

The Collaboration Perspective on Continuous Development

Stijn Hoppenbrouwers^{1,2}

¹ HAN University of Applied Sciences, 6802 CE Arnhem, the Netherlands

² Radboud University, 6525 EC Nijmegen, the Netherlands
stijn.hoppenbrouwers@han.nl

Abstract. Continuous development, including continuous RE, requires more intensive, better organized, and better supported involvement of, and interaction with, business stakeholders than ever before. We should move from looking at (business) stakeholder involvement as an occasional activity in a design/development project to a regular activity very much embedded in and integrated with the daily, core business work of end-users and other stakeholders. There are similarities between 'requirements governance' and innovative, community-oriented approaches to 'data governance'. Building on work in collaborative conceptual modelling over the last decade, a perspective is presented in which the creation and maintenance of development artefacts (including requirements documents) is mirrored in a goal-oriented, continuous 'conversation about applications' (or rather, a set of interlinked sub-conversations) between various stakeholders. Gamification of this systematic interaction may add to making it more engaging and more lively. Organization and support of these conversations can add to lowering the threshold for business stakeholders to get on board and stay on board the continuous development cycle.

Keywords: continuous development, collaborative modelling, dialogue games

1 Larger context: Stakeholder Involvement in Continuous RE

Continuous RE, as part of Continuous Development [1], includes BizDev as well as DevOps, though the latter is currently (over)emphasized in both literature and practice. This keynote focuses mostly on the BizDev side of development.

Continuous or not, modern approaches to system development and business-IT alignment generally claim to closely involve business stakeholders (end users, managers, but also many 'secondary' stakeholders like for example lawyers, HRM officers, controllers, etc.). Usually the perspective on such stakeholder involvement is still that it is part of project-oriented 'design' activities, like eliciting requirements or

Copyright 2018 for this paper by its authors. Copying permitted for private and academic purposes.

asking them for feedback on designs and prototypes, in various project phases.

This perspective is beginning to conflict with more radical views that now emerge concerning continuous development. With ‘more radical’ I mean the vision that in principle, it should be possible to intensively and frequently and indeed *continuously* adapt running systems based on requests and other input from many sources: diverse kinds of stakeholder feedback (bug reports, irritations, complaints, suggestions, but also fresh ideas, and even positive experiences) as well as more classical sources (requirements interviews and workshops, business modeling, co-design) or more indirect information sources (monitoring of system use and user behavior, measured impact on the organization or on other systems; etc.).

I focus here on direct interaction between on the one hand, people in ‘continuous development teams’, i.e. IT stakeholders, and business stakeholders on the other. Such interaction can take many forms, and indeed we should look beyond highly useful but limited traditional forms of communication like requirements sessions and interviews, or one-way, formal change requests. In particular, the vision is to make interaction with stakeholders much more part of their everyday routine; to embed it into daily ‘core business’ work and in the regular *use of* and *communication about* applications as part of that. So what we refer to here is direct, purposeful communication, in whatever form, about work/business supporting systems. Very useful information can of course also be acquired through monitoring (automated or not) of users and of running software, but this is not our focus here.

When we speak about ‘interaction with business stakeholders’, the assumption is that this is somehow an organized, or at least explicitly recognized and supported, activity. Since it at least involves collaboration between IT stakeholders and business stakeholders, but very likely also collaboration among business stakeholders (possibly of various backgrounds) and collaboration among IT stakeholders (also of various backgrounds), it seems quite reasonable to characterize all this activity as inherently *collaborative*.

There is a parallel here with a field closely related to requirements engineering: business modelling in many shapes and forms. Relevant directions include Enterprise Modelling, Conceptual Modelling, Business Analysis, Functional Maintenance, Articulation of Work, and so

on. Model-oriented approaches in particular have since long taken the perspective that:

- Describing the business should preferably be done by the experts, i.e. the business itself, or at least conscientiously build on information directly derived from and checked by the business;
- Business descriptions can be created for various reasons, including knowledge management, process improvement, LEAN change management, risk assessment, and of course also: requirements engineering;
- It is not realistic to assume that business modelers, if they are indeed ‘from the business’, can simply use conceptualizations, languages, techniques and tools from IT practice and culture. Unfortunately, IT-oriented people often underestimate how different the perspectives, concerns, ‘mental models’, competences, goals and motivations of business people are from their own.

An important difference, however, between business modelling and straight-up RE is that RE more emphatically focuses not on the business as such, but on applications, systems and software, and what the business stakeholders expect and demand of it. In this respect, it is useful to distinguish, at least reflectively if not in practice, between ‘communication about work/organisation’ and ‘communication about applications/systems’. In RE, the two modes of communication are in fact often combined, and business stakeholders are quite capable of both – but on their own terms, and in their own terms. Co-workers and colleagues talk a lot about their work/organisation and how to improve it, and they also talk a lot about the systems they work with. The question is how we can harness this ‘grassroots’, daily communication to gather input for continuous development.

