Design of Interactional End-to-End Web Applications for Smart Cities

Erich Ortner
TECHNUM
Constance, Germany
e.ortner@technum.biz

Marco Mevius
HTWG Konstanz
Constance, Germany
mmevius@htwg-konstanz.de

Peter Wiedmann
AXON IVY AG
Munich, Germany
peter.wiedmann@axonivy.com

Florian Kurz
HTWG Konstanz
Constance, Germany
fkurz@htwg-konstanz.de

ABSTRACT
Nowadays, the number of flexible and fast human to application system interactions is dramatically increasing. For instance, citizens interact with the help of the internet to organize surveys or meetings (in real-time) spontaneously. These interactions are supported by technologies and application systems such as free wireless networks, web -or mobile apps. Smart Cities aim at enabling their citizens to use these digital services, e.g., by providing enhanced networks and application infrastructures maintained by the public administration. However, looking beyond technology, there is still a significant lack of interaction and support between "normal" citizens and the public administration. For instance, democratic decision processes (e.g. how to allocate public disposable budgets) are often discussed by the public administration without citizen involvement. This paper introduces an approach, which describes the design of enhanced interactional web applications for Smart Cities based on dialogical logic process patterns. We demonstrate the approach with the help of a budgeting scenario as well as a summary and outlook on further research.

Categories and Subject Descriptors
D.2.11 [**Software Engineering**]: Software Architectures – *patterns.*

General Terms
Documentation, Design, Human Factors, Standardization, Languages

Keywords
Interaction, Application System, Dialogical Logic, BPMN, Smart City

1. INTRODUCTION
The term "Smart Cities" has been variously defined in the literature [1–3]. These definitions had a common aspect of Smart Cities, which is the integration of information technology (IT) into the "daily life" of public administration and citizen interaction. For this purpose, Smart Cities aim at enabling their citizens to use these digital services, e.g., by providing enhanced web applications and network infrastructures maintained by the public administration (e.g. "Code of America" [4], "Free Wireless Hotspots Munich [5]). Furthermore, Smart Cities provide different options of collaboration, e.g., in form of electronic governance (e-governance) to improve internal and external functioning [6]. This leads to a closer interaction between the public administration and the citizens [7]. For instance, decisions (e.g. how to use disposable budget) can be made more transparent and fair. Application systems provide a wide set of functionalities supporting the interaction processes of all participants. In fact, several web portals and mobile applications have been developed during the last period in the context of e-government. However, looking beyond technology, a significant lack of interaction and support between "normal" citizens and the public administration still exists. Often, the development of application systems is just focused on the software and hardware implementation [8]. Moreover, even if application systems have been used to support interaction processes, the decision-making by consensus have not been satisfying.

This paper describes an approach for using dialogical logic to develop enhanced interactional application systems for Smart Cities. Dialogical logic can enhance the interaction between all participants, i.e. citizens and public administration. Especially, in areas with critical human work or high collaboration a detailed investigation of interaction is required to extract a fair and transparent result (cf. [9]). Dialogical logic is based on the operative approach to constructive logic [10, 11]. In "dialogical games" two interaction partners compete with each other. For winning a dialog, the proponent (interaction partner who started the dialog) must not absolutely know if the elementary propositions are true or not [12]. This option – having a "non liquet" status – can be used in the context of Business Process Management (BPM) too. BPM defines the management of business processes using methods to design, enact, control and analyze business processes [13]. Therefore "non liquet" does not define an additional value. The term defines only that there is no definite status decided yet. This constructive approach allows more creativity [14] within interaction process activities and leads to an understandable result. In other words, the development from a dialogical logic point of view is the next step toward interactional application system. For this purpose, the paper is structured as follows: in section 2 related work is described, especially the different approaches of design pattern are discussed. Section 3 introduces the terms of dialogical logic and introduces a more detailed investigation of interaction patterns. The fourth section presents a budgeting process modeled with the help of the modeling language BPMN^{Easy}[15]. The evaluation of the described scenario indicates the positive effect of dialogical logic-based dialogs based on interactional application systems. The last section concludes the approach and introduces an outlook on further research.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2743906

2. RELATED WORK

The literature provides several overviews about application systems, modeling pattern, modeling languages and techniques, with consideration of visualization and execution of interaction processes [16–19].

