

Domain-specific Design of Patient Classification in Cancer-Related Cachexia Research

Alexander Wickert
University of Potsdam
Potsdam, Germany
alwicker@uni-potsdam.de

Anna-Lena Lamprecht
Utrecht University
Utrecht, Netherlands
a.l.lamprecht@uu.nl

Tiziana Margaria
University of Limerick and Lero
Limerick, Ireland
tiziana.margaria@lero.ie

ABSTRACT

We apply an IDE for user-level process design and composition to a real-life case study: a complex workflow from an ongoing global cancer-related cachexia research project. Originally buried in a manually operated spreadsheet, the process is now fully automated and integrated into the project database, ensuring the immediate availability, consistency and reproducibility of the outcomes. Our integrated solution enables the scientists to immediately execute the processes and easily customize both processes and data model to continuously changing experimental setups. The data modeling is provided by the Dynamic Web Application framework and the process modeling functionalities by the Java Application Building Center, both following the paradigm of eXtreme Model-Driven Design for model-driven software development.

KEYWORDS

scientific workflows, workflow design, process modeling, domain-specific modeling, model-driven software development

ACM Reference Format:

Alexander Wickert, Anna-Lena Lamprecht, and Tiziana Margaria. 2018. Domain-specific Design of Patient Classification in Cancer-Related Cachexia Research. In *FormalISE '18: FormalISE '18: 6th Conference on Formal Methods in Software Engineering, June 2, 2018, Gothenburg, Sweden*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3193992.3194002>

1 INTRODUCTION

Researchers in the life sciences are often not IT professionals, thus they need comprehensive computational support to cope with the data collected in their labs and the data processing workflows. Typically, they use one or more computational frameworks (e.g. Matlab, GNU R), but not in a setup to make them work together efficiently, nor with the data definition and management techniques that complex and evolving experimental settings require.

In a collaborative project, headed by a Cancer Metabolism Research Group in São Paulo and with a large number of participating research groups spread over three continents, we have witnessed

the IT challenges that scientists face when they create a large, multidisciplinary and distributed experimental infrastructure in the field of cancer-related cachexia research [3, 8–10]. The project team needs to cooperate across different languages and a variety of scientific backgrounds, therefore a semiotically intuitive, graphical approach is preferred over textual explanations and documents. To provide a suitable overall IT strategy, we needed to understand how the researchers work and how to enable them to define and describe their evolving experimental procedures on their own, including data and process management. Accordingly, the central challenge is to provide a framework that is able to manage the complexity and the change and growth of these workflows, and that helps to efficiently produce more reliable results, while at the same time appearing simple and intuitive to its users (cf. [11]).

In this paper, we describe how we apply an IDE for co-design and definition of data and process models based on the functionalities provided by the Dynamic Web Application (DyWA) and the Java Application Building Center v.4 (JABC4) modeling framework, to support this group of users with easily executable domain-specific processes. The complex patient classification in the domain of cachexia research determines which individuals are suited to belong to the different patient and control groups that are analyzed. It is a multifaceted evaluation of a number of interdisciplinary criteria, using data collected by a large number of individual professionals. We organized and simplified the patient classification process and made it easily accessible and shareable worldwide via a web application. The integrated framework now collects the (anonymized) data of patients in a central repository, easy to access and to back up, and also allows for agile adaptations of the scientific workflows and the underlying data model.

2 PATIENT CLASSIFICATION IN EXCEL

Cachexia is a complex wasting syndrome associated with a marked detrimental effect upon life quality and survival in patients with cancer, chronic obstructive pulmonary disease, chronic heart failure, AIDS, and chronic kidney disease, among other conditions [20]. The cachexia definition of Evans et al. [4] is multicriterial: cachexia can be diagnosed if the patient shows an unintended weight loss of at least 5% in 12 month or less, or has a BMI below 20kg/m^2 and has at least 3 of the following 5 symptoms: (1) decreased muscle strength, (2) fatigue, (3) anorexia, (4) low fat-free mass index, and (5) abnormal biochemistry.