On the other hand, system development, continuous or not, calls for information that is structured in a way that makes it directly useful for developers. This is what underlies the creation, over the last decades, of many types of representation, models, and documentation (and related procedures and practices) like Use Cases, the UML, BPMN, business rules, scenarios, storyboards, persona’s, domain models, and so on. So the trick is to somehow get *from* ‘grass roots’ communication about work and about applications *to* effective, well-structured and goal-oriented information at the right level of abstraction to be useful for developers (or even as direct input for generative machinery).

A similar vision has indeed recently been voiced in the European enterprise engineering community [2,3]. Another parallel can be found in community-oriented approaches to Data Governance and even Architecture Governance: how to actively and more continuously engage a wider community in keeping documentation (in our case, requirements) *alive* and thereby *up to date*. These approaches share the conviction that active, continuous involvement of business stakeholders should be deeply embedded in the organization, to be part of its DNA instead of being an occasional, necessary distraction. However, most practitioners (perhaps more than academics) do realize that this is not easily achieved, and will only come at a serious price. Indeed, big challenges exist here.

Over the last fifteen years, as part of a worldwide (though largely uncoordinated) effort to address some of those challenges, I have focused not so much on the many types of *representation* that exist (and are sometimes even used) in enterprise modelling and requirements engineering, but on understanding and support of the *act of creating such representations*: the interactive, dialogue-like processes in which people collaboratively ‘generate’ or ‘author’ such descriptions (that indeed can be seen as *texts*). In the second part of this keynote, I will briefly outline this perspective, in the hope (and belief) that it is relevant to the future of continuous development and continuous requirements engineering.

2 Collaborative Dialogues Mirroring Representations

The core of the ‘dialogue perspective on business modelling / requirements engineering’ is that **every piece of documentation (representation) leading to the creation of a computational (or other) system is the result of a conceptualization-and-formulation oriented, collaborative dialogue between stakeholders**. The resulting documentation may be a set of formally required documents that are explicit project deliverables, but in principle they also include highly informal, undocumented doodles and coffee table conversations. This also, at the other end of the spectrum, includes programming code, which is a highly specialized form of text. Documentation and design/development representations do after all not appear out of thin air: they are the explicitly formulated, structured outcomes of people thinking, talking, and writing down their shared thoughts and creations.

Indeed, it seems that in the age of ‘agile development’, elaborate documentation (including requirements) as once advocated is now pretty much ‘out of grace’ [4]. Still, even in undocumented, strictly verbal or only whiteboard-supported SCRUM style conversations (e.g. standups), a lot of communication does still take place, and ideas are collaboratively developed.

Such conversations generally share some properties concerning their nature, purpose and outcome:

- They work towards clear, unambiguous conceptualizations that can be used as input for engineering-style design and construction
- In their gradual structuring, they make increasing use of concepts (terms, relations; notations) supporting specific perspectives of engineering activity (e.g. functional decomposition, process flow, entities and relationships, logic, machine states, etc.)
- They often include the classic ‘divergence – convergence’ pattern known from collaboration engineering [5]
- They include patterns of negotiation and joint decision making: bids (proposals), argumentation for and against, accepts and rejects of proposals, and decision mechanisms
- They include three levels of agreement, that are ‘stacked on each other’, level 3 being dependent on level 2, and level 2 on level 1 [6]:
 1. Understanding (people understand each other, even if they disagree)
 2. Consent (people agree on some description being ‘the right description of some situation’ –be it existing, or future/desired)
 3. Commitment (people agree to take action based on the agreed description, e.g. to use it in realizing a system).

My work has mostly been in ‘collaborative modelling’, and therefore is mostly based on ‘collaborative dialogues aiming to create a representation in view of some meta-model’. We could say the approach is that of ‘meta-model driven, guided conversations’. This is certainly not the only possible direction, but I will stick with it for now.