Software systems and their artifacts can be visualized, constructed and documented with the Unified Modeling Language (UML) [20]. Different standard model types in UML are used to capture all relevant information. UML can be used to generate a common understanding about e.g., design, configuration and maintenance. For instance, an "activity diagram" or "interaction overview diagram" includes a sequence of activities in (business) processes. Especially in agile environments there is a great demand for visual models. Unfortunately, there is no possibility to model the interaction procedure itself.

In the context of (business) process management [21], workflow technologies, e.g. to automate a budgeting process, support the handling of interactions. Many equal business requirements can be found in different business processes. Workflow patterns are a general and abstract solution for these business requirements in the view of the business process execution. Riehle and Züllighoven [22] described pattern in general as " the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts". The workflow pattern have to be independent from the workflow implementation language and the underlying workflow-system [23]. Therefore, a large set of workflow pattern have been developed (see e.g. [23–25]). But a rule based interaction like the dialogical logic is not sufficiently taken into account.

McCowan et al. [26], described a mathematical view of interaction modeling. They introduced some basic interaction situation pattern generated from statistical data of meeting recordings. The interaction patterns introduced were used to analyze meetings and identify important information from them. In addition, to monologue and discussion interaction, the set of pattern included consensus and disagreement situations. The interaction pattern were modeled in mathematical Hidden Markov Models (HMM). Thus, a transformation in a business process model is difficult and the set of patterns do not include logic-based approaches such as the dialogical logic.

Since 2013, the International Organization for Standardization (ISO) provide a standard for business process modeling – the Business Process Model and Notation (BPMN) [27]. BPMN is supposed to be a modeling language which is understandable by all participants, such as business users or information technology (IT) experts. In addition to the business processes (business process diagram) the BPMN also provides a notation for conversations (conversation diagram). However, conversation diagrams cannot be executed by workflow systems [28] and business diagrams do not describe interaction in detail.

An approach which is driven by interaction situation in the context of knowledge work is the Human Interaction Management (HIM) [29]. HIM extends the Role Activity Diagramming (RAD) with understanding and integration of interactions of all participants. Links between specific activities and user roles offer more detailed perspectives for a user role concept. The model complexity of the so called human-driven processes is a challenge for the maintenance of such RAD models.

Furthermore, an end-to-end perspective is provided by Ortner et al. [30]. From their point of view, pure technique-focused requirements lose. According to this end-to-end perspective, the Figure 1 shows three layers (communities, applications, devices) of an end-to-end application architecture. Supported by the technologies (c.f. Fig. 1 applications and devices layer), all "stakeholders" (e.g. citizens) can interact and synchronize themselves regarding knowledge and the interaction context end-to-end.

Communities	**User**	Smart City Citizens
	Interaction-Management	Workflow-Management-System
Applications	**Application-Management**	Application-Management-System
	Data-Management	Database-Management-System
Devices	**Network-Management**	Network-Management-System
	Internet of Everything	Network-Protocol

Figure 1: Architecture to support end-to-end applications

Using web applications, which cover such an end-to-end approach, citizens and the public administration, are enabled to share their creativity or expertise. Another example is known as BPM Touch [31]. BPM Touch specifies a mobile application, which offers an innovative way of capturing interaction processes with tablets end-to-end. Hereby, light-weighted and multimedia-based features lead to a very intuitive handling of this tool. Especially in agile environments or tight situations BPM Touch supports a fast way of collecting all needed information. For instance, descriptions of interactions such as how two participants interact within an activity can be recorded as video or audio file.

