A Flexible Metamodelling Approach for Healthcare Systems

Author1, Author2, Author3

Institution Name, Country
email1, email2, email3

Abstract.

1 Introduction

Rising costs, ageing populations and increased expectations are making the current healthcare systems in the developed world unsustainable. Information technology has the potential to support healthcare but its application to support the continuum of care has not nearly reached its full potential. Barriers include the proliferation of systemseven within one hospital – which do not support interoperability; the fact that systems must be highly customized to adequately serve local situations, (usually a time consuming, and error prone process); the fact that change in health care is the only constant as adaptations to deal with such things as new medications, changing protocols and management strategies are constantly in need of being made; the fact that software engineering itself for such safety critical systems as healthcare needs new strategies to ensure that systems behave correctly in every possible scenario; and finally, the fact that the healthcare process itself is often a team process involving many players, including family physicians, nurses, therapists of various kinds, specialists, lab technicians, managers of both hospital and community based programs, all with different needs from IT and all with different and often limited ability to handle complex technology. The active participation of the patient (and his family) in the management of his own health is becoming a critical issue as the cost of chronic diseases is quickly out pacing the resources that can be directed to healthcare.

MDSE is an emerging and promising methodology for software development, targeting challenges in software engineering relating to productivity, flexibility and reliability of systems. The construction of various kinds of models (eg. blueprints, mockups etc) is a well-known approach in the more traditional engineering fields; these models are used as artefacts to enable engineers to describe designs and validate whether a proposed design has desired qualitative and quantitative properties. We propose to use MDSE in an analogous manner for the development of reliable and robust workflow software systems supporting the diverse and often safety critical requirements of healthcare process in both primary care and community care environments.

While the use of executable graphical languages and domain specific modelling languages greatly reduces the possibility of coding errors, these techniques do not follow the metamodelling approach and may lead to impaired links between the workflow model and the generated code. Many different MDSE technologies automatically generate code from models [ADD Tools, REFERENCES]: these technologies are particularly suited to specifying the structural aspects of software systems; generally, whereas the actual behavior is programmed manually. Some technologies for behavioural modelling in MDSE exist ([2], [1]), but current approaches are often at a low level of abstraction and lack domain concepts for specifying behavior [6].

A collaborative group of researchers in Norway and Canada have been working on various issues relating to these problems. We [17], [14], [8] proposed a formal approach to workflow modelling based on the Diagram Predicate Framework (DPF) [13], [16] which provides a formalism of (meta) modelling and model transformations based on category theory and graph transformations ([4], [5], [Ehrig etal]) We [19] extended the formal foundation of DPF to define (static) semantics for timed and compensable workflow models and defined the dynamic semantics of models by a transition system where the states are instances and transitions are applications of transformation rules, and showed how to exploit reduction methods and sweepline methods to model check complex workflows [7]. We developed a domain specific language to expedite workflow system development [12], [11] and began the development of a user-friendly interface to allow the health practitioner to determine the correctness of behavioral properties of a healthcare workflow protocol [15].

2 A metamodeling approach to healthcare workflows

In [10] we discussed the implementation of a user-friendly tool to aid clinicians and patients in the workflow process implied in a clinical practice guideline with an accompanying module that gives the patient a user friendly interface for self management of lifestyle attributes which caused the patient to be at risk. The patient can input data for such lifestyle attributes such as, exercise, smoking, intake of fruits and vegetables and record such attributes as weight and blood pressure. The web-based tool would allow both the patient and the clinician to view summary data on lifestyle parameters between visits [give some pictures here!!] and provide calendar views of past activities, future appointments, etc.

The PhD thesis of [3] promoted a metamodelling approach to the development of workflow processes in healthcare settings, which incorporated the concept of monitor. Many healthcare processes involve numerous stakeholders with different requirements, and the user becomes a critical part of the healthcare workflow process whether it be the physician, a specialist, a lab tech or the patient, in situations where management of lifestyle parameters is a critical component of the process. We are now researching a metamodelling approach to workflows which incorporates the concepts of stakeholder and monitor and provides user friendly interfaces for a variety of users. Such processes are safety critical and must be adapted to particular clinical setting and changed/updated frequently due to changing guidelines, medications, and patient preferences; thus they serve as an excellent application domain for the development of evolving, adaptable and correct software systems, in general.

