

# Managing Complexity in Process Digitalisation with Dynamic Condition Response Graphs

Thomas Hildebrandt<sup>1</sup>, Søren Debois<sup>1</sup>, Tijds Slaats<sup>2</sup>, and Morten Marquard<sup>3</sup> \*

<sup>1</sup> Department of Computer Science  
IT University of Copenhagen, {debois, hilde}@itu.dk

<sup>2</sup> Department of Computer Science  
Copenhagen University, tslaats@di.ku.dk

<sup>3</sup> Exformatics A/S, mmq@exformatics.com

**Abstract.** Digitalisation of work and business processes with the aim to provide more efficient services at a higher and consistent quality is high on the agendas in many countries. In Denmark the push for digitalisation is witnessed by the national strategy for digitalisation published every four years. Sadly, it is also witnessed by a number of expensive failed digitalisation projects. In this paper we point to two key problems in state-of-the art BPM technologies: 1) the use of rigid flow diagrams as the “source code” of process digitalisation is not suitable for managing the complexity of knowledge workflows and regulations and 2) insufficient support for continuous and agile end-user mapping and adaptation of processes. We report on research in progress on how the Dynamic Condition Response (DCR) Graph technology and collaborative process repository DCRGraphs.net could support agile and collaborative, bottom-up digitalisation of business and workflow processes.

## 1 Introduction

The idea of automating office work beyond computerising individual tasks goes back to at least the '70s. In his PhD thesis [12], inspired by Petri Nets [1], Zisman described office processes as imperative flow diagrams. There was a great deal of optimism about the approach, as illustrated by the following quote:

*By considering the specification language, the internal representation, and the design of a prototype system using one unified model, Zisman has been able to study the office as a system rather than simply as a collection of isolated tasks and pieces of equipment. Although Zisman suggests the language and the model need refinement, his basic notions will probably have great impact on the office of the future. [4]*

This optimism dried out during the '80s, but re-appeared in the '90s and '00s, when standards for internet web services and business process definition languages

---

\* This research is supported by ITU, Exformatics A/S, the Velux Foundation through the Computational Artefacts (CompArt) project (<http://www.compart.ku.dk>), and the Danish Council for Independent Research through the Hybrid Business Process Management Technologies project (DFR-6111-00337).

such as BPEL [11] and BPMN made it much more viable to share process diagrams and reuse information and web services across organisations. Hereto came the popularity of LEAN management and business process re-engineering methodologies [5] that support the ideas of mapping business processes and process optimisation. However it should be noted that business process re-engineering actually advocated a focus on eliminating tasks rather than automating them.

The mapping of processes unavoidably made companies and organisations more aware of their processes. However, time and experience has also shown that many diagrams were drawn and subsequently abandoned in desk drawers or large repositories and quickly became out of date.

So why is it, that we have not seen the great positive impact of the basic notion of flow diagrams and BPM technologies on office work as envisioned in the early '70s? We believe that flow diagrams have some inherent problems, related to how they deal with the unpredictability and the constant need for change that are key to office work.

First of all, flow diagrams do not explicitly capture why the activities are ordered in the given sequence, which makes them very difficult to update when the underlying reasons change. Consider implementing the rule found in the General Data Protection Regulation (GDPR), that if personal data is collected as part of signing a contract which is need for later but not obviously needed for offering the job, a consent from the employee must be given before signing. In a flow diagram, the rule can be implemented by making sure that an activity for getting consent appears on every path from the start of the process to the activity of signing the contract, but we can not specify the actual rule formally as part of the diagram. So, if we later update the diagram (or the rule gets changed) we have no formal guarantee that the rule will still be correctly enforced. This problem was experienced by the danish repository of local government workflows [3] created by Local Government Denmark, which ambitiously collected more than 900 workflows with links to associated law texts. By January 1st, 2013 Local Government Denmark gave up on keeping the repository up to date and compliant with the constantly changing regulations.

Secondly, many activities in an end-to-end business process, in particular processes involving some elements of knowledge work, can not be automated, in particular because the ordering of activities depends on many factors only known when the process is carried out and best determined and evaluated by human actors. Indeed, in the recent McKinsey report [8] on the impact of automation on future work processes, the most optimistic prediction is that on average less than half of the activities in future work processes will be automatable, and less than 5% of work processes will be fully automatable. Trying to predict and specify all possible orderings as a flow diagram leads to complex spaghetti diagrams, still suffering from the first problem. Trying to simplify the process, leads to rigid and in-flexible processes and workers by-passing their case management systems and prescribed processes.

