

CoreWEB - Semantic Expressions in Conceptual Models for Generation of Information Systems

Tomas Jonsson,¹ Matts Rimfors²

Abstract: Information systems are partially defined by conceptual models. Conceptual models declare structures of concepts which need to be augmented with semantic logic in some form in order to get towards fully functional information systems. CoreWEB is an environment with a one button transformation of conceptual models with expressions declaring semantics, into executing information system. In CoreWEB, semantic logic is added to declared concepts by defining expressions (parameterless functions without side effects). In the executing information system, consistency of data as stated by expressions is maintained by an event driven data-flow mechanism.

Keywords: Model Driven Systems; Information Systems; Conceptual Models; Semantic Models; Model Execution; Data-flow

1 Introduction

Our research and development focus is on seamless life cycle management of Information Systems (IS) which are semantically consistent with their users. As part of this effort, CoreWEB was developed as a tool for education, research and prototyping.

Model driven development and execution of IS has a long history [JE15; Pa04; PG92; Re77]. Automated transformation of IS systems, from requirements and conceptual models, is available from some independent suppliers and has proven to work for full scale enterprise information systems. Documented benefits of automatic transformation of domain models, a category which includes conceptual information systems models, into software systems include lead time, cost and quality in an order of magnitude [MD08].

We consider pure conceptual models as declarations of structures of concepts, whereas semantic models also need definitions of the meaning of each concept. For information systems construction, we consider conceptual models describing structures of data to be suitable. For the purpose of automatic transformation of these models into information systems, formal machine readable definitions of concepts are required. I.e. rules regarding the concept and if the concept is a property, what is the rule for the value of this property.

In many approaches, these kinds of rules are embedded in processing concepts separate from the data concept they affect, such as in object methods, object constraints, processes

¹ Genicore AB, R&D, Östra Hamngatan 52, S-41109 Göteborg, Sweden, tomas@genicore.se

² Genicore AB, R&D, Östra Hamngatan 52, S-41109 Göteborg, Sweden, matts@genicore.se

or transactions. This allows for multiple and conflicting definitions of concept rules. In our demonstrated modelling tool it is only possible to annotate declared data concepts with its logic definitions. Thus, if a data value is incorrect, conflicting definitions can be ruled out and the location of the fault is obvious.

CoreWEB, previously demonstrated [JE17] as an object oriented entity relational conceptual modelling and execution environment, has been extended with the ability to define formal semantics as expressions on object attributes. Expression execution is at runtime controlled by a generated data flow engine, working with two phases in the expression dependency Directed Acyclic Graph (DAG). First phase is a bottom up mark (properties to be recalculated) and second phase is top down DAG traversal to recalculate values. This approach ensure transactional integrity and avoids multiple recalculations of the same value for single data change events.

In order to simplify implementation and providing generality of expression language, expression syntax and semantics is simply javascript, executed by the WEB browser.

2 Calculations in Sales Order Model

To exemplify concept definition using expressions, a model of an *order* is used and the expression of calculating the price of the order as the sum of the price of the order items. In Fig. 1 the object types of the order is shown and their relations. The notation was developed for information modelling and is briefly explained in [JE19] page 10)



Fig. 1: Object types of an order system, where Orders can have relation to many (circle symbol) order items of type Amount and an Amount can have a relationship to one (rectangle symbol) product type.

Fig. 2: Panel for defining expressions, in this case order price as the sum the price of order items.

Fig. 2 shows the panel for expression definition, where properties of orders and related amounts are listed, as well as the *sum* button which conveniently generate the javascript pattern of summing.

3 Calculation for Summing Node Values of a Tree Structure

In this example, we define a tree graph of nodes and relations, with an expression calculating the sum of node values and the consecutive sum of all tree branches. Instead of the recursive tree traversal approach of imperative or functional languages, in the expressional data-flow approach nodes sum is just declared as the sum of its children plus its own value. Node object type and relations is shown in Fig. 3

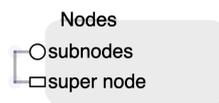


Fig. 3: Node object type and its relations to subnodes and super node

Expression in principle: $Node.treeValue = \text{sum } subnodes.treeValue + nodeValue$.

