No-Code Application Development Using Ontology Weaver

Christian Vibe Scheller and Pavel Hruby

REA Technology, Copenhagen, Denmark, http://reatechnology.com

1. Introduction

The low-code & no-code platforms contain configuration languages close to the concepts that citizen developers intuitively understand, for example, flowcharts in the case of Wem.io, and pipelines in the case of Quickbase. For several years, the authors of this paper experimented with an application language based on the REA (resources, events agents) ontology [1, 2, 3, 7], which contains concepts intuitively understood by economists. In all these cases the application ontology is static, and can only be changed by a new release of the platform.

At VMBO 2022 workshop at CAiSE'22 we will demonstrate a novel approach, where a citizen developer can adapt the platform's programming model by selecting different ontologies that cover the problem domain. The platform is available at https://nocodeplatform.azurewebsites.net/ .

2. Ontologies for Software Application Development

Software applications can be divided into several broad solution domains [4]:

- Transaction systems. They record events that occurred in the real world and provide analytics over these events. Typical examples are ERP (enterprise resource planning), supply chain systems, financial applications, and systems, in which transactions are not economic.
- Translation systems. They receive input in one form and transform it into another form. Typical examples are compilers, video encoders, image recognition, and language translators.
- Reactive systems. They respond to inputs from their environment, such as the end-users and other systems. Typical examples are computer games, simulators, and microcontrollers in industrial applications.
- Data storage and management systems. They store and retrieve data, such as music and videos. Typical examples are iTunes, YouTube, OneDrive, etc.

Each of these domains can be described by one or more ontologies, that can be used to construct models and declarative configuration languages for these domains. For example, transaction systems can be described by the REA ontology [1, 2, 3] and the POA ontology [8].

In addition to these broad categories of software systems, implementations of application logic patterns enhance the capabilities and behavior of software applications. Examples are Notification, Calendar, Location, Identification, Classification, and Workflow. They can be described by their own "micro-ontologies" and "weaved" into the application's domain ontology [5, 6].

3. Ontology Weaver

A challenge with this approach is its limitation to a single domain ontology. For example, a low-code & no-code platform in the domain of transaction systems is suitable for recording financial and other economic transactions, but unsuitable for designing a speech recognition app or a flight simulator.

Proceedings of the 16th International Workshop on Value Modelling and Business Ontologies (VMBO 2022), held in conjunction with the 34th International Conference on Advanced Information Systems Engineering (CAiSE 2022), June 06–10, 2022, Leuven, Belgium
EMAIL: scheller2@msn.com (C.V. Scheller); phruby@acm.org (P. Hruby)
ORCID: 0000−0002−5502−0880 (P. Hruby)

This problem cannot be solved by using a more general metamodel, because it would eventually turn into a general-purpose programming language; the platform will lose the domain semantics and consequently the benefit of high speed and productivity in application development.

A traditional solution to this problem is to develop the missing functionality by writing code in a general-purpose language. This approach has many drawbacks, such as decreased productivity, and when a new version of the platform is released, the customization code often needs to be rewritten.

To overcome the problem of limitations of a single domain ontology, we developed an alternative approach allowing the developer to choose from a variety of ontologies and add new ontologies on demand, to cover the required problem domain.

In the VMBO 2022 presentation, we will demonstrate how the no-code platform at https://nocodeplatform.azurewebsites.net/ can be used to develop a software application based on multiple ontologies simultaneously.

4. References

[1] G.L. Geerts, and W.E. McCarthy, "The Ontological Foundations of REA Enterprise Information Systems." Annual Meeting of the American Accounting Association, Philadelphia, PA. 2000, URL: https://msu.edu/user/mccarth4/Alabama.doc

[2] G.L. Geerts, and W.E. McCarthy. "An Ontological Analysis of the Primitives of the Extended REA Enterprise Information Architecture". The International Journal of Accounting Information Systems, (March): 1-16, 2002

[3] J. Kiehn: "Object-oriented REA using Dbquity". VMBO 2022.

[4] Neil Harrison, Personal Communication, 2015

[5] P. Hruby, J. and Kiehn, J., C.V. Scheller, "Model-Driven Design Using Business Patterns". Springer, 2006

[6] P. Hruby," Ontology-Based Domain-Driven Design", OOPSLA 05 Workshop on Best Practices for Model-Driven Software Development, San Diego, CA, USA, 2005, URL: http://www.softmetaware.com/oopsla2005/hruby.pdf

[7] G.L. Geerts, and W.E. McCarthy, G. Gal, "Congruent and Meronymic Constellations in the REA Ontology," VMBO 2016, URL: http://www.loa.istc.cnr.it/vmbo2016/wp-content/uploads/2016/02/VMBO2016_paper_15.pdf

[8] C.V. Scheller P. Hruby. "Business Process and Value Delivery Modeling Using Possession, Ownership, and Availability (POA) in Enterprises and Business Networks." Journal of Information Systems: Summer 2016, Vol. 30, No. 2, pp. 5-47.

[9] L. Hammer: "Demonstrating Dbquity – a declarative software platform", VMBO 2022.