Consequently, state-of-the-art process mapping and automation technologies yield too rigid processes and introduce a new complex manual process of producing and maintaining the process maps, which can not be carried out in practice and thereby results in proces descriptions that do not match reality. This leads to the research question of how to manage the complexity of digitalising knowledge work processes regulated

by law? Based on our work so far, we conjecture that an answer could be to support a continuous, agile and bottom-up process digitalisation that 1) provides process support that facilitates business users and knowledge workers, 2) captures both the why and the how of a process mapping, and 3) can be kept in synch with a changing reality.

Below we illustrate how the Dynamic Condition Response (DCR) Graphs technology and collaborative process repository DCRGraphs.net [10] supports these three activities. The technology will be the basis for a forthcoming research project in which the conjecture will be tested jointly with process consultants, lawyers and case workers in municipalities.

## 2 Agile Digitalisation with DCR Graphs

Dynamic Condition Response (DCR) Graphs [7] are the result of (so-far) 10 years of collaboration between IT University of Copenhagen (ITU) and Exformatics on the research and development of a declarative process technology that allows for flexibility and adaptability in both the design and support of business processes. During the last five years, DCR Graphs have been implemented in a commercial cloud service [2,6,10]<sup>4</sup>, which is freely available for academic and trial users. The service is at the same time a design, analysis and simulation tool, a social collaboration platform, and an active process repository. When logging in to the tool, users can connect to colleagues and friends in other organisations, with whom they can share and simulate processes. The technology is easily extended through its flexible software-as-a-service architecture and is rapidly improving based on continuous feedback from users in academia, public organisations, and private companies around the world.

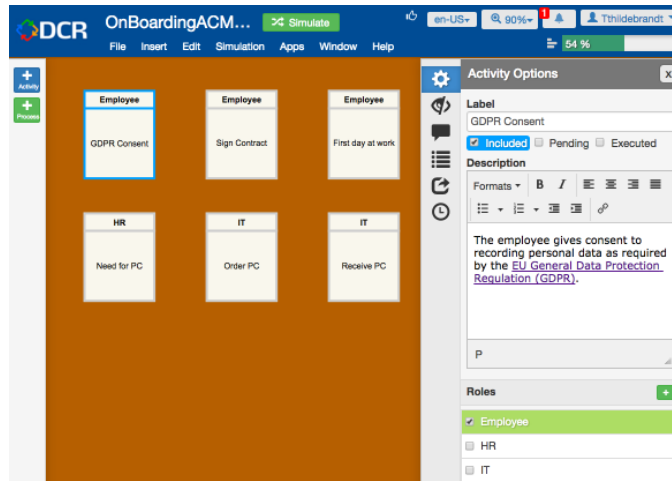
In the following we exemplify the collaborative and agile process mapping methodology supported by the tool using a prototypical example of an on-boarding process, which can be accessed at <http://www.dcrgraphs.net/Tool?id=4587>.

When using DCR Graphs, process mapping typically starts by placing activities as "post-it stickers" on a virtual "brown paper", labelled with an activity name and roles. Fig. 1 shows an example of such an initial graph, with six activities: GDPR Consent, Sign Contract, First day at work, Need for PC, Order PC, and Receive PC. (The activities are added simply by right clicking on the canvas). Detailed textual descriptions of each activity, including links to external references, can be added, as illustrated in the figure for the GDPR Consent activity. Each activity shown in the figure has been assigned one of the roles: GDPR Consent, Employee, HR and IT. At the present point, the role assignment is the only constraint of the process. Any of the activities can be executed any number of times by an actor with the correct role.

