The TEMPORA Approach: Information Systems Development Based on Explicit Business Rules with Time

Benkt Wangler

The TEMPORA Approach: Information Systems Development Based on Explicit Business Rules with Time

Benkt Wangler

Department of Computer and Systems Sciences
Stockholm University, Forum 100, SE-16440 Kista, Sweden
benkt.wangler@dsv.su.se

Abstract. This paper provides an account of the project TEMPORA, run from 1989 to 1994 and sponsored by the European Union, in which the Swedish Institute for Systems Development (SISU) and the research institute SINTEF from Norway were two of the partners. The project aimed at developing a prototypical systems development environment that involved the time dimension and it was based on the explicit representation of business rules. The Nordic partners played important roles in the project, in designing, building, and evaluating modeling formalisms and tools, as well as in designing methodological support.

Keywords: Business rules, information systems development environment, time dimension

1 Background

During the 1980s, IT professionals started to become increasingly aware that the most fundamental part of information systems was the business rules embedded in the software. Such rules are, more or less, all the conscious rules that govern the business. They derive from business objectives (customers have to pay their invoices within ten days), laws and regulations (each car must have a registration number), and laws of nature (a child can only have one genetic mother). These rules are implemented into the various programs of an information system, without much help in keeping track of them. What rules are there? When a rule changes, which code do I have to change? Those questions required answering when a change occurred in a system.

Another trend starting somewhat earlier was the interest in the temporal dimension. That is, how does one take requirements involving time (e.g. a rule such as in order or in one state; to be divorced you had to have been married in some previous state) into consideration during requirements capture and how should databases and information systems be designed with regard to representing and coping with time. The research that was carried out concerned understanding and implementing time in an information system context that resulted in various kinds of temporal logic and in temporal databases usually realized in the relational database model.

TEMPORA [1] was a project within the European ESPRIT program aimed at developing practical methods and tools for systems development that integrated rule-based methods and temporal reasoning. Involving time meant that one could express the requirements for historical information and that rules may refer to time. As indicated in Fig. 1, the TEMPORA development discipline ranged from business analysis all the way through to design and implementation.

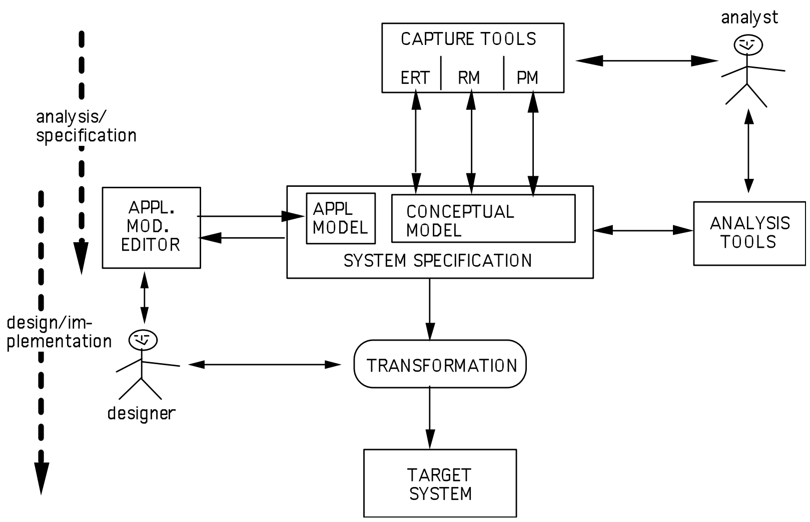

Fig. 1. The TEMPORA systems development environment.

Central to the environment was a conceptual model comprising three parts: a structural entity-relationship-time (ERT) specification, a rule model, and a process model. Around this specification, we developed methods and tools both for capturing requirements and creating/analyzing the model and for taking it further to implementation.

The remainder of the paper briefly presents the project, and then concentrates on the conceptual model and the evaluation of its feasibility. The paper concludes by accounting for some experiences of the project as a whole.

2 The Project

As mentioned, TEMPORA was a European research project running under the ESPRIT 2nd framework program in which the Swedish Institute for Systems Development (SISU) and SINTEF, a Norwegian research institute, were two of the partners. Since Sweden and Norway were not members of EU, national authorities funded the work. The other partners were:

- the Belgian company BIM (cf. BIM-Prolog), which was formally responsible vis-à-vis the European Union (EU),
- the University of Liege in Belgium,
- the University of Manchester Institute of Science and Technology (UMIST, now part of the University of Manchester), which acted as project leader,
- Imperial College, London,
- Logic Programming Associates (LPA), a London based company, and
- Hitec, a company from Greece.