In his thesis, Baarah [3], proposed an application framework for care process monitoring that collects and integrates events from event sources, maintains the individual and aggregate state of the care process and populates a metrics datamart to support performance reporting. He presented a UML-style metamodel for the care process monitoring application that had 3 main components: a process model , a performance model and an enterprise model. The process model defines the care process in terms of states to be monitored, resources and rules that specify the transition from state to state as events are received from the enterprise model. The performance model measures how well the goals for the care process are being achieved in terms of metrics computed from the monitored states, and events for the process. Alerts are defined to flag when targets are not being met. No automated implementation of the metamodel was attempted, correctness of the process was investigated only through the use of test scripts, and user interface issues were not considered.

In the thesis, Baraach basically structured a workflow system into 3 components:

- a workflow component;
- a monitoring component this will typically send alerts etc. when a critical event occurs; and,
- a data source component to persist the data that are relevant in the workflow.

We extend this model to include users and user interactions; allowing us to model user interaction as part of the process. The users may interact with:

- part of the workflows
- part of the database
- the monitoring system, that is, users will typically receive alerts from the monitoring system

All interactions occur at specific times. A workflow consists of a network of tasks and specified users are responsible for performing each task. While performing a task a user provides data to the system, typically, this is filling out a standardized web-based form, or some form which provides automatic integration with the appropriate healthcare database.

3 Metamodelling of a healthcare system

We modeled a Hypertension management workflow from the guideline of clinical pathways [1]. In this section we provide an overview of the system. We propose

[1] The Chinook Primary Care Network: `http://www.chinookprimarycarenetwork.ab.ca`

the use of seperate meta models for the user access, workflow process, monitor, UI view, data source and integrate them by using indexed set (see Fig. 1). The directed arcs from one meta model to another in Fig. 1 represent abstract relations between meta models. Using seperate meta models for modeling different aspects of a system gives us the flexibility for remodelling and also makes models more readable.

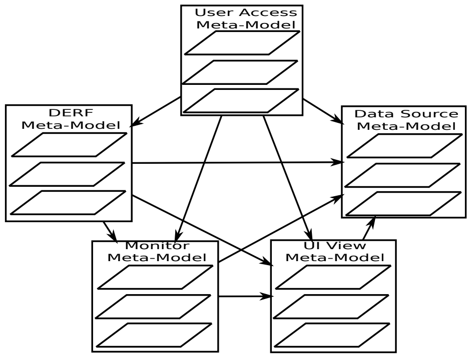

Fig. 1: Multiple meta modelling hierarchy

For workflow modeling we used DERF [14,18] where one can graphically design a workflow model. Fig. 2 shows the overall model of the Hypertension Management Workflow. There are some composit tasks such as 'Visit1', 'CBPM' in the workflow, those are abstractions of subworkflows. The subworkflow for 'Visit1' composit task is shown in Fig. 2. These abstractions are sometime useful in order to reduce the number of repetation.

Initially patients' blood pressure is measured at the 'Initial BP' Task which may cue the clinical hypertension management procedures if the BP is found to be greater than or equal to 140/90. If the initial BP is found normal (<= 140/90), the workflow terminates. In Fig. 2 patients' Hypertension is managed through investigation and treatment. The clinical procedure (i.e., 'Visit1', and all other subsequest tasks in Fig. 2) starts at the doctors' office. Patients with high BP have risk of organ failures and/or other chronic illness. During the first visit at doctors' office ('Visit1') BP is measured twice, an initial assessment is done, and an investigation is started with diagnostic tests. The workflow executes 'Ambulatory blood pressure monitoring' ('ABPM') or 'Self home blood pressure monitoring ('SHBPM') if they are available otherwise a Clinical BPM is performed which refers to the 'CBPM' task in the overall workflow model.

The user access to a DERF workflow model is defined by an Indexed set. We defined a seperate meta modeling hierarchy for the user model. The left hand side of Fig. 3 shows two modelling levels M_2 and M_1 of a user model where $\mathfrak{S}_{u_2}$ and $\mathfrak{S}_{u_1}$ are the specifications (respectively). The 'Copy-acc' arc is used to copy