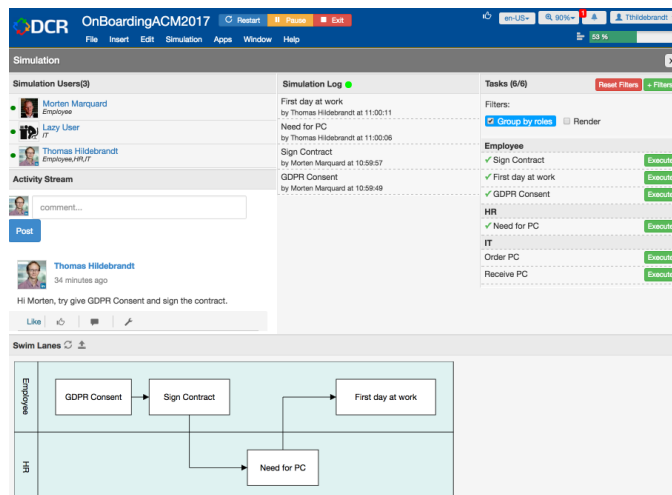
At any time, the modeller can simulate the process by pressing simulate and invite friends or automated users to play any of the roles. Automated users may join either as an aggressive user that performs every action that is possible and not executed before, or as a lazy user that performs an action if it is possible and required.

The collaborative simulation shows a task list interface resembling a case management tool, the history of performed actions as an event log and a swimlane diagram.

<sup>4</sup> <http://www.dcrgraphs.net/>



**Fig. 1.** DCRGraphs.net virtual "brown paper" with activities as post-it notes.



**Fig. 2.** Collaborative Simulation with friend playing the Employee role and a lazy user playing the IT role.

Thereby the end-users and modellers can explore and understand how the process can be performed. The screenshot in Fig. 2 shows a collaborative simulation with a friend, Morten Marquard, playing the role of Employee and an automated lazy user playing the role of the IT department. In the simulation there was a need for PC, but the PC was not ordered by the lazy IT department before the first day at work.

Simulations can be recorded as part of the documentation of the process and tagged as happy, un-happy or neutral, to indicate respectively, if the simulation is part of the desired, undesired or optional behaviour of the intended process.

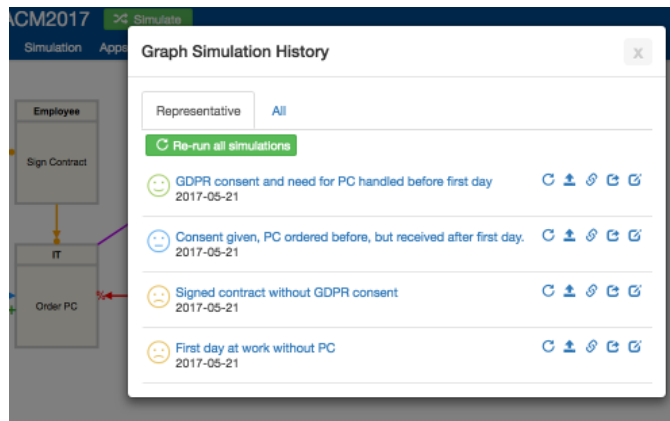


Fig. 3. Simulation History with Happy, Neutral and Unhappy simulations.

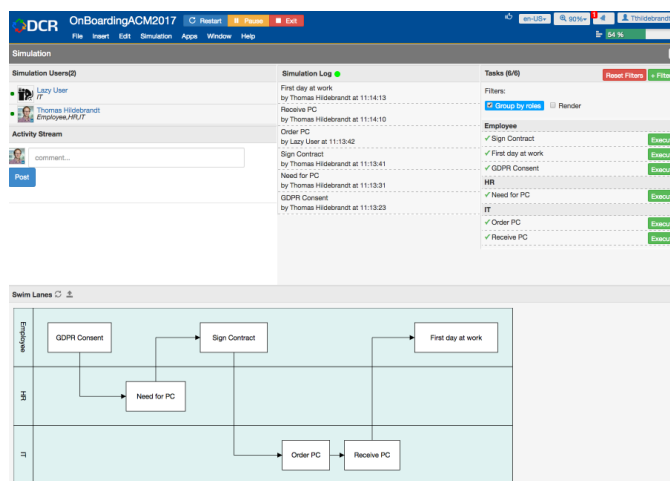


Fig. 4. GDPR consent and need for PC handled before first day.

The screenshot in Fig. 3 shows that we recorded the above simulation as unhappy, and named it “First day at work without PC”. We have also made an unhappy simulation without GDPR Consent before signing and a neutral simulation where consent is given and the PC is ordered (but not received) before the first day of work. Finally we made a happy path simulation, where the PC was also received before the first day. The latter example is shown in Fig. 4.