The author of this paper led the SISU effort. The project had a considerable budget and lasted for just over five years, from January 1989 to the beginning of 1994.

The responsibility of SISU was to design and specify the semantics of the ERT notation, to develop the drawing tools for ERT and the process model, and to contribute to the parts of the methodology that concerned early phases of requirements modeling. The Swedes were also responsible for evaluating the usability of all three notations.

The Norwegian partner shared many of these responsibilities, also bringing and adapting the PID formalism into the project. Both Nordic partners made substantial contributions to the project. In general, universities and research institutes conducted the more research-oriented tasks, while industrial partners were providers of technology.

There were regular plenary meetings about four times a year at locations offered by the various partners, as well as, at times, extra meetings between some of the partners to solve specific issues. Once a year, one of the plenary meetings was used a whole day for review and discussion with a representative from the European Commission and an independent reviewer.

3 What We Did

The idea of TEMPORA was to build a systems development environment based on the explicit representation of business rules in notations that were easy to use and understand, and which should also make it possible to represent requirements that involved time. The environment should provide support in devising system requirements expressed in these notations and from which an executable prototypical system could be derived. Hence, we proposed the following:

- A conceptual schema representation comprising a triplet of interrelated notations:
  - Process interaction diagrams (PID), a graphical process modeling language similar to dataflow diagrams but enhanced with triggers,
  - ERT, an entity-relationship (ER) type of language where one could mark for which entity and relationship types one wanted with its temporal behavior to be preserved,
  - External rule language (ERL), an elaborate, though still reasonably easy to understand and use, textual version of a temporal logic

- Tools for use in formulating statements in these languages and in analyzing the resulting conceptual model. The tools were implemented using RAMATIC, a meta-case platform developed at SISU.
- A generator that could take a model built in these languages as input and transforms it into an executable system. The generator and resulting system were implemented using BIM-Prolog and a relational database system (SYBASE).
- Methodware, rather elaborate procedures, guidelines, and advice regarding how the analyst/designer should think, reason, and use the above languages and tools during the process of capturing and formulating a model in those languages and, through such use, finally devise executable software that fulfill the requirements represented in the model.

In the following two sections, I briefly present and provide examples of the representation of time, and the business rule notation.

3.1 The Temporal Dimension

Semantic data models of the past focused on data representation without any in-depth consideration of the temporal and behavioral aspects of information modeling. In [2] we explored the enhancements to a binary entity-relationship model necessary to express requirements for temporally related information. We employed a time model in which time proceeds in ticks. Hence, central to this model were time points represented by ticks and periods starting at a certain tick and ending at a later one. Periods corresponding to those of our own calendar down to a sufficiently narrow granularity were also included.

When adding the temporal dimension to models belonging to the ER family, we must consider

- Enhancing the relational database to incorporate temporal information, and
- Augmenting the graphical and textual modeling formalisms with time concepts, so that temporal requirements, integrity constraints, and other rules involving time can be specified.

The topic of adding the temporal dimension to the relational model had received considerable attention in the database literature. It was made clear that two types of temporal information can be recorded in addition to attributes/tuples, to fully capture the evolution of an attribute/tuple over time, these being the

- Event time which records, as a series of time points, the time period over which we know (or think) a piece of information holds in the Universe of Discourse (UoD), and
- Transaction time which records, as a series of time points, the time period during which the information is stored in the database.

In practice, the event time is of most interest, since the modeling of the UoD is usually the objective of an IS. The transaction time is only of use in meta-rules which review the activities of the IS and that would need to know what information the system used at any point in its execution. Since the latter was not the purpose of TEMPORA, the temporally extended ER type of model developed within TEMPORA incorporated the event time and not transaction time. It did this in the form of time-stamps (start- and end-time) attached to the database representations of entities and relationships for which one wanted to keep track of history.

Hence, the ERT language was augmented with the possibility to T-mark the entity types and relationships for which one wished to keep track of the period of validity, that is, the period for which they hold. Fig. 2 shows an example of an ERT model. The T-mark on certain entity classes and relationships expresses that instances of these are valid (hold) only for a limited period, as represented in the database. Instances of classes that have no T-mark hold for all time, as in a standard database system. A “car” has a certain life span from production to destruction. During this time, it may have several registration numbers and several owners, each with a certain life span. In addition, a person may have a certain life span. As we delete instances of classes that are not T-marked, they are erased from the database. Instances of classes and relationships that are T-marked will stay in the database after they are “killed,” but they will not hold as true any more. The dashed lines indicate relationships derivable from other relationships.