Fig. 1 shows the original Excel-based spreadsheet tool used for the patient classification based on an adaptation of this cachexia definition. Refined and improved in many rounds, it was the starting point for our case study. The spreadsheet combines the cachexia classification with the available cancer diagnosis, yielding a group

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

FormalISE '18, June 2, 2018, Gothenburg, Sweden

© 2018 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.
ACM ISBN 978-1-4503-5718-0/18/06...\$15.00
<https://doi.org/10.1145/3193992.3194002>


PATIENT'S INFORMATION				FIRST CRITERION - WEIGHT LOSS		
Identification	Gender	Age (Years)		Weight variation	BMI (kg/m ²)	Result
1654	Male	51		-10%	28.88	IN
Height (m)	Prev. weight (kg)	Current weight (kg)		Treatment		Hernia
1.59	81	73				
SECOND CRITERION - WEIGHT STRENGTH				THIRD CRITERION - FATIGUE		
Method	Score		Result	Method	Score	Result
Questionnaire (QLC-C30)	53.33333333		OUT	Questionnaire (QLC-C30)	33.33	IN
Answer 1	1			Answer 10	4	
Answer 2	4			Answer 12	3	
Answer 3	2			Answer 18	2	
Answer 4	3					
Answer 5	2					
FOURTH CRITERION - ANOREXIA				FIFTH CRITERION - FAT FREE MASS INDEX		
Method	Score		Result	Method	Score	Result
Questionnaire (QLC-C30)	100.00		OUT	DEXA Scan	6.09	IN
Answer 13	1			Lean mass (kg)	15.4	
SIXTH CRITERION - BIOCHEMICAL PARAMETERS				GROUP CLASSIFICATION		
Parameters	Concentration		Result	BARCODE		
C-reactive protein (mg/l)	6.10		IN	CACHEXIA WITHOUT CANCER		
IL-6 (pg/ml)	5.34			LEVEL OF EXCLUSION CRITERIA		
Anemia - Hb (g/dl)	12.30			1.54CACHXIA		
Albumin (g/dl)	4.89					
Adapted from Evans, 2008				 0 NONE		

Figure 1: The patient classification spreadsheet.

```
=IF(AND(A9="Questionnaire (QLC-C30)"; B10<>""; B11<>""; B12<>""; B13<>""; B14<>""); (1-(AVERAGE(B10:B14)-1)/3)*100; IF(AND(A9="Handgrip Test"; B3="Male"; B10<44; B10<>""); "POOR"; IF(AND(A9="Handgrip Test"; B3="Male"; B10>44; B10<>""); "GOOD"; IF(AND(A9="Handgrip Test"; B3="Female"; B10<23; B10<>""); "POOR"; IF(AND(A9="Handgrip Test"; B3="Female"; B10>22; B10<>""); "GOOD"; "")))
```

Figure 2: Exemplary formula from the spreadsheet (cell B9).

classification result (cell E23) for this patient: Cancer Cachectic, Cancer Without Cachexia, Cachexia Without Cancer (like in Fig. 1), Control, or Excluded By Weight.

Clearly visible from this figure, the tool collects for each patient basic anthropometric information (ID, gender, age, height, previous and current weight) and the results from several analyses carried out by specialists, and then evaluates the five criteria. As the Excel sheet is a stand-alone tool, all these data need to be entered manually. The tool classifies correctly, but for regular use in a large-scale, long-term, international and interdisciplinary research project it has two significant drawbacks:

- (1) For every new patient, someone has to manually enter the individual values. When all required cells are filled with valid values, the cells in *Group Classification* will yield the classification result. Then, this result has to be transferred where it is needed – again manually. Clearly, this approach is error-prone and does not scale.
- (2) The Excel formulas are complex and not easily understandable. For example, cell B9 (Fig. 2), evaluates the score of the second criterion, *Weight Strength*. As such, maintenance, modifications, and extensions are difficult and error-prone.