Simulations can be shared and replayed at any point, even if the process subsequently has changed. Moreover, saved simulations can be run as a test against the current version of the process, resulting in an indication by a green or red bar whether the execution trace can still be carried out. This allows a test-driven modelling approach,

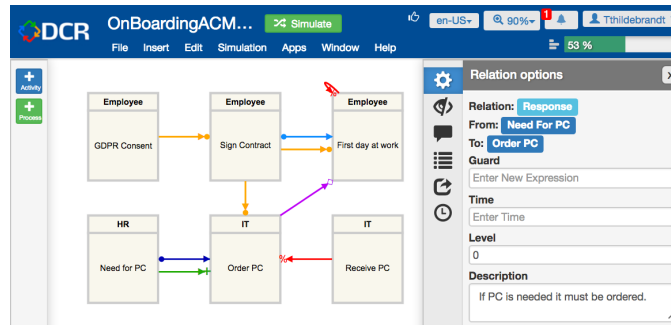
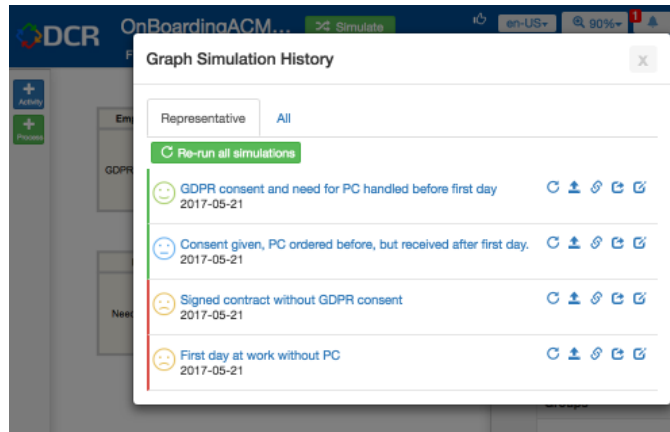


Fig. 5. DCR Graph with relations capturing why activities

where users can add and remove constraints to ensure that all happy paths are possible and all un-happy paths are ruled out.

In Fig.5 we show a process map where we have added DCR constraints to the process. The yellow arrows with a bullet at the target are condition constraints and model the fact that some activity can not happen before another activity has happened. In our case, we have modelled that the GDPR Consent must be given before the contract can be signed, and the contract must be signed before a PC can be ordered and before the first day of work. The blue arrows with a bullet at the source are response relations. The response relation from Need for PC to Order PC models, as explained in the Description field in the Figure, that if HR registers a need for PC, then IT must in response eventually order the PC. The response relation from Sign Contract to First day at work models that if the contract is signed it is expected that the employee eventually has a first day at work. The red arrow with a % sign at the target is an exclusion relation. The exclusion relation from First day at work to itself models that when First day at work happens, it is excluded and can thus not happen again. The exclusion arrow from Receive PC to Order PC models that Order PC is excluded from the process when a PC is received. Dually, the green arrow with a + sign at the target from Need for PC to Order PC is an inclusion relation, which models that if Need for PC happens, then Order PC is included (if it was previously excluded). Finally, the purple relation with a diamond at the target is a milestone relation. It models that as long as Order PC is pending (required as a response) and included, then First day at work can not happen. In other words, whenever a need for PC is registered, Order PC must happen again or a PC must be received before the first day of work.

We can now re-run the simulations, as shown in Fig 6, and see that we have ruled out the two unhappy paths and kept the happy and the neutral path. Looking back at the simulation in Fig. 4, it was actually performed on the constrained graph and with the computer playing the IT role as a lazy user. This time the PC was ordered by the lazy user after the contract was signed, because the action was required as a response to the need for PC and not enabled before the contract was signed.