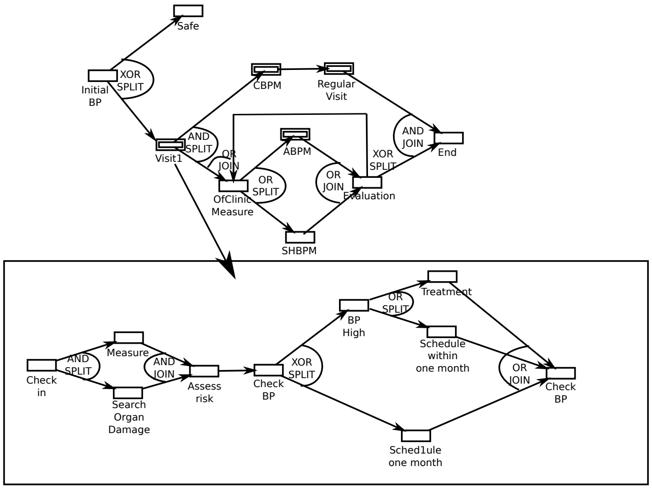

Fig. 2: Hypertension Management Workflow (Overall)

access from one user to another. One instance of the 'Copy-acc' arc from 'Doctor' to 'Nurse' in Fig. 3 gives Doctors all the access that Nurses have. The right hand side shows two modelling levels of a DERF workflow model with specifications $\mathfrak{S}_{w_2}$ and $\mathfrak{S}_{w_1}$.

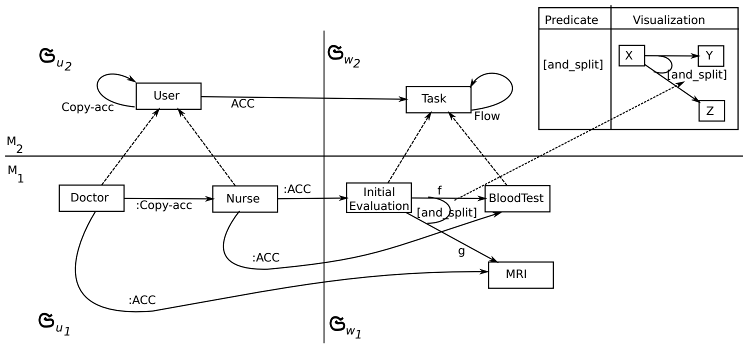

Fig. 3: Meta modelling of User Access to Workflow model

To visualize the user access for a workflow model we propose an user interface in Fig. 4 where the user nodes (e.g., Doctor, Nurse, Patient) are displayed at the bottom of a workflow model. Selection of a user node from the bottom window highlights all the accessible tasks from the workflow model. In the figure, the user Patient has only access to the 'SHBPM' that has been shown in gray color.

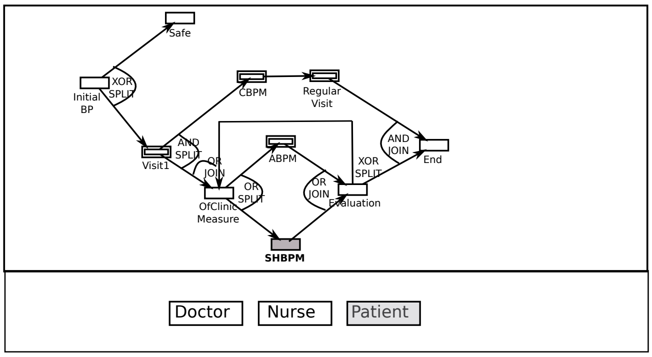

Fig. 4: Visualization of Hypertension Workflow Model and User Access

While executing 'SHBPM' task the patient registers his life style information. So in this workflow the patient is responsible for registering his life style information and doctors and nurses are responsible for the rest of the workflow. Both the doctor and the patient should have access to the lifestyle information. Doctors have a user interface a bit similar to the one the patient uses, but it has many more features (e.g., sending an e-mail to the lab for a lab test).