3 STATE OF THE ART

So far, we have found no other framework or solution that integrates the modeling of the data types *and* the related processes, providing comfortable access and design capabilities to the domain experts. OpenClinica [19] and similar products offer a web-based software mainly for managing clinical data and building custom studies, but do not offer modeling of processes, especially by domain experts, and thus also no analysis and enactment. The direct use of a pure PostgreSQL database allows no flexible adaptation

and customization of the data model for users with lower IT affinity, and no process definition ability. Workflow modeling tools from the scientific community such as Taverna [18] or Kepler [1] lack the integration of domain-specific modeling capabilities, the scalability to large workflows, and the integration in a modern web application framework.

4 INCREMENTAL MODELING OF DATA AND EXECUTABLE PROCESSES

Originally carried out with a manually operated spreadsheet, the patient classification process is now fully automated as a workflow modeled with the jABC4 [16, 21] and integrated into a database provided by the DyWA [6, 17]. jABC4 is a model-driven environment for designing the processes of the workflow, whereas DyWA is a meta-schema based data definition and management tool that caters to standard relational databases. Their interplay provides an integrated environment for data and process modeling along the eXtreme Model-Driven Design (XMDD) [14] paradigm. This in turn supports a Service-oriented Continuous Engineering approach [13] to the formalization and definition of a domain-specific language and process landscape for patient triage, classification and scoring.

4.1 Co-Development of Types and Processes

The evolutionary co-design of domain-specific data types and processes with the DyWA and jABC4 frameworks works as follows:

- (1) The scientists start with the definition of some **data types** inside the DyWA framework (cf. Sec. 4.2),
- (2) **Microservices** for the CRUD operations (i.e. Create, Read, Update, Delete) are automatically generated for every defined type and field by the DyWA and immediately exported to the jABC4 (cf. Sec. 4.3),
- (3) In the jABC4, the microservices are used to design the **workflow models** that manipulate the data (cf. Sec. 4.4). Specifically, our microservices are *Service Independent Building Blocks* (SIBs) [5, 7, 21]. At the same time, the workflow graphs are formal models akin to Kripke Transition Systems [12], thus analyzable with techniques like model checking. Domain-experts **model and validate** the processes inside the jABC4, then export them to the DyWA framework.
- (4) Finally, the processes are seamlessly **executed** by the scientists in the DyWA web interface: when entering the data of a patient, the processes that compute the individual and global scores are executed and the outcomes are made available to the user and persisted in the database. (cf. Sec. 4.5).

In a live environment there are several iterations of process and data type refinement.

4.2 Data Type Definition

The natural approach of the scientists turns out to be inherently well aligned with a DSL design approach: they start by fixing the vocabulary of the domain. In the first step, this happens through the web interface of the DyWA: they define a set of domain-relevant “things” with their respective types. DyWA provides the domain-independent Java types (e.g. String, Integer) as initial type collection.

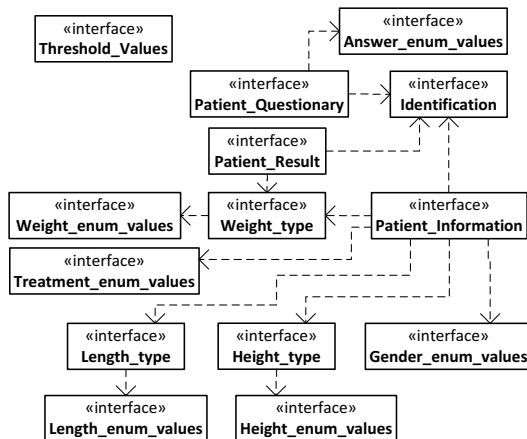


Figure 3: UML class diagram of the DyWA domain model.

Any self-modeled domain specific type (e.g. *Weight*) becomes directly available and can be used as a field or attribute of complex data types, which are like records or tables in a database schema.

