Towards a REA policy specification language

Hans Weigand[1], Paul Johannesson[2], Birger Andersson[2]

[1] Tilburg University, P.O.Box 90153,
5000 LE Tilburg, The Netherlands
{H.Weigand}@uvt.nl
[2] Stockholm University
Department of Computer and Systems Sciences, Sweden
{pajo,ba}@dsv.su.se

Abstract. Although the REA ontology includes the notion of policy, a REA policy specification language does not exist yet. The goal of this paper is to explore the requirements for such a language and evaluate the potential of rule language described in Van der Aalst' conceptual model of online auditing.

Keywords: REA business ontology, Coloured Petri Net, online auditing

1 Introduction

The Resource-Event-Agent (REA) ontology was first formulated in [6] and has been developed further, e.g. in [3,5]. It has been shown recently that REA structures also provide a basis for auditing [7]. The REA axioms express fundamental integrity constraints that can be used for both the design of control mechanisms (preventive) and for the detection of deviating behavior (detective) that may indicate errors or fraud.

In the original REA paper, REA was introduced as an accounting framework for use in a shared data environment and developed on top of Entity Relationship modeling, so with a clear focus on database design. However, events play a key role as atomic data units and so a dynamic interpretation is quite natural. In a previous paper, we have shown how REA models can be specified as Petri Nets[9]. However, this interpretation did not include REA policies. The objective of our research is to find a REA policy specification language that (a) is in line with existing proposals in the accounting domain, notably the proposal of Van der Aalst 2009 [2], and (b) can be included in the Coloured Petri Net specification in the form of guard expressions. In this paper, we only explore the requirements for such language and the possibilities. In section 3, we discuss previous work on REA policies, and in section 4, a comparison is made between the REA ontology and the conceptual model (of online auditing) of Van der Aalst. In section 5, we end with some discussion items.

2. REA Policy language

The Business Rules Group (2000) defines a policy as ‘‘a general statement of direction for an enterprise.’’ [3] define a ‘‘policy’’ as a “description of economic phenomena that could, should, or must occur”. Geerts and McCarthy distinguish among the following three types of policy definitions: knowledge-intensive descriptions, validation rules, and target descriptions.

- A *knowledge-intensive description* defines characteristics of a concept that apply to a group of objects. Such characteristics can take the form of a policy definition: e.g., “the price of product X is $5’’
- A *validation rule* represents permissible values, typically for preventive controls. For instance, the payment amount must be less than $1000, unless it is authorized.
- *Target descriptions* provide benchmarks regarding economic phenomena, and they can take at least two different forms: standards and budgets. Generally, *standards* are specifications to be followed, for example: ‘‘How much raw material does it take to manufacture a bike?’’ *Budgets* are described as quantified performance measures most often related to a specific time period such as ‘‘How many cars do we expect to sell in the second quarter of 2006?’’

In their article, Geerts & McCarthy have introduced a policy layer and typification to make policy descriptions possible. However, they do not provide a policy specification language. Taking the same classification as a starting point, the REA Petri Net formalization offers the following possibilities to implement these policies:

Target descriptions are annotated to the value chain, as sketched in the previous paper[9]. The coefficients [4] state that for the production of X liter of beer, you need Y liter of water etc. This kind of data is based on benchmarks and historical data. Although based on historical data, they coefficients have a normative value. They can be used in Material Planning, and auditors can use them as equations that relate the physical flows in the enterprise. Material deviations can occur, depending on the kind of process, but should be explained. In REA, these equations are a quantification of the duality constraints. Therefore, we call them duality equations. In a Petri Net model, the equations can be coded as CPN guard expressions. Two kinds of usage can be distinguished. The guard expressions can be used in *simulation* to execute the effect (units consumed, units produced) of events. In an *operational* Petri Net based system, actual values may deviate from the norms. The guard is not restrictive but takes a role of producing an expected inflow/outflow. Hence, deviations are immediately visible. These deviations can be used in the validation rules, in such a way, that too large deviations are not permitted and the transaction is blocked.

Budgets should not be viewed as isolated objects but as part of the management cycle [8] and influencing managerial action. Managers, on different levels, formulate goals and from there derive interventions such as hiring personnel. For this reason, we do not try to specify budgets in the REA Petri Net that we have so far. However, we foresee a

situation where an operational REA Petri Net is integrated in a simulation environment that allows the user to specify goals and that can analyze the consequences (feasibility) of these goals using the Petri Net simulation.

Validation rules can be translated into CPN guard expressions, but some remarks are due. Validation rules that apply to data values, e.g. the date field must be filled in, are considered not to be in the scope of REA as a business ontology that focuses on economic events. Most relevant validation rules refer to *authorizations*, so the natural place is to include them in the agency layer [ref] . Their specification is based on authorizations as token objects and guard expressions that inspect these authorizations. Basically, the structure that we propose is that the guard includes the call of an *authorized* function that checks the rules such as they are specified in the authorization token.

Knowledge-intensive descriptions characterize a group of objects. This category of policy requires more discussion. What is included here and what not? Is any group description a policy? Assuming that we want to keep the REA economic perspective, only those descriptions should be included that have an economic aspect. Price is an obvious example, However, price is not limited to groups, but can be specific for one transaction. There are list prices and actual prices, and prices can change. So price is not just an attribute, but should be seen as part of a statement, e.g. a quotation (a commitment).