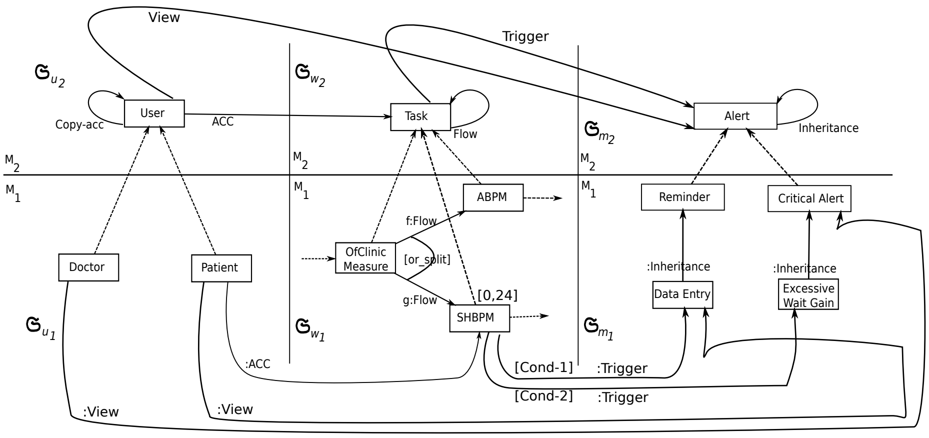

Fig. 5: Meta modelling of Workflow monitor

Fig. 6 shows a meta model of workflow monitor having a seperate meta model hierarchy and its association with a DERF workflow meta model and a user access meta model. Tasks from a DERF workflow model can trigger alerts. We have two types of alerts: 'Critical Alert' and one less urgent called 'Reminder'.

The 'SHBPM' task triggers a 'Data Entry' alert if the patient forgets to enter data on some day. It also triggers a 'Excessive Weight Gain' alert if the patients weight gain over fewer than 5 days exceeds 10 pounds. This firing condition is encoded in the predicate $[Cond - 2]$. Different users have different view access to the alerts. In this figure, the doctors are only informed about the 'Excessive Weight Gain' alert to indicate that the patient is retaining excessive fluids and the doctor should consult that patient immediately. The patient has view access to the 'Data Entry' alert. In DERF workflow a task may have time constraints [19]. The task 'SHBPM' has a time constraint of '0 hour' delay and '24 hours' duration. $[Cond - 1]$ is a predicate that triggers the 'Data Entry' alert if the '24 hours' duration is over and the 'SHBPM' task is not executed.

The meta model of the 'Data Source Meta Model' and 'UI-View Meta Model' are out of the scope of this paper.

4 User-friendly user interface

We developed a user-friendly user interface to interact with the system. Fig. 6 shows 4 windows named 'Workflow Viewer', 'Task Execution Viewer', 'Lookup Viewer', and 'NOVA Browser'. 'Workflow Viewer' shows the list of all available tasks on the right side of the window and whenever a task is executed, it shows the task name on a calendar. By default the calendar shows the 'month view' but different level of granularity may be configured, such as: Yearly view, weekly view, hourly view, etc. A task being executed are shown in the 'Task Execution Viewer'. Inputs required from the end user are shown in branches and the end user is supposed to select a branch and assign a value to it through the 'Lookup Viewer'. The 'Lookup Viewer' helps the end user to input information either by showing relevant values from an ontology or by allowing end user to enter information. Once a task is executed, the information are hierarchically displayed in the 'NOVA Browser'. We introduced 'NOVA Browser' in [9]. Using 'NOVA Browser' user may expand a branch to see detailed information, may use the time travel view to traverse backward/forward and view information. the details Fig. 6 is showing that the user is executing the 'Measure BP' task. Inside the 'Task Execution Viewer' window user provides input to execute the task. This is an alternative way of taking user input instead of using *Forms*. This view can also be configured to traditional 'Form view' where the user provides input in 'Form fields' (e.g., text boxes, drop down boxes, etc.). While executing tasks the user has access to historical information, workflow instance; this arrangement provides the user more user-friendly environment to concentrate on care.

The 'Lookup Viewer' at the bottom left of Fig. 6 provides options allowing the user to select or enter data. This window is connected to a database and shows only relavent data from the database. This view can also be configured to connect to an ontology and show relevant terminology from an ontology to help user input information while executing a task. Data inserted or selected by the user in the 'Lookup Viewer' is reflected in the 'Task Execution Viewer' window.