Javascript pattern is in this case partly generated by selecting the relation (subnodes) and the property to summarise (tree value) and then adding the node value itself. Expression in javascript is shown in Fig. 4 and runtime system in Fig. 5

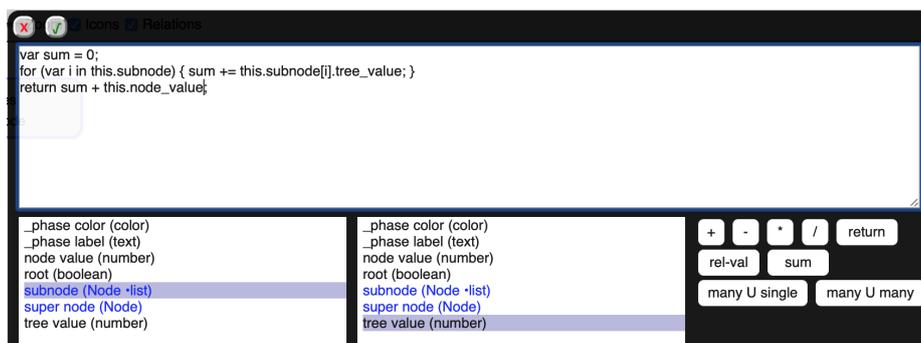


Fig. 4: Panel for defining expressions, in this case sum of tree nodes

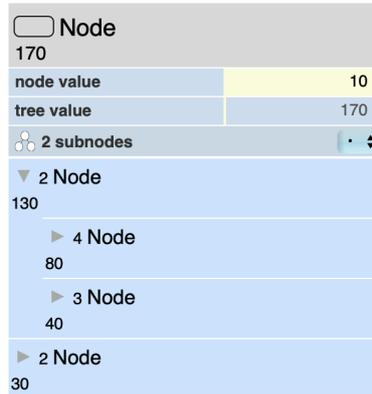


Fig. 5: Partial node tree during execution displaying tree-value by the generated user interface. All nodes have a node value of 10 and there are 17 nodes in total.

4 Concluding remarks - Models as the New Software

CoreWEB is an experimental environment which lacks several features that would be required for commercial use, however it demonstrates the concept of Models as Software. CoreWEB is currently used for teaching, semantic modelling and application prototyping in research, as well as for a tailor made and deployed system for a role play organisation with 600 members. A commercial environment based on similar principles, used to implement an ERP system is described in [JE15].

CoreWEB environment registration and some demonstration videos is available at <https://www.ameis.se/cm1>. Educators and researchers are encouraged to use the free environment CoreWEB to model, execute and experiment with models. We urge researchers and practitioners with interest in model driven IS to join in a triple helix effort <https://www.ameis.se> to develop and disseminate knowledge and practice.

References

- [JE15] Jonsson, T.; Enquist, H.: CoreEAF -A Model Driven Approach to Information Systems. In: CEUR Workshop Proceedings. CAISE2015. Vol. 1367, Stockholm, pp. 137–144, June 2015.
- [JE17] Jonsson, T.; Enquist, H.: Semantic Consistency in Enterprise Models-Through Seamless Modelling and Execution Support. In: CEUR Workshop Proceedings. ER2017. Vol. Vol-1979, Valencia, p356–359, Nov. 2017.

- [JE19] Jonsson, T.; Enquist, H.: Phenomenological Framework for Model Enabled Enterprise Information Systems. In (Welzer, T.; Eder, J.; Podgorelec, V.; Wrembel, R.; Ivanović, M.; Gamper, J.; Morzy, M.; Tzouramanis, T.; Darmont, J.; Kamišalić Latifić, A., eds.): *New Trends in Databases and Information Systems*. Vol. 1064, Springer International Publishing, Cham, pp. 176–187, 2019.
- [MD08] Mohagheghi, P.; Dehlen, V.: Where Is the Proof? - A Review of Experiences from Applying MDE in Industry. In (Schieferdecker, I.; Hartman, A., eds.): *Model Driven Architecture – Foundations and Applications*. Vol. 5095, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 432–443, 2008.
- [Pa04] Pawson, R.: *Naked Objects*, Dublin, Ireland: University of Dublin, Trinity College, June 2004, 223 pp.
- [PG92] Pastor, O.; González, A.: OO-Method: An Object-Oriented Methodology for Software Production. In (Tjoa, A. M.; Ramos, I., eds.): *Database and Expert Systems Applications*. Springer Vienna, Vienna, pp. 121–126, 1992.
- [Re77] Reenskaug, T.: PROKON/PLAN-A Modelling Tool For Project Planning And Control. In: *IFIP Proceedings*. IFIP. Toronto, Canada, pp. 717–722, 1977.