The current line of research started out as an attempt [6] to further develop the ‘modelling procedure’ (Conceptual Schema Design Proce-

ture) in an ‘information modelling’ (conceptual modelling) method called ORM, an exponent of the Fact Based Modelling approach [7]. The idea that system development involves a series of interlinked conversations was initiated shortly before in [8]. Gradually, we realized that not only do meta-models set ‘goals for modelling’ (i.e. syntax goals, setting ‘todo’s’ based on syntactic constraints and expectations) but that many other goals are also involved (agreement goals, clarification goals, creation goals, social goals, ...) [9]. Also, the insight emerged that conversations-for-modelling were not so much patterned as ‘fixed cookbook-like flows’ but as much more open-structured interactions governed by *rules*. This triggered a line of research investigating overlap between ‘method engineering’ and ‘game design’ [10], which in turn led to development of the *dialogue game* concept: dialogues literally governed by ‘rules of play’, helping to guide participants through the offering of limited ‘conversational moves’ set in context of specific situational modelling goals –yet still also mirroring meta-models and modelling concepts [11]. Further study of realistic, cognitively feasible patterning of dialogue games led to the ‘Focused Conceptualisation’ approach: the design of dialogue games based on the participants systematically and iteratively switching between short episodes of limited focus on small, well contextualized questions and formatted answers [12,13]. This was augmented by aspects of *facilitation*: human facilitation or (potentially) automated facilitation (guiding) of conversations for modelling [12,14]. Further gamification of such structured conversations was also explored [15]. Recently, it has been suggested that Fact-Based Modelling, being rooted in the foundations of natural language in combination with the basics of predicate logic, may be seen and used as a theoretical and practical basis for business modelling in general, with its many forms and languages [16]. This seems to fit nicely with the dialogue approach to modelling.

A good review of the work mentioned as part of a larger perspective on ‘elicitation of descriptions of work processes’ can be found in [17].

3 Discussion and conclusion

Let us return to the outset of this keynote: continuous development, continuous requirements engineering, and stakeholder involvement in a collaborative setting. The fundamental stance that every representation

created or used in a development activity is mirrored in a conversation that leads to it, and that a resulting representation –text– may also be input for further conversations, seems a realistic enough position to take. Contrarily, the much more limited idea of purposeful guiding and supporting of ‘conversations for modelling’ (or in the current context: ‘conversations for requirements engineering’) is still merely an explorative direction that may or may not lead to actual practical application. It is the perspective and framework for understanding that I consider the main contribution to the literature, though I keep aspiring to realizing the more practice-oriented ‘guided dialogue’ approach at a practical level.

In section 1, we mentioned the challenge of organizing, and perhaps supporting, and maybe even guiding, the many different types of conversation that can be observed or imagined in practice. To include all of these conversations in an explicit and even governed organizational effort seems highly over-ambitious and unrealistic, if even useful. However, when restricted to a subset of conversations that have clear goals, settings, participants and resulting artefacts (representations, documents), it may yet be interesting and useful to employ the *dialogue game* approach. In particular, on-line supported and communicated versions of such conversations may be recorded, stored and even monitored or analyzed as an asset complementary to the requirements and design documents they result in and/or are about. Even in informal, volatile contexts like SCRUM/Agile standup conversations, dialogue games in some form could be used to keep the eyes on the ball and help SCRUM masters keep chaos and loss of information at bay.

As voiced in Steven Alters keynote to PoEM 2017 (unfortunately unpublished), which was a response to [2], many challenges do face us if we want to pursue the vision presented in section 1. Dialogue games and the collaborative dialogue perspective (including the gamification element) may or may not contribute to meeting some of the challenges. However, I do *firmly* believe that a *Human Centered* approach to continuous development is key to future progress towards creating better, more useful and more fitting software for business support, in particular in the context of continuous development. The collaborative dialogue perspective on RE, business modelling and systems design seems to indeed contribute something, both theoretical and concrete to the human centered approach to software engineering in general, and may

help understand and support the successful interaction between humans and machines at the development level.