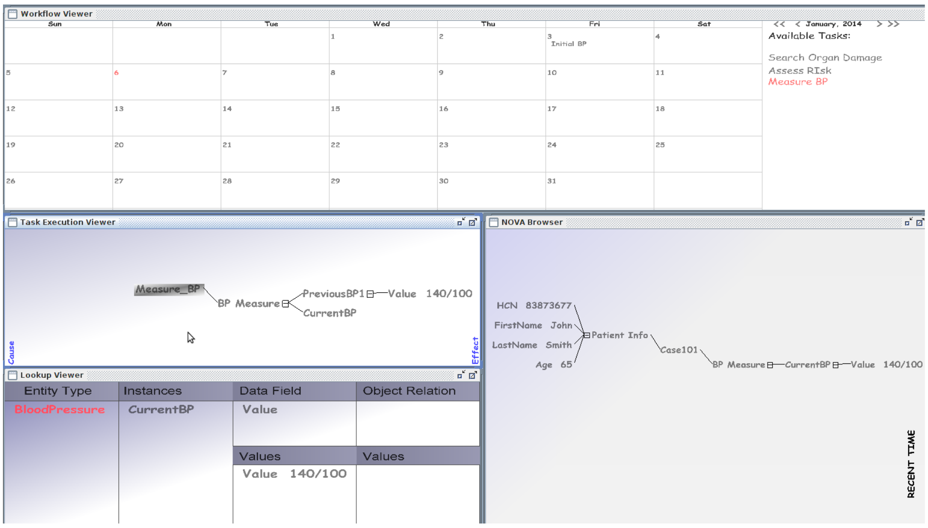

Fig. 6: Hypertension Management Workflow (Visit1)

When the user is finished entering all input for a task, the task is executed and this updates ‘NOVA Browser’ nodes.

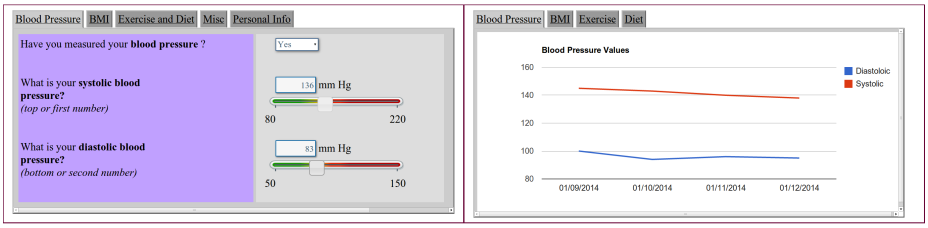

Fig. 7: Personal health monitor (Blood pressure monitoring)

We developed a ‘Personal health monitor’ smart phone application for the patients to execute the ‘SHBPM’ (see Fig. 2) from home. The purpose of the application is to assist patients keeping their health record such as blood pressure record, body mass index, hours of exercise, dietary, etc. and monitor their performance with their lifestyle target that was set by the physician from ‘Hypertension management workflow’. It is required that the ‘Personal health monitor’ application interacts with the ‘Hypertension management workflow’. The integration has been accomplished by a task named ‘SHBPM’ (see Fig. 2) from the ‘Hypertension management workflow’.

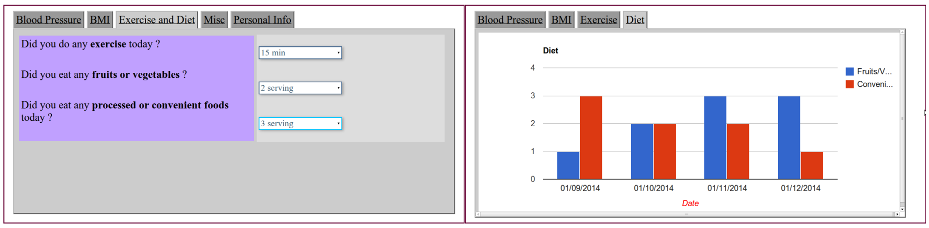

Fig. 8: Personal health monitor (Exercise and diet pressure monitoring)

We have developed several interfaces that give summary data to the patient and doctor by projecting patient's data into graphs and charts. Fig. 7 shows two screenshots from the smart phone application that takes blood pressure input from the patient and displays a graph of their recent blood pressure changes. Fig. 8 shows another two screenshots that patient can use to monitor their exercise and eating behavior change. This smart phone appliction can also fetch appointment date from the workflow and reminds patient about the next visit date.

5 Conclusion

Incorporating stakeholders and monitors in the MDSE paradigm is highly innovative, however these features are essential if software systems are to automatically perform the kinds of tasks users are increasingly demanding. We believe that the metamodeling together with Model Driven Software Engineering principles in general (1) can be used as the main artefact in the development process of correct and adaptable workflow software systems targeting healthcare applications which incorporate monitors for sensitive data parameters and interfaces for a variety of stakeholders, and, moreover, (2) can be manipulated by computer tools to automatically produce a workflow implementation which can be deployed in healthcare settings, which can easily be adapted to reflect the numerous customizations and changes required in a healthcare setting due to requirements and limitations of local settings, updates in protocols, etc. The DPF framework is used to implement our ideas. A demo of a small prototype may be found at: `http://www.screencast.com/t/QxG5knxVoO`.