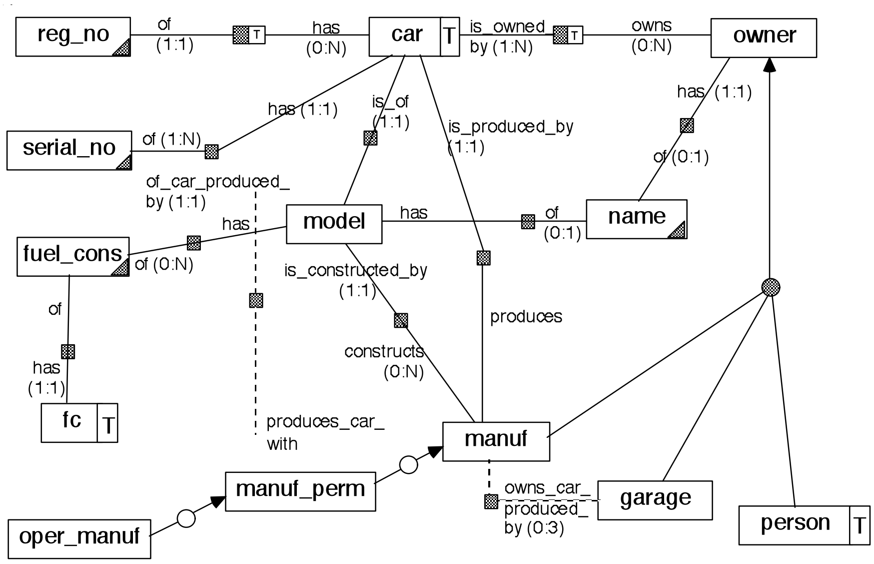

Fig. 2. The ISO-case ERT schema.

With reference to this model, rules referring to time may be expressed by means of temporal logic, which is a modal logic able to express that some condition should hold “some time in the past,” “in the next state,” “always in the future,” and so on. In addition to such temporal relations, the ERL language was augmented with various logical and temporal operators and connectors. Examples appear in the next section.

3.2 Business Rules

Business rules are essentially of three kinds [3]:

- Constraints expressing restrictions on permissible states of the information base,
- Event-condition-action rules expressing that if a certain event occurs, and provided a pre-condition holds, a certain action should be conducted, i.e., the database management system should somehow ensure that a certain post-condition is made to become true,
- Derivation rules expressing how certain information, which is not explicitly stored in the database, may be derived from information stored in the database.

In addition, a rule may be *static*, referring to a single state of the database, or *dynamic*, referring to several states. Cardinality constraints of relationships such as the ones expressed in Fig. 2 are examples of static constraints.

To be expressed as precisely referring to explicit states of the database, the rules must be presented in great detail. A few examples of static and dynamic rules expressed in ERL and referring to Fig. 2 follow below.

a) Sample static constraint: "The serial number is unique for all cars of one manufacturer." In ERL, this rule may be stated as: "if at some point there are observations that there are cars produced by the same manufacturer and these observations do not concern the same car, then their serial numbers cannot be the same."

```
If Manufacturer.M produces car.C1 and manufacturer.M
produces car.C2 and not C1 = C2 then not car.C1 has
serial.no = car C2 has serial.no
```

b) Sample dynamic constraint: "Each car has one serial no for its entire lifetime." `always_in_future` is a temporal operator referring to all future states in the lifetime of the car.

```
If car.C has serial.no = X then always_in_future (car.C
has serial.no = X)
```

c) Sample dynamic ECA-rule: "Only in January may a car be registered as being produced in the previous year." The rule may be formulated as below, where registration time is the start time of the "car has reg_no" relationship. Some_time_in_past is a temporal operator referring to all states preceding the current one. This_year, this_month and last_year are temporal literals referring to the corresponding periods.

```
When car.C has reg_no if some_time_in_past(car.C and
just_before not car.C and Prod_year = this_year) then
Prod_year = this_year or this_month = 'January' and
Prod_year = last_year
```

In the communication with the customer/user, rules are often expressed ambiguously. By stating the rules in ERL, formulations are demanded that in a precise way refer to the temporal database, although they sometimes have to be explained to those not used to thinking in temporal database and ERL terms.