Fig. 3 shows the UML representation of the complete DyWA domain model of the patient classification data. Every interface class in the figure has a corresponding implementation class (not shown here) with methods to get and set the data fields of the DyWA types. Many enumeration types enable DyWA's users to only choose valid values from drop-down lists.

4.3 Generation and Export of CRUD SIBs

From the DyWA domain model, a generator automatically creates the SIBs for the CRUD operations of the data types and its fields. They encapsulate domain-specific services and are added to the pre-existing collection of domain-independent services and services from other domains. The actual processes in the jABC4 use these building block collections. While designing these processes, one can successively change the underlying domain model in the DyWA at the same time, thus there can be many short cycles of improvement and refinement.

208 CRUD SIBs are generated for the domain of cachexia classification. The 37 modeled processes are themselves reusable blocks for other processes, so the total number of available SIBs is 245; classified by the taxonomy in Fig. 4.

4.4 Process Definition

The processes are modeled in jABC4 using the available SIB palettes just described. The calculations contained in the Excel in Fig. 1 result in a collection of 37 modularized and reusable processes. The maximum depth in the graph hierarchy is two. Fig. 5 depicts one possible path through the three levels of the graph hierarchy ending up in the most specific calculation process *Criterion3_Calculate_QLC_C30_Score_Calculation*. More complex processes are non-linear, with control flows that include decision points and loops.

4.5 Process Execution

The application user can see all available processes in the main view. The execution of a process is triggered by click on a button.

If the process has inputs, corresponding objects need to be chosen. Then a message of un-/successful execution is shown. The output results are automatically stored in the DyWA database and can be looked up in the web application view. At any time, the processes can be adapted in the jABC4 framework and redeployed to the DyWA framework with one single Maven command.

5 DISCUSSION AND CONCLUSION

We showed how the transformation of a spreadsheet-based tool into an integrated data and process modeling environment that blends into the existing research practices of the lab results in improvements concerning data collection, data transformation, automation and reproducibility of results. Overall, the transformation was a straightforward process: After the identification of the data types used in the spreadsheet and their definition in DyWA, first the lowest-level processes and then successively the higher-level processes were defined, until the complete functionality of the original tool was covered. Many of the processes have the potential to be reused by other health care applications, or even in other domains.

Adaptations to changing experimental setups are still possible, while the processes are immediately executable and remain customizable. For example, classification thresholds for a possible what-if analysis is done centrally and uniformly in one process: Currently, a patient is *IN* criterion one, if the BMI is below 20. Should a different threshold value be of interest, e.g. 18, it suffices to edit it in the DyWA. The underlying jABC4 processes are modeled in a way that they do not have to be touched, and the scientists do not have to take care of the processes running in the background that update the classifications of all patients in the database automatically. In the spreadsheet version, the change was necessary for each patient.

This new environment provides in practice a significant step towards the large-scale applicability of a formal model-based and formal methods-supported, model-driven, generative IDE for scientists, contributing to a new kind of engineering of complex computing systems. It enormously simplifies the access and uptake of such capabilities for life science researchers. From their point of view, the IDE ensures that the modeling of domain-specific data types *and* the processes using these components happens in one coherent system, at a user-accessible level. This trait makes it particularly suited for software development efforts in interdisciplinary contexts. The immediate availability, consistency and reproducibility of the outcomes, without need of IT expertise, and the coherence and evolvability of the entire collection of data schema and processes are key assets to these users.

As a current limitation, in the context of privacy and security considerations, access to the data and processes in the web application should be based on a proper roles and rights management, as provided for example in DIME [2], a Cinco-product [15], that includes the user/rights management perspective, integrates further the data and process views, and bases internally on DyWA, thus is compatible with the data model. We are considering a move to those environments for future projects. Ongoing work also addresses provenance tracking and auditing of all the data collected and accessed, in order to know who did what when with which permissions, and to maintain truly complete records of experimental results.