However, business processes modeled by existing approaches provide only minor value in terms of holistically capturing the interaction itself. For instance, even if interaction details, such as interaction time or quality information are stored in models, it is "difficult" to analyze their negotiation. Moreover, the existing approaches take patterns, which aiming at a fair and transparent interaction, not into account sufficiently. This paper discusses an approach focusing on closing this gap, by adapting and extending the existing approaches.

3. INTERACTION CAPTURING

3.1 Dialogical Logic

Dialogical logic has been introduced by Lorenzen. In a dialog two parties (e.g. two interacting humans) argue about an assertion respecting certain fixed rules [39]. This dialog has two roles: the "defender" of the thesis called proponent and the "attacker" of the thesis called opponent. Each dialog ends after a finite number of moves with the winning of one player and the loss of the other. The rules are divided into structural rules and particle rules. The structural rules define the general course of a dialog game. The particle rules describe which moves are allowed to attack or defend. Lorenz summarized the structural rules as follows [39]:

1) Structural rules

a. (start): The proponent starts the dialog by providing an initial statement. Each move following the initial statement is considered either an attack or a defense.

b. (delay): No delaying tactics are allowed. This means only moves which change the dialog situation are allowed.

c. (formal): Only elementary statements, which have been claimed by the opponent, can be claimed by the proponent.

Statements which do not contain other statements as part of them are called elementary.

d. (winning): The proponent or opponent wins if it is his turn and cannot move (i.e. neither attack nor defend).

e. (intuitionistic): In any move, each player may attack the (complex) formula asserted by the rival. Otherwise the player may defend himself against the last attack that has not yet been answered.

(Note: As an alternative rule to the intuitionistic the "classical rule" could be chosen.)

2) Particle rules

Particle rules describe which attacks or defenses within a dialog are allowed. Figure 2 lists the formal description of the operations.

	¬, , ∨, →, ∀, ∃	Attack	Defense
1	¬A	A	(no defense)
2	A → B	A	B
3	A ∧ B	?L(eft) ?R(ight)	A B
4	A ∨ B	?	A B
5	$\forall_x A$	$?_n$	A[n/x]
6	$\exists_x A$	?	A[n/x]

Figure 2: Particle rules of dialogical logic-based on [39].

The particle rules can be seen as an abstract argumentation form, which has no reference to the context of argumentation in which the rules are used [24].

The negation (cf. row 1, Figure 2) negates the statement simply. Hereby, only a counterattack is possible and no defense. The material implication (= "Subjunction", cf. row 2, Figure 2) defines "if A is true, then B is also true". The statement is only not true in case A is true and B is false. The difference between disjunction and conjunction (cf. row 3 & 4, Figure 2) lies in the hands of the player with the immediate next subformula. Moreover, Mevius and Wiedmann [24] described that "in a conjunction, the challenger may choose, confident that either disjunct can be refuted; in a disjunction the choice lies with the defender. Thus, to defend a conjunction, a player must be able to defend any of the conjuncts, while in the case of a disjunction, it is sufficient to be able to defend one of the disjuncts. For the quantifiers (cf. row 5 & 6, Figure 2): when attacking a universal quantifier, the challenger can choose the instantiation he fancies for the bound variable"If the statement is existentially quantified, the defender then picks the instantiation [24].

3.2 Pattern overview

The definition of patterns differs in various literature. According to Coad et al. [32], patterns are "a fully realized form, original, or model [..] for imitation"**.** In contrast to this definition Alexander and Ishikawa et al. [33] define "Each pattern is a three part rule, which expresses a relation between a certain context, a problem, and a solution". Focused on process modeling patterns, we use the definition from Riehle and Züllighoven [22], "A pattern is the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts. A large set of workflow patterns have been developed and described in the literature (see e.g. [23–25]). A popular collection of workflow patterns were published by van der Aalst and Hofstede et al. [23]. The collection contains 20 basic patterns subdivided into six sections. In addition to sequence-flow patterns, branching patterns and parallelization patterns, there are treasury patterns for structure, multiple instances and abort situations (Table 1) [23].

