca F, Marchetto A (2010) Are web applications more defect-prone than desktop applications? Int J Softw Tools Technol Transfer 13:151–166. doi:10.1007/s10009-010-0182-6

Trochim W, Donnelly JP (2006) The research methods knowledge base, 3rd ed. Atomic dog

van Ommeren E, Duivestein S, deVadoss J et al (2009) Collaboration in the cloud: how cross-boundary collaboration is transforming business. Microsoft and Sogeti, Groningen

VMware (2009) Solving the desktop dilemma with user-centric desktop virtualization for the enterprise. 6

Wang Y (2006) E-collaboration—a literature analysis. Intelligent production machines and systems. Elsevier, pp 132–137

Wang L, Tao J, Kunze M, et al. (2008) Scientific cloud computing: early definition and experience. IEEE 825–830

Winter JCF, Dodou D (2010) Five-point Likert items: t test versus Mann–Whitney-Wilcoxon. Pract Assess Res Eval 15:16

Wu W-W (2011a) Mining significant factors affecting the adoption of SaaS using the rough set approach. J Syst Softw 84:435–441. doi:10.1016/j.jss.2010.11.890

Wu W-W (2011b) Developing an explorative model for SaaS adoption. Expert Syst Appl 38:15057–15064. doi:10.1016/j.eswa.2011.05.039

Xu X (2012) From cloud computing to cloud manufacturing. Robot Comput Integr Manuf 28:75–86. doi:10.1016/j.rcim.2011.07.002

Yan Z, Reijers HA, Dijkman RM (2010) An evaluation of BPMN modeling tools. In: Mendling J, Weidlich M, Weske M (eds) Business process modeling notation. Springer, Berlin, pp 121–128

Yang S, Yoo B, Jahng J (2010) Does the SaaS model really increase customer benefits? Asia Pac J Inf Syst 20:87–101

Gregor Polančič is an assistant professor in Computer Science & Informatics at University of Maribor, Slovenia. He received his Ph.D. in Computer Science & Informatics from the same university in 2008. His main research interests include empirical investigations of Business Process Modeling and Management, IT management, IT acceptance, e-communication and e-collaboration. He has appeared as an author or co-author in more than 15 peer-reviewed scientific journals. In all, his bibliography contains more than 170 records.

Gregor Jošt received BSc in Computer Science & Informatics from the University of Maribor, Slovenia in 2010. Currently he is a Ph.D. student and a teaching assistant at the University of Maribor. His main research interests include empirical investigations of Process Modeling and Automation, Cloud Computing, Web technologies and Mobile solutions. He has participated in several research and applied projects and appeared as an author or co-author in several scientific and conference papers.

Marjan Heričko received his PhD in Computer Science & Informatics from the University of Maribor in 1998. He is a full professor at the University of Maribor, as well as the deputy head of the Institute of Informatics and the head of the Laboratory of Information Systems. His main research interests include all aspects of information systems development with an emphasis on Software and Service Engineering, Agile methods, Mobile solutions, advanced software architectures and patterns, Knowledge Management and Semantic Technologies, Distributed Object Models Performance and Efficiency. He has published more than 300 research papers and participated in numerous research and applied projects.