References

1. Fitzgerald, B. and Stol, K.-J (2015): Continuous software engineering: A roadmap and agenda, *Journal of Systems and Software*, Volume 25, July 2015, Pages 1–14.
2. Kurt Sandkuhl, Hans Georg Fill, Stijn Hoppenbrouwers, John Krogstie, Andreas Leue, Florian Matthes, Andreas L. Opdahl, Gerhard Schwabe, Ömer Uludag and Robert Winter (2016): Enterprise Modelling for the Masses – From Elitist Discipline to Common Practice. In: *The Practice of Enterprise Modeling*, Volume 267 of the series *Lecture Notes in Business Information Processing* pp225-240; proceedings of PoEM 2016, Skövde, Sweden, Nov 2016.
3. Sandkuhl, Kurt; Fill, Hans-Georg; Hoppenbrouwers, Stijn; Krogstie, John; Matthes, Florian; Opdahl, Andreas; Schwabe, Gerhard; Uludag, Ömer; and Winter, Robert (2018) "From Expert Discipline to Common Practice: A Vision and Research Agenda for Extending the Reach of Enterprise Modeling," *Business & Information Systems Engineering*: Vol. 60: Iss. 1, 69-80. Available at: <http://aisel.aisnet.org/bise/vol60/iss1/6>
4. Theunissen, T.; van Heesch, U. (2016): The Disappearance of Technical Specifications in Web and Mobile Applications. In: *Proceedings of ECSA 2016*.
5. de Vreede, G. J., & Briggs, R. O. (2005). Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. In: *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS-38)*. Big Island, HI, USA: IEEE Computer Society.
6. S.J.B.A. (Stijn) Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. (Theo) van der Weide. Formal Modelling as a Grounded Conversation (2005). In: G. Goldkuhl, M. Lind, and S. Haraldson, editors, *Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling (LAP'05)*, pages 139–155, Kiruna, Sweden, EU, June 2005. Linköpings Universitet and Hogskolan I Boras, Linköping, Sweden, EU.
7. G. M. Nijssen and T. A. Halpin (1989). *Conceptual Schema and Relational Database Design: a fact oriented approach*. Prentice Hall, Englewood Cliffs, New Jersey. ISBN: 0-13-167263-0
8. G.E. Veldhuijzen van Zanten, S.J.B.A. (Stijn) Hoppenbrouwers, and H.A. (Erik) Proper (2004). System Development as a Rational Communicative Process. In: *Journal of Systemics, Cybernetics and Informatics*, Nr: 4, Vol: 2, pp47-51. International Institute of Informatics and Systemics (IIS).
9. P. van Bommel, S.J.B.A. (Stijn) Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide (2006): Exploring Modelling Strategies in a Meta-modelling Context. In: R. Meersman, Z. Tari, and P. Herrero: *OTM Confederated International Workshops and Posters 2006*, Montpellier, France, Proceedings, Part II. *Lecture Notes in Computer Science*, Vol. 4278
10. S.J.B.A. (Stijn) Hoppenbrouwers, P. van Bommel, and Aki Järvinen (2008). Method Engineering as Game Design: an Emerging HCI Perspective on Methods and CASE Tools. In: *Proceedings of EMMSAD'08 (Exploring Modelling Methods for System Analysis and Design)*, held in conjunction with CAiSE'08. Montpellier, France.
11. S.J.B.A. Hoppenbrouwers and E.A.J.A. Rouwette (2012): A Dialogue Game for Analysing Group Model Building: Framing Collaborative Modelling and its Facilitation. In:R.

- Magalhaes (edt.), *International Journal of Organisational Design and Engineering (IJODE)*, vol. 2, no. 1, p19-40; special issue on collaborative modeling. New York, USA: Interscience Publishers, 2012.
12. Hoppenbrouwers, S.J.B.A and Wilmont, I (2010): Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games. In: Bommel, P. van, Hoppenbrouwers, S.J.B.A., Overbeek, S., Proper, H.A., and Barjis, J.: *The Practice of Enterprise Modeling. Proceedings of the Third IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2010)*, held November 9-10 in Delft, the Netherlands. *Lecture Notes in Business Information Processing (LNBIP)* vol. 68. Berlin: Springer.
 13. Stijn Hoppenbrouwers: *Asking Questions about Asking Questions in Collaborative Enterprise Modelling* (2012). In: *The Practice of Enterprise Modeling, 5th IFIP WG8.1 Working Conference, PoEM 2012* (Rostock, Germany). Springer, LNBIP series vol. 134, p16-30.
 14. S.J.B.A. Hoppenbrouwers and van Stokkum, W. (2013): From Dialogue Games to m-ThinkLets: Overview and Synthesis of a Collaborative Modeling Approach. In: *International Journal of E-Collaboration (IJeC)* (9)4, 32-44. Special issue on Collaborative Usage and Development of Models and other Visualizations.
 15. S.J.B.A. Hoppenbrouwers, B. Schotten, and P.J.F. Lucas (2010): Towards Games for Knowledge Acquisition and Modeling. In: *International Journal of Gaming and Computer-Mediated Simulations, October-December 2010, Vol. 2, No. 4. Special issue on AI and Games*. IGI Global.
 16. Proper H.A., Bjeković M., van Gils B., Hoppenbrouwers S.J.B.A. (2018) *Towards Grounded Enterprise Modelling*. In: Debruyne C. et al. (eds) *On the Move to Meaningful Internet Systems. OTM 2017 Workshops. OTM 2017. Lecture Notes in Computer Science*, vol 10697. Springer, Cham
 17. Oppl, Stefan (2016): Evaluation of collaborative modeling processes for knowledge articulation and alignment. In: *Information Systems and e-Business Management, August 2017, Volume 15, Issue 3, pp 717–749*. Springer Verlag.