

On the design of enterprise ontology-driven software development

Citation for published version (APA):

Krouwel, M. R. (2023). *On the design of enterprise ontology-driven software development*. [Doctoral Thesis, Maastricht University]. Maastricht University. <https://doi.org/10.26481/dis.20231103mk>

Document status and date:

Published: 01/01/2023

DOI:

[10.26481/dis.20231103mk](https://doi.org/10.26481/dis.20231103mk)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

10.3 Impact

In this section the broader impact of this research will be outlined.

10.3.1 Research Goal and Results

The aim of this research is to work towards a method to generate enterprise software from enterprise models. The idea is that enterprises need to continuously adapt to stay ahead of competition, to keep their customers happy, and in order to comply with changing laws and regulations. Moreover, in order to support ‘enterprise agility’, it was identified that enterprise software should support their end users, should be adaptable, should be created quickly, and that it should be traceable how the end users’ requirements are supported by the software. The only way to answer these needs, is by creating a structured method that guides its users on the path of software development for enterprises. Methods are not new, but due to advancements in technologies and enterprise modeling, this is the moment to create a new method.

In order to guarantee speed and traceability in the software development process, this research adopts an approach where software is fully generated from enterprise models. In order to guarantee that the right – and possibly all – user requirements are supported, DEMO models are used that only describe *what* a user needs and not *how* the user should be supported. As these models tend to leave out specific details, it is much easier to communicate and validate them. These models could even be used as reference models for a particular type of enterprise, e.g., a pizzeria or a bank. In order to prevent hard coding important details in the software, a framework was developed to capture additional design decisions. This framework also enables to explicitly design the required adaptability for enterprise software.

By means of exploratory case studies it has been shown that this approach is indeed feasible and addresses the identified needs. Moreover, four method elements were created that later can be combined into a complete method to generate software from enterprise models. This research thus has laid the foundations to create such a method.

10.3.2 Contributions

The results of this research show that the development of software can (and needs to) be drastically changed in order to support the so much desired enterprise agility. For scientific research it is required that the fields of enterprise modeling and software development are further integrated. A framework was developed

that can be used to bridge the gap from enterprise models to software, and it is suggested that it will be adopted in both scientific research and practical use cases. The practical research has shown that it is possible to use practical cases in a scientific context. Moreover, only by applying scientific research to practical cases, the real value can be validated. Scientific research in this area is encouraged to apply a practice-driven approach.

For enterprises, being public or commercial or some other, this research contributes to the improvement of the software development process, and to achieve enterprise agility in general. More specifically, with the designed approach, more software can be produced in less time, thus providing a way to lower the ever-growing IT budgets. As software can be changed more easily using the designed approach, it also offers new opportunities with a shorter time-to-market for new business innovations. As the approach uses high-level models that can be considered reference models, it becomes fairly easy to create specifically tailored software to similar organizations. Especially for municipalities and for financial, healthcare, and educational organizations, there is a big need to use reference models and standards while accommodating local differences in implementation.

Combining higher-level enterprise models and code generation has one more advantage: When technology advances, the only thing that is needed is to create a new transformation from the enterprise (meta)model to the new technology (metamodel). As soon as that mapping has been devised, all software can easily be regenerated from the enterprise models, without a detailed analysis of what the old legacy systems exactly do or should do.

Research shows that a lot of IT projects fail.³ There are two dominant variables that define project failure or success: expectation (in terms of requirements, timelines and budget) and reality. If the gap is (too) big, a project is considered a failure. This research does not help to plan better for budgets or timelines. It does however help to improve and speed up implementation. This means that as long as plannings are being improved and expectations stay stable, this research can help in improving IT project success rate and save money by reducing project failure.

This research shows that it is possible to automate software development to a large extent, and use models that can typically be created in little time. As a result, less and less software developers are needed. The results of this research could therefore be part of a solution to deal with the tech shortage⁴ as the designed approach enables persons without a background in IT to create software. This approach can therefore be considered an implementation of citizen development that enables and encourages non-IT people to create enterprise applications. The biggest difference between the current implementations of citizen development and this research is that current implementations rely on visualizing software components, whereas this research starts from the business level, i.e., processes, rules,

³See, e.g., a report from the Standish group: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.

⁴See, e.g., <https://www.gartner.com/en/newsroom/press-releases/2021-09-13-gartner-survey-reveals-talent-shortages-as-biggest-barrier-to-emerging-technologies-adoption>.

and information. The designed approach can be considered an implementation of the paradigm of composable applications, where software solutions are being created by combining existing (functional) software components. If the DEMO transaction kind is considered a (functional) component, that is easily converted into working software, creating larger applications becomes nothing more than creating a chain of transaction kinds. This research could help in bringing citizen development and application composition to the next level, lowering the need for highly-skilled software developers.

Shortening the cycle from business idea to working software also has the potential to add value in the field of (Agile) Enterprise Design. By quickly implementing working software, the feedback cycles become shorter, and (potential) design flaws, in terms of business objectives, enterprise ontology, and enterprise implementation, can arise earlier in the process, reducing wasted investments.

10.3.3 Relevance

As the impact of this research is both scientific and practical, it is relevant for both researchers and practitioners. It is relevant for researchers in the area of software development as well as in the area of enterprise modeling and enterprise architecture. For practitioners, at first it may seem only relevant for ‘IT people’. As software is an integral part of an enterprise and should be close to the end users, this research is also relevant for process owners, team leads, etc., that play a role in enterprise improvement and/or enterprise change (management) up to the highest management level. Any employee in an organization can use this research to show that software development can be improved and challenge the IT department to either use this research or come up with a better way. Moreover, tool builders could adopt this approach to further support the automatic creation of software.

The true value of this research can only be found by practical application. It is therefore necessary to apply it to more use cases, in small and big, and in public and commercial organizations. It is necessary to further validate and show the results of this research. Only then may such an approach be adopted by more and more enterprises, and may it become a new standard for software development. Adoption by one of the larger standardization entities and/or governmental agencies could speed up the adoption in practice.