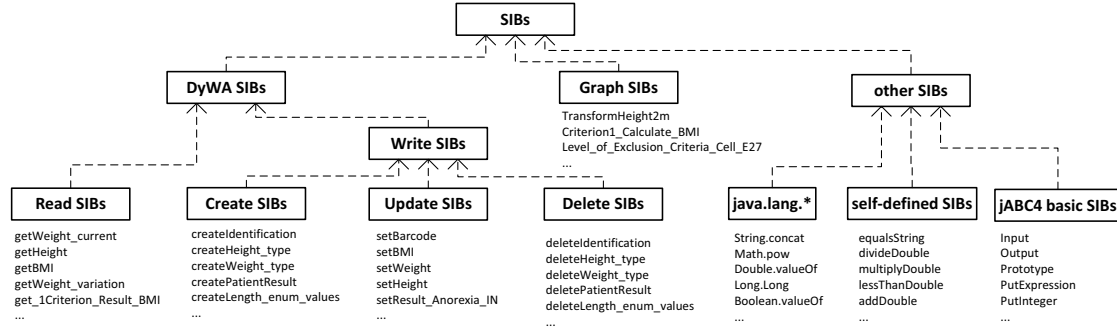
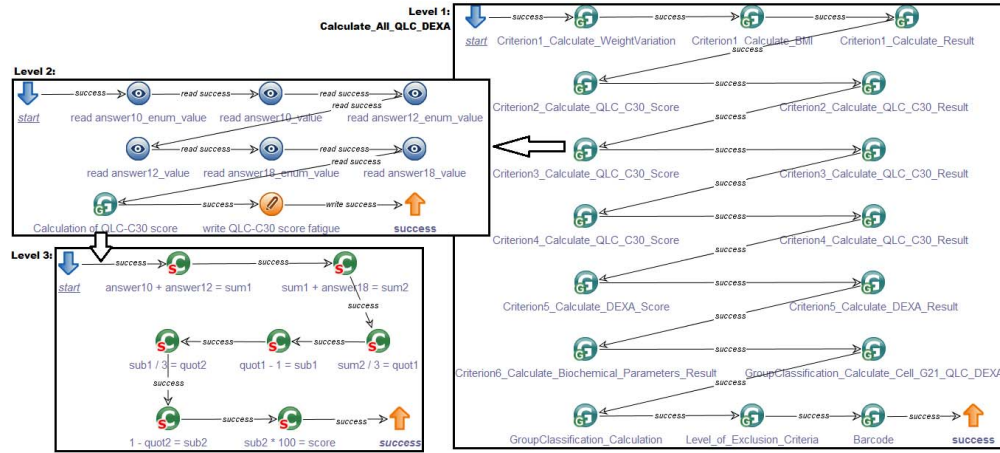


Figure 4: The taxonomy of SIBs.

Figure 5: A possible path in the graph level hierarchy to *Criterion3_Calculate_QLC_C30_Score_Calculation*.

ACKNOWLEDGMENT

This work was supported, in part, with the financial support of the Science Foundation Ireland grant 13/RC/2094.