Table 1: Workflow-Pattern according to van der Aalst [23]

Section	#	***Pattern***
Basic Control Flow Patterns	1	Sequence
	2	Parallel Split
	3	Synchronization
	4	Exclusive Choice
	5	Simple Merge
Advanced Branching and Synchronization Patterns	6	Multi-choice
	7	Synchronizing Merge
	8	Multi-merge
	9	Discriminator
Structural Patterns	10	Arbitrary Cycles
	11	Implicit Termination
Patterns involving Multiple Instances	12	Multiple Instances Without Synchronization
	13	Multiple Instances With a Priori Design Time Knowledge
	14	Multiple Instances With a Priori Runtime Knowledge
	15	Multiple Instances Without a Priori Runtime Knowledge
State-based Patterns	16	Deferred Choice
	17	Interleaved Parallel Routing
	18	Milestone
Cancellation Patterns	19	Cancel Activity
	20	Cancel Case

Additional workflow patterns include special patterns for loop- and cycle-situations. Such situations can be found in the most business processes so there is a big need to model them. [34]

As a rule-based interaction situation, the dialogical logic describes flows in which the interactions have to be executed. The aim of our dialogical logic-based interaction pattern is an execution without dependencies of a concrete workflow system. To achieve this aim the pattern for the dialogical logic-based interaction has to be built exclusively based on a known workflow pattern which will be executed by current workflow systems. The basic workflow patterns, which will be used to create a dialogical logic-based interaction pattern with BPMN 2.0, are introduced in a further section.

One example of the basic workflow patterns is the Structured Loop Pattern. If a repeated execution of a part of the process is required, a Structured Loop Pattern could be used. Therefore, there is either a pre-test or a post-test condition according to which the test will be evaluated either at the beginning or at the end of the loop.

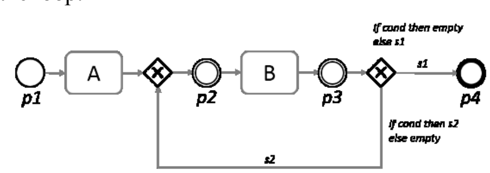

Figure 3: Structured Loop Pattern in the BPMN^{Easy} notation

The example shown in Figure 3 illustrates a Structured Loop Pattern with a post-test condition modeled in the BPMN^{Easy} notation.

3.3 Dialogical Logic Pattern for Interaction Process Execution

The described dialogical logic is used to outline the implementation of an interaction pattern. With regard to an execution of concrete implementations of this pattern, we illustrated the interaction pattern in a BPMN^{Easy} process model. The notation includes activities (manual, user, service and generic), events (start, intermediate and end) and gateways (exclusive and parallel). In addition, all elements can be enriched by multimedia data, e.g., video -or audio sequences. Every BPMN^{Easy} process model is persisted in the BPMN-XML format. This allows the import of each interaction pattern in a BPMN-based workflow-system [15, 35].

According to Riehle and Zullighoven [22] and to the well-known workflow patterns of van der Aalst & Hofstede [23], patterns need to include a description, examples, identified problem and an implementation. Based on these requirements the Dialogical Logic Pattern is defined as follows:

Description

A logical truth based interaction of two interaction partners which is executed in the framework of dialogical logic.

Example

Two dialog partners discuss a change request in a knowledge database. At the beginning of the interaction, the proponent exposes his topic why the entry should be changed. During the dialog the opponent investigates all sub-statements. Upon an attack from the opponent the proponent have to prove his sub-statements. The interaction process will be ended if all sub-statements are proved or one of them is detected as wrong. . Hereby, the general idea delivers promising advantages to current and future Smart Cities, e.g., for organizing a fair and transparent interaction and capturing all (dialog-) relevant data.