**Fig. 6.** Re-running the simulations with the constrained graph we see that the unhappy paths are no longer possible.

### 3 Conclusion

The declarative specifications given by DCR Graphs capture the reasons for why actions must be done in certain order, and leave the flexibility to do anything which has not been ruled out. That is, a DCR graph explains the why, and relations and activities can be added and removed when requirements change, even at run-time. It is well known (see e.g. [13]) that it can be difficult for users to understand the logic of a complex declarative specification and deduce the allowed flows, i.e. understand the how. Similarly to the test-driven modelling approach considered for DECLARE in [13], the problem of understanding declarative models is mitigated in DCRGraphs.net by the ability to perform and record simulations of processes. We believe that the combination of collaborative design, simulation and test is a good way of facilitating an agile, incremental and bottom up approach to the digitalisation of processes: Domain-experts can start by identifying key activities and roles, followed by collaborative simulations of happy (desired), un-happy (un-desired) and neutral (optional) paths. Simulations can be run on the original process in which they were done or re-run against the current process. Our belief is backed by the empirical study of test-driven declarative modelling [13] indicating that,

*domain experts are inclined to use test cases ...and prefer them over the process model*

and

*the adoption of test cases significantly lowers cognitive load and increases the perceived quality of changes.*

The DCRGraphs.net approach is currently being evaluated in the mapping of a real on-boarding process of a danish municipality, involving more than 200 activities and a

great deal of flexibility. Finally, it is worth noting that the processes can not only be simulated. Processes made in DCRGraphs.net can be accessed and executed via a RESTful interface, or can be exported to the process engine of a case management system. Currently, two commercial case management systems support DCR Graphs, one being Exformatics ECM [6] and the other being the next version of KMD WorkZone [9]. In a forthcoming research project we plan to validate the conjecture that DCR Graphs based Adaptive Case Management solutions are superior to flow-graph based solutions for managing the complexity of knowledge work processes in local government.

## References

1. Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, editors. *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II, Proceedings of an Advanced Course, Bad Honnef, 8.-19. September 1986*, volume 255 of *Lecture Notes in Computer Science*. Springer, (1987).
2. Søren Debois, Thomas Hildebrandt, Tijs Slaats, and Morten Marquard. A Case for Declarative Process Modelling: Agile Development of a Grant Application System. In *EDOC Workshops '14*, pages 126–133. IEEE, September (2014).
3. Local Government Denmark. Arbejdsgangsbanken. <http://www.kombit.dk/arbejdsgangsbanken> last accessed 2017/08/10.
4. Clarence A. Ellis and Gary J. Nutt. Office information systems and computer science. *ACM Comput. Surv.*, 12:27–60, March (1980).
5. Michael Hammer. Reengineering work: Don't automate, obliterate. *Harvard Business Review*, 68(4):104–112, July-August (1990).
6. Thomas Hildebrandt, Morten Marquard, Raghava Rao Mukkamala, and Tijs Slaats. Exformatics declarative case management workflows as dcr graphs. In *International Conference on Business Process Management (BPM 2013)*, number 8094 in LNCS, Beijing, China, August (2013).
7. Thomas Hildebrandt and Raghava Rao Mukkamala. Declarative event-based workflow as distributed dynamic condition response graphs. In *Post-proceedings of PLACES 2010*, (2010).
8. McKinsey Global Institute. A future that works: Automation, employment, and productivity, january (2017). [http://www.mckinsey.com/-/media/McKinsey/Global%20Themes/Digital%20Disruption/Harnessing%20automation%20for%20a%20future%20that%20works/MGI-A-future-that-works\\_Full-report.ashx](http://www.mckinsey.com/-/media/McKinsey/Global%20Themes/Digital%20Disruption/Harnessing%20automation%20for%20a%20future%20that%20works/MGI-A-future-that-works_Full-report.ashx) last accessed 2017/08/10.
9. KMD. KMD WorkZone. <https://www.kmd.dk/loesninger/kmd-workzone> last accessed 2017/08/10.
10. Morten Marquard, Muhammad Shahzad, and Tijs Slaats. Web-based modelling and collaborative simulation of declarative processes. In *International Conference on Business Process Management*, pages 209–225. Springer International Publishing, (2015).
11. OASIS WSBPEL Technical Committee. Web Services Business Process Execution Language, version 2.0, (2007). <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf> last accessed 2017/08/10.
12. M. D. Zisman. *Representation, Specification and Automation of Office Procedures*. PhD thesis, Wharton School, University of Pennsylvania, (1977).
13. Stefan Zugal, Cornelia Haisjackl, Jakob Pinggera, and Barbara Weber. Empirical evaluation of test driven modeling. *IJISMD*, 4(2):23–43, (2013).