3.3 Evaluation

The TEMPORA approach was tested in parts and in total, using a number of smaller and larger cases. One of the first cases we used was the so-called ISO case, involving forty-seven more or less complex statements and rules governing car registration in a fictitious country (cf. Fig. 2 and rule examples of Section 3.2 and [3]). It transpired that it was indeed possible to model all the rules, but that some of them became complex and actually required a great deal of "Prolog programmer thinking."

The largest case used for testing the modeling environment concerned building a complete model of the order system at Sweden Post. This case generated several hundred business rules at different levels of abstraction, some of which were quite complex. Although the case again showed that it was possible to model all the rules, it also demonstrated that keeping track of such a huge mass would require a much more elaborate and efficient way of structuring and handling them in the model repository.

The case that was used to test the complete environment was much smaller, but demonstrated that it was indeed possible to derive a working system with functioning transactions, albeit the system was very small and the user interface primitive.

4 Concluding Remarks

Ultimately, we did not use many results directly, but the thinking and the understanding of business rules affected other work on business modeling at SISU. Above all, the project was a great learning exercise generating much useful knowledge and experience per se. The TEMPORA approach itself might have resulted in a realistically useful environment, if we had another fifty person-years to build it. The project was also a tremendous exercise in working together as an international team. The Nordic partners worked quite closely together on the modeling formalisms and the methodology, sometimes meeting in Trondheim or Stockholm. In addition, email proved to be a really useful tool for communicating. It is indeed a challenge to write down your thoughts in such a way that makes them possible for others to grasp.

Additionally, we further advanced the knowledge on information modeling, business rules, and representation of time in new ESPRIT projects such as F3, Nature, and Ores, in which SISU took part. F3, which means From Fuzzy to Formal, was a project that focused on the early stages of requirements capture, representation, and management. Nature (Novel Approaches to Theories Underlying Requirements Engineering) intended, as the name indicates, to strengthen the foundations of requirements engineering. Finally, Ores was a project that built a temporal database manager and tested it on electronic medical record applications for a Spanish hospital.

The OMG Business Motivation Model [4] is largely formulated along the same line of ideas that were put forward in TEMPORA. Business rules are the rules governing the business and, as such, they are important assets of the business. They are derived from a means-end breakdown of business objectives, from man-made laws and regulations, or even from laws of nature. However, in order to be incorporated into computerized information systems, they need to be broken down into detail and stated very precisely, with regard to the database

The representation of time in requirements and in databases was subject to quite a lot of research in the 1980s and early 1990s. It seems, however, that the interest for general solutions to problems of the temporal dimension has decreased lately. One can wonder why. Perhaps it is too hard conceptually and technically, or it may be that the time for generic and usable solutions has not yet arrived. Nevertheless, a temporal extension to SQL, TSQL2 [5], has been developed and there are at least a couple of relational DBMS implementations that accept statements in this language.

In general, the project partners got along very well, although a few did not contribute too much to the project, probably because they were not allocated clear enough tasks. Having participated in several international projects of this size, it is the author's experience that it is very important for each of the partners to be assigned work that matches their competence and interest. A controversial issue in the project concerned the rule language and its implementation. This was resolved in a way that perhaps made one of the partners less pleased. However, as a whole, the project was a very valuable experience that provided much useful knowledge that transferred into education programs at the related universities. It also resulted in many lasting relationships and continued collaborative research between researchers in Norway and Sweden.

Acknowledgments. The author wishes to thank all the members of the TEMPORA team, in particular Rolf Wohed (SISU), Anne Helga Seltveit (SINTEF) and Peter McBrien (Imperial College).

References

1. TEMPORAConsortium: TEMPORA: Integrating Database Technology, Rule-Based Systems and Temporal Reasoning for Effective Software. ESPRIT Proj. E2469 (1988)
2. McBrien P., Seltveit A-H., Wangler B.: An Entity-Relationship Model Extended to Describe Historical Information. In: Proceedings of International Conference on Information Systems and Management of Data – CISMOD'92 Bangalore, pp. 244–260, Indian National Scientific Documentation Centre (1992)
3. Wangler B.: A Conceptual Schema with Time: Modelling the ISO-Case with TEMPORA, In: Jaakkola H., Kangassalo H., Kitahashi T., Markus A. (eds.) Information Modelling and Databases V, pp. 313–333, IOS Press (1994)
4. Business Rules Group: The Business Motivation Model: Business Governance in a Volatile world (1.3). Available at http://www.businessrulesgroup.org/bmm.shtml (2010)
5. Wikipedia: Temporal Database, available at http://en.wikipedia.org/wiki/Temporal_database, Last visited 2010 June 11.