Problem

It has to be ensured, that during the dialog all particle rules are applied. Each sub-statement has to be proven until a dialog can be won.

Implementation

The Dialogical Logic Pattern represents the modeled rules from the particle rules of the dialogical logic as described in the previous section. In preparation for executing the process model every process flow composited from the previously described workflow patterns was implemented. So there is a strict separation of exclusive choice, simple merge and structured loop gateways.

The concept of dialogical logic is based on the logical truth of a statement. Neither a consensus nor an eloquent argument can decide such a dialog. Therefore, it is indispensable that both participants are familiar with the rules. To solve a possible informational imbalance, each task and gateway is annotated with multimedia files. Furthermore, patterns enable "new" participants to understand the underlying theory of dialogical logic. A basic form of the dialogical logic-based interaction process modeled in BPMN^{Easy} is illustrated in Figure 4.

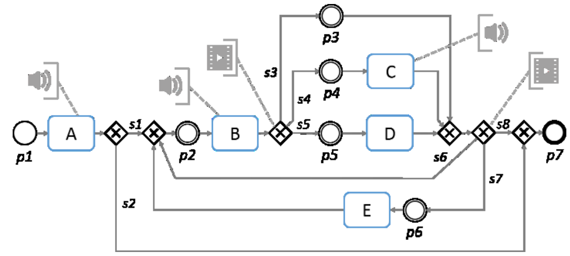

Figure 4: Dialogical Logic Pattern

In order to simplify the process flow the process model is supplemented with a textual description for each modeled notation element. The used events are labeled as follows:

- p1. Dialog started
- p2. Doubts present
- p3. No counterattack possible
- p4. Counterattack possible
- p5. Attack refutable
- p6. Sub-statement defended
- p7. Dialog ended

In agreement with the BPMN^{Easy} notation tasks types differs in the color of their frame. The textual description of the tasks is:

- A. Formulate dialog topic
- B. Attack sub-statement
- C. Defend sub-statement (Refute attack)
- D. Counterattack sub-statement
- E. Choose next sub-statement

For reasons of clarity and ease of navigation, all sequences were also labeled:

- s1. Sub-statement is challenged
- s2. Sub-statement is not challenged
- s3. Attack not defendable
- s4. Attack refutable
- s5. Counterattack possible
- s6. Attack possible
- s7. Sub-statement defended AND more sub-statements present
- s8. No further sub-statements present

4. EVALUATION

Since more than three years different research projects in cooperation between the Constance Institute of Process Control (kips) and the City of Constance target the context of a Smart City Constance and the improvement of citizen participation in various administration processes. The introduced approach has been applied during a research project at the Constance Institute of Process Control (kips). Therefore, we have selected specific tools to set up the required application architecture. Afterwards, we have captured the chosen interaction process using BPMN^{Easy} and the introduced dialogical logic pattern. Finally, we enriched the process model with technical details enabling the execution of the interaction process model in a workflow-system.

4.1 Scenario

A real scenario for the Smart City Constance has been chosen to apply the introduced approach of a Dialogical Logic Pattern. This scenario is concerned with the decision of allocating public budgets, allowing the influence of every participant. A Participatory Budgeting Project based on this consideration is available since 2005. The mission of the project is determined as follows: "(...) empower people to decide together how to spend public money. We create and support participatory budgeting processes that deepen democracy, build stronger communities, and make public budgets more equitable and effective.“ [36]. Argyris [37] introduced major problems for budgets which are defined without any participations of employees. Argyris publicize the lack between motivation, success and participation for budget which was underpinned by a study at the Michigan State University [38]. During a budgeting process in which participation is promoted, two parties can be identified; one for a higher budget, and the other against it [39]. To guard a fair and truthful dialog of these parties there is a need to control the interaction. Our dialogical logic is suitable to support such a budgeting interaction between two parties.