REFERENCES

- [1] I. Altintas, C. Berkley, E. Jaeger, et al. 2004. Kepler: An Extensible System for Design and Execution of Scientific Workflows. In *SSDBM 2004*. IEEE Computer Society, 21–23.
- [2] S. Boßelmann, M. Frohme, D. Kopetzki, M. Lybecait, S. Naujokat, J. Neubauer, D. Wirkner, P. Zweihoff, and B. Steffen. 2016. DIME: A Programming-Less Modeling Environment for Web Applications. In *Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications*, T. Margaria and B. Steffen (Eds.). Springer Int. Publishing, Cham, 809–832.
- [3] S. Boßelmann, A. Wickert, A.-L. Lamprecht, and T. Margaria. 2017. *Modeling Directly Executable Processes for Healthcare Professionals with XMDD*. Springer Int. Publishing, Cham, 213–232. https://doi.org/10.1007/978-3-319-46412-1_12
- [4] W. Evans, J. Morley, J. Argilés, et al. 2008. Cachexia: A new definition. *Clinical Nutrition* 27, 6 (2008), 793–799.
- [5] F. Froehlich and A. Kent (Eds.). 1995. *The Froehlich/Kent Encyclopedia of Telecommunications*. Vol. 9. Marcel Dekker, INC. <https://books.google.de/books?id=TR8fMQAACAAJ>
- [6] M. Frohme. 2013. *Agile Domänenmodellierung für prozessgesteuerte Webanwendungen*. Bachelor thesis. TU Dortmund.
- [7] ITU-T: Recommendation Q.1203. 1992. Intelligent Network - Global Functional Plane Architecture. (10 1992). <http://www.itu.int/rec/T-REC-Q.1203-199210-S/en>
- [8] F. Lira, B. Antunes, M. Seelaender, and J. Rosa Neto. 2015. The therapeutic potential of exercise to treat cachexia. *Current opinion in supportive and palliative care* 9, 4 (12 2015), 317–324. <https://doi.org/10.1097/SPC.0000000000000170>
- [9] F. Lira, J. Rosa Neto, and M. Seelaender. 2014. Exercise training as treatment in cancer cachexia. *Applied physiology, nutrition, and metabolism = Physiologie appliquee, nutrition et metabolisme* 39, 6 (03 2014), 679–686. <https://doi.org/10.1139/apnm-2013-0554>
- [10] T. Margaria, B. Floyd, A.-L. Lamprecht, et al. 2014. Simple Management of High Assurance Data in Long-lived Interdisciplinary Healthcare Research: A Proposal. In *ISO/LA 2014 (LNCS)*, Vol. 8803. Springer.
- [11] T. Margaria, B. Floyd, and B. Steffen. 2011. IT Simply Works: Simplicity and Embedded Systems Design. In *IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, 2011. 194–199.
- [12] T. Margaria and B. Steffen. 2004. Lightweight coarse-grained coordination: a scalable system-level approach. *Software Tools for Technology Transfer* 5, 2-3 (2004), 107–123. <https://doi.org/10.1007/s10009-003-0119-4>
- [13] T. Margaria and B. Steffen. 2009. Continuous Model-Driven Engineering. *IEEE Computer* 42, 10 (10 2009), 106–109. <https://doi.org/10.1109/MC.2009.315>
- [14] T. Margaria and B. Steffen. 2012. Service-Oriented: Conquering Complexity with XMDD. In *Conquering Complexity*. Springer London, 217–236.
- [15] S. Naujokat, M. Lybecait, D. Kopetzki, and B. Steffen. 2017. CINCO: a simplicity-driven approach to full generation of domain-specific graphical modeling tools. *International Journal on Software Tools for Technology Transfer* (12 05 2017). <https://doi.org/10.1007/s10009-017-0453-6>
- [16] J. Neubauer. 2014. *Higher-Order Process Engineering*. PhD thesis. TU Dortmund. <http://hdl.handle.net/2003/33479>
- [17] J. Neubauer, M. Frohme, B. Steffen, and T. Margaria. 2014. Prototype-Driven Development of Web Applications with DyWA. In *ISO/LA 2014. LNCS*, Vol. 8802. Springer Berlin Heidelberg, 56–72.
- [18] T. Oinn, M. Addis, J. Ferris, et al. 2004. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20, 17 (2004), 3045–3054.
- [19] OpenClinica. 2018. OpenClinica. <https://www.openclinica.com>. (January 2018). <https://www.openclinica.com> [Online; last accessed 20-January-2018].
- [20] M. Seelaender, A. Laviano, S. Busquets, G. Püschel, T. Margaria, and M. Batista. 2015. Inflammation in Cachexia. *Mediators of Inflammation* 2015 (2015), 2 pages. <https://doi.org/10.1155/2015/536954>
- [21] B. Steffen, T. Margaria, R. Nagel, S. Jörges, and C. Kubczak. 2007. Model-Driven Development with the jABC. In *Hardware and Software, Verification and Testing*. LNCS, Vol. 4383. Springer Berlin/Heidelberg, 92–108.