Acknowledgement

We would like to thank Natural Sciences and Engineering Research Council of Canada, Atlantic Canada Opportunities Agency (ACOA) for supporting this research. We are greatful to Miao Huang for his contributions to the Hypertension Workflow Model and Jane Newlands for providing valuable feedbacks to us.

References

1. Fujaba Development Team: The Fujaba Tool Suite, http://www.fujaba.de/
2. Object Management Group: Semantics of a Foundational Subset for Executable UML Models (FUML) (February 2011), http://www.omg/spec/FUML/1.0/
3. Baarah, A.H.: An application framework for monitoring care processes. PhD thesis, University of Ottawa, 2013
4. Barr, M., Wells, C. (eds.): Category Theory for Computing Science, 2Nd Ed. Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK (1995)
5. Diskin, Z., Wolter, U.: A diagrammatic logic for object-oriented visual modeling. Electr. Notes Theor. Comput. Sci. 203(6), 19–41 (2008)
6. Kindler, E.: Model-based software engineering: The challenges of modelling behaviour. In: Proceedings of the Second International Workshop on Behaviour Modelling: Foundation and Applications. pp. 4:1–4:8. BM-FA '10, ACM, New York, NY, USA (2010), `http://doi.acm.org/10.1145/1811147.1811151`
7. Kristensen, L.M., Lamo, Y., MacCaull, W., Rabbi, F., Rutle, A.: Exploiting progress for memory efficient verification of diagrammatic workflows. 25th Nordic Workshop on Programming Theory (2013), Extended Abstract.
8. Lamo, Y., Wang, X., Mantz, F., Bech, Ø., Rutle, A.: Dpf editor: a multi-layer diagrammatic (meta) modelling environment. Proceedings of SPLST (2011)
9. MacCaull, W., Rabbi, F.: NOVA Workflow: A Workflow Management Tool Targeting Health Services Delivery. In: FHIES 2011: International Symposium on Foundations of Health Information Engineering and Systems (2011)
10. Rabbi, F., MacCaull, W.: Designing user-friendly uis for the execution of clinical practice guidelines. 27th International Symposium on Computer Based Medical Systems, 2014, Submitted
11. Rabbi, F., MacCaull, W.: Model driven workflow development with $T_{\Box}$;. In: Bajec, M., Eder, J. (eds.) CAiSE Workshops. Lecture Notes in Business Information Processing, vol. 112, pp. 265–279. Springer (2012)
12. Rabbi, F., MacCaull, W.: $T_{\Box}$: A Domain Specific Language for Rapid Workflow Development. In: MODELS 2012. pp. 36–52. Lecture Notes in Computer Science, Springer (2012)
13. Rutle, A.: Diagram Predicate Framework: A Formal Approach to MDE. Ph.D. thesis, Department of Informatics, University of Bergen, Norway (2010)
14. Rutle, A., MacCaull, W., Wang, H., Lamo, Y.: A Metamodelling Approach to Behavioural Modelling. In: Proceedings of BM-FA 2012: 4^{th} Workshop on Behavioural Modelling: Foundations and Applications. pp. 5:1–5:10. ACM (2012)
15. Rutle, A., Rabbi, F., MacCaull, W., Lamo, Y.: A user-friendly tool for model checking healthcare workflows. Procedia Computer Science 21(0), 317 – 326 (2013), the 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013) and the 3rd International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH)
16. Rutle, A., Rossini, A., Lamo, Y., Wolter, U.: A formal approach to the specification and transformation of constraints in mde. J. Log. Algebr. Program. 81(4), 422–457 (2012)
17. Rutle, A., Wang, H., MacCaull, W.: A formal diagrammatic approach to compensable workflow modelling. In: Weber, J., Perseil, I. (eds.) FHIES. Lecture Notes in Computer Science, vol. 7789, pp. 194–212. Springer (2012)
18. Rutle, A., Wang, H., MacCaull, W.: A Formal Diagrammatic Approach to Compensable Workflow Modelling. In: Liu, Z., Wassyng, A. (eds.) Foundations of Health

Informatics Engineering and Systems, Lecture Notes in Computer Science, vol. 7789, pp. 194–212. Springer Berlin Heidelberg (2013)
19. Wang, H., Rutle, A., MacCaull, W.: A Formal Diagrammatic Approach to Timed Workflow Modelling. In: Proceedings of TASE 2012: 6^{th} International Conference on Theoretical Aspects of Software Engineering. vol. 0, pp. 167–174. IEEE Computer Society (2012)