The implementation is based on the dialogical logic pattern modeled in a BPMN^{Easy}. Due to the fact that the BPMN^{Easy} model conforms to the BPMN 2.0 standard notation, the dialogical logic pattern can be easily imported in a BPMN modeling application. After importing the pattern we enriched the dialogical logic process with technical attributes needed for the execution in a workflow system. For example, further information about the database in which the topic, sub-statements and the statements made were stored. Figure 5 illustrates the implemented and enriched dialogical logic pattern in the BPMN 2.0 notation.

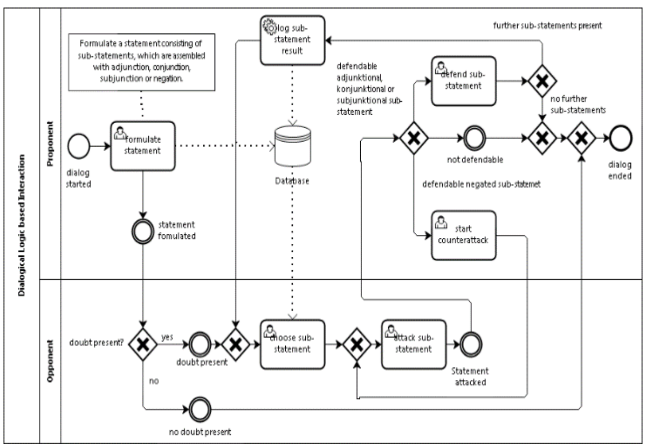

Figure 5: Dialogical Logic Pattern transformed in BPMN 2.0

Additionally, some service tasks were added for automatically logging the current state in a database. In a further step we created graphical user interfaces and linked them to the user tasks. A sample graphical user interface used for the interaction process executed in a web-based workflow system is illustrated in Figure 6.

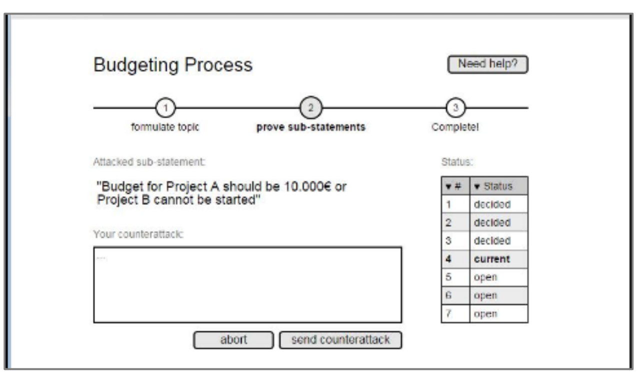

Figure 6: Sample User Interface for an End-to-End Web Application for dialogical logic-based budgeting processes

For an efficient and fair dialog the participants follow the particle rules which are modeled in the interaction process based on the Dialogical Logic Pattern. The execution of interactions is guided by a workflow-system, which adapts all dialogical logic rules strictly.

4.2 Analysis

Within the executed scenario it was demonstrated, how interactions can be assisted by application systems. For instance, dialogs have been executed systematically. With the help of human-centered workflow engine, it was able to reach understandable dialog results. Furthermore, the approach provides an end-to-end verification and validation of interaction processes, which is expected for transparent decisions within Smart Cities. An additional benefit of the dialogical logic pattern is the ability to attach multimedia files to the BPMN^{Easy} models. Moreover, information about the theory of dialogical logic and examples of this theory were attached. Thus, an easy introduction to the subject of dialogical logic is given and participants can easily understand the current state and the necessary steps during an interaction process execution very easily. While many “traditional” interactions between public administration and citizens were not comprehensible, the usage of dialogical logic examined a well understanding of the different statements needed (e.g. of arguments). However, the scenario showed the complexity of automatically analyzing the interaction dialog content (e.g. analysis of natural language).

5. CONCLUSION AND OUTLOOK

This paper introduced an approach describing the possibilities of dialogical logic-based application system development. Section 1 has provided a detailed overview about the current situation in growing Smart Cities from the perspective of the application system development. In section 2 the related work focusing on the interaction processes have been described. The scenario and evaluation in section 3 and 4 investigated, how interactions, for instance to find a fair budgeting, can be implemented. Thus, our approach enables Smart Cities to interact with their citizens with more fairness and an increased transparency supported by dialogical logic-based end-to-end web application. Next, we will investigate, how direct feedback (e.g. user experience of citizens) can be continuously captured and used to optimize interaction processes.

6. References

[1] Gaggioli, A. 2014. CyberSightings. *Cyberpsychology, Behavior, and Social Networking* 17, 1, 59.

[2] Neirotti, P., Marco, A. de, Cagliano, A. C., Mangano, G., and Scorrano, F. 2014. Current trends in Smart City initiatives: Some stylised facts. *Cities* 38, 25–36.

[3] Santis de, R., Fasano, A., Mignolli, N., and Villa, A. 2014. Smart city: fact and fiction. *Available at SSRN 2410739*, 1–19.

[4] Code for America Labs. 2015. *Citizens have the power to help their cities*. http://www.codeforamerica.org/geeks/. Accessed 20 January 2015.

[5] Stadtwerke München GmbH. 2015. *WLAN Hotspots in München*. http://www.muenchen.de/leben/wlan-hotspot.html. Accessed 20 January 2015.

[6] Boughzala, I., Janssen, M., and Assar, S. 2015. E-Government 2.0: Back to Reality, a 2.0 Application to Vet. In *Case Studies in e-Government 2.0*, I. Boughzala, M. Janssen and S. Assar, Eds. Springer International Publishing, Cham, 1–14.

[7] Ma, X. and Xi, B. 2014. Satisfaction evaluation of e-government based on structural equation model. In *2014 International Conference on Management Science and Engineering (ICMSE)*, 110–114.
[8] Janssen, M. and Estevez, E. 2013. Lean government and platform-based governance—Doing more with less. *Government Information Quarterly* 30, S1.
[9] Wegner, P. 1997. Why interaction is more powerful than algorithm. *Communication of ACM,* Vol 40, No.5, 80–91.
[10] Laurent, K. 2011. *Dialogical Logic*. http://plato.stanford.edu/archives/sum2011/entries/logic-dialogical. Accessed 20 January 2015.
[11] Lorenzen, P. 1987. *Lehrbuch der konstruktiven Wissenschaftstheorie*. J.B. Metzler, Stuttgart [Germany].
[12] Görz, G. 2005. Dialogische Logik und mathematischer Unterrichtsdiskurs. *Mathematik, Logik, Philosophie - Ideen und ihre historischen Wechselwirkungen*, 185–211.
[13] van der Aalst, W.M.P. 2004. Business Process Management: A Personal View. In *Business Process Management Journal*, Vol 10, 135–139.
[14] Fischer, M., Grollius, T., and Ortner, E. 2010. An Application of Constructive Logic in Computer Science. *Konstruktive Informatik*.
[15] Mevius, M. and Wiedmann, P. 2013. BPM(N) Easy1. 2-Gebrauchssprachliche Gestaltung IT-basierter Prozesse. In *BSOA-BCloud 2013. 8. Workshop Bewertungsaspekte service- und cloudorientierter Architekturen, 12. November 2013, Basel*, M. Hanin and A. Schmietendorf, Eds. Berliner Schriften zu modernen Integrationsarchitekturen 12. Shaker, Aachen, 31–46.
[16] Becker, J., Rosemann, M., and Uthmann, C. von. 2000. Guidelines of Business Process Modeling. In *Business Process Management*, van der Aalst, Wil, J. Desel and A. Oberweis, Eds. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 30–49.
[17] Moody, D. 2009. The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IIEEE Trans. Software Eng.* 35, 6, 756–779.
[18] Javidroozi, V., Shah, H., Amini, A., and Cole, A. 2014. Smart City as an Integrated Enterprise: A Business Process Centric Framework Addressing Challenges in Systems Integration. In *SMART 2014, The Third Internationl Conference on Smart Systems, Devices and Technologies*, 55–59.
[19] Mulligan, C. and Olsson, M. 2013. Architectural implications of smart city business models: an evolutionary perspective. *IEEE Communications Magazine* 51, 6, 80–85.
[20] Object Management Group (OMG). 2011. *Unified Modeling Language (UML) 2.4.1*. http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF. Accessed 16 December 2014.
[21] van der Aalst, W.M.P. 2001. Exterminating the Dynamic Change Bug: A Concrete Approach to Support Workflow Change. *Information Systems Frontiers* 3, 3, 297–317.
[22] Riehle, D. and Züllighoven, H. 1996. Understanding and Using Patterns in Software Development. *Theory and Practice of Object Systems* 2, 1, 3–13.
[23] van der Aalst, W.M.P., ter Hofstede, A., Kiepuszewski, B., and Barros, A. P. 2003. Workflow Pattern. *Distributed and Parallel Databases* 14, 1, 5–51.
[24] White, S. A. 2004. Process modeling notations and workflow patterns. *Workflow handbook* 2004, 265–294.
[25] Thom, L.H. and Lochpe, C. and Reichert, M. EMMSAD, BPMDS, BUSITAL, 349–358.
[26] McCowan, I., Bengio, S., Gatica-Perez, D., Lathoud, G., Monay, F., Moore, D., Wellner, P., and Bourlard, H. Modeling human interaction in meetings. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, IV.
[27] Object Management Group. 2013. *Business Process Model and Notation (BPMN) 2.0.2*. http://www.omg.org/spec/BPMN/2.0.2/PDF. Accessed 1 January 2014.
[28] Börger, E. 2012. Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. *Softw Syst Model* 11, 3, 305–318.
[29] Harrison-Broninski, K. 2010. Dealing with Human-Driven Processes. In *Handbook on Business Process Management 2*, J. Vom Brocke and M. Rosemann, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 443–461.
[30] Ortner, E., Mevius, M., and Wiedmann, P. 2015. *Enhanced Stakeholder Socialization using Common Language in Agile BPM. Living business processes models instead of rigid documentations*. (accepted paper), 1–8.
[31] 2015. *BPM Touch® - The Revolution*. http://www.bamero.de. Accessed 20 January 2015.
[32] Coad, P., North, D., and Mayfield, M. ©1997. *Object models*. *Strategies, patterns, and applications*. Yourdon Press computing series. Yourdon Press, Upper Saddle River, N.J.
[33] Alexander, C., Ishikawa, S., and Silverstein, M. 1977. A Pattern Language. *Center for Environmental Structure* 2.
[34] Russell, N., ter Hofstede, A., van der Aalst, W.M.P., and Mulyar, N. *Workflow Control-Flow Patterns: A Revised View*.
[35] Konstanzer Institut für Prozessteuerung (kips). 2015. *BPMNEasy*. http://www.bpmneasy.org. Accessed 20 January 2015.
[36] Menser, M. and Dolan, K. 2014. *The Participatory Budgeting Project (PBP)*. http://www.participatorybudgeting.org/. Accessed 20 January 2015.
[37] Argyris, C. 1953. Human problems with budgets. *Harvard business review : HBR* 31, 1, 97–110.
[38] Krishnan, R., Marinich, E., and Shields, M. D. 2012. Participative Budgeting, Psychological Contracts, and Honesty of Communication. *SSRN Journal,* 06, 1–45.
[39] Lerner, J. and Secondo, D. 2012. By the People, For the People: Participatory Budgeting from the Bottom Up in North America. *Journal of Public Deliberation* 2, 8, 1–9.