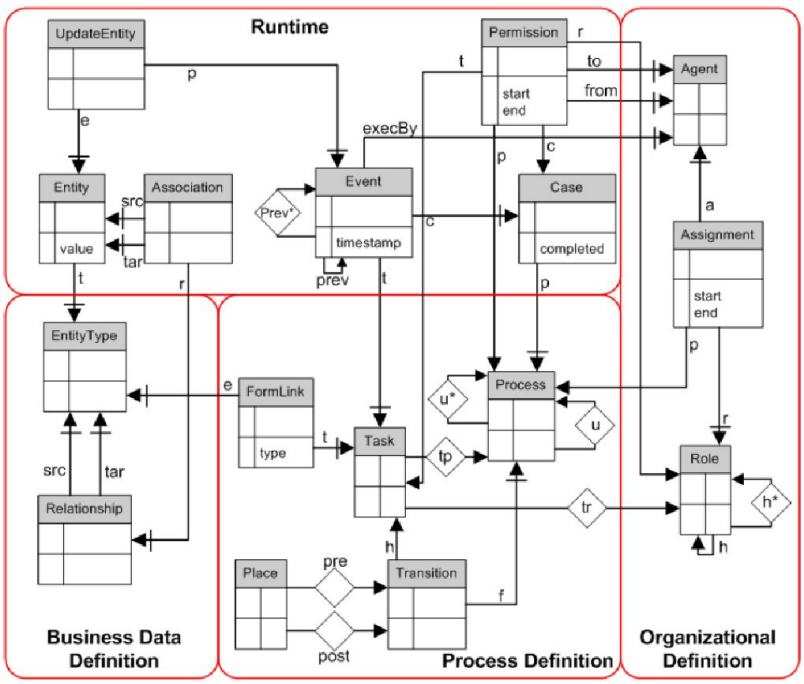

Fig 1 Conceptual model of online auditing (Van der Aalst [2])

3 An online auditing conceptual model

One purpose of policy specification is to ensure that an enterprise operates according to rules and policies. From an auditing perspective, such a control structure is required in order to generate reliable information about the enterprise. Auditors typically check which internal control mechanisms have been defined and whether they are effectively in place. For that reason, it is interesting to look at earlier work on policy specification languages in auditing, more in particular, the work of Van der Aalst [1,2] on online auditing. This article presents a conceptual model of business processes and then identifies a couple of auditing patterns. The conceptual model is redrawn in fig. 1.

We compare this model with the REA ontology. At a first glance, it is easy to recognize the REA concepts agent (in the organizational definition), event (run time) and resource (entity in run time), but there are some subtle differences and it is also clear that the conceptual model contains a lot of information not covered by core REA, in particular the process definition part. In the organizational definition, a distinction is made between agents (e.g. Ron Ross) and roles (e.g. sales clerk), corresponding to agents and agent types in REA, and there is a hierarchy relationship (h) corresponding to the responsibility relationship in REA. Agents execute events. Events and entities are related via the UpdateEntity relationship. We note a small deviation from REA: the UpdateEntity logs represent the stockflow events in REA, and Event is something of a bigger granularity. It could be a duality in REA, but clearly the Event does not have duality semantics. The concepts in Business Data Definition and Process Definition are all on the REA policy (type) level. The conceptual model has a clear bias towards control flow specification which is absent from the REA ontology. Roughly spoken, this lack is compensated in REA by the notions of commitments and the duality relationships that also induce a certain ordering of events, but not on the basis of control flow but on intrinsic reasons. Finally, it must be observed that the conceptual model contains permissions in the run time model that give agents permission to perform a task, for some period of time. Authorizations can be specified in REA at the policy level (although so far it is not made explicit how this is done); but here they are at the run time level. However, note that this permission does not cover all authorizations; an agent is also authorized to execute an action if he is assigned to the right role (his role is related to the task – type level). The permission is for specific agents.

Summarizing, we can say that REA models can be mapped to the conceptual model (also commitments, they can also be mapped to entities) although some of the semantics is lost. On the other hand, the conceptual model contains a couple of concepts that cannot be represented directly in REA – in particular, the control flow specification and run-time permissions.

The conceptual model defines a business rule language. Using first-order logic, the business rule language uses predicates for concepts. First-order logic is very expressive, but from an auditing perspective, not all expressiveness is needed. Van

der Aalst [2] identifies a couple of patterns, such as the 4eyes principle, a limit on the update value (and more specifically, an approval limit), and task precedence.

4. Issues and Discussion points

There are a couple of issues that we would like to discuss at the VMBO workshop, such as:

- Is it useful to have a policy specification language as described above, and if so, why isn't that included in the REA business ontology already?
- Do "knowledge-intensive descriptions" belong to a REA model? What is the criterion?
- Should authorizations be recognized in REA as properties of the Agent/Principal relationship?
- Price is an important accounting concept. How can the various notions of price be accommodated in REA?
- Principles vs procedures. What level or levels of abstraction should a policy language take? For instance, a task precedence (delivery only after payment) is a rather low-level description that implements a higher goal of payment risk mitigation, that in itself can be seen as way of achieving ideal economic duality. What are the basic policy principles in REA?

References

[1] Aalst, W. van der, Hee, K. van, Werf, J. van der, Verdonk, M.: Auditing 2.0: Using process mining to support tomorrow's auditor. Computer; 43(3): 90-93 (2010)
[2] Aalst, W., van Hee, K., van der Werf, J.M., Kumar, A., Verdonk, M.: Conceptual model for on line auditing. Decision Support Systems 50, 636–647 (2011)
[3] Geerts, G.L., McCarthy, W.E.: An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. Int. Journal of Accounting Information Systems 3: 1-16 (2002)
[4] Griffioen, P. Type inference for linear algebra with units of measurement. Technical report, Centrum Wiskunde en Informatica (CWI), Amsterdam (2013)
[5] Hruby P. Model-driven design of software applications with business patterns. Springer (2006).
[6] McCarthy W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review: 544-577 (1982)
[7 Weigand, H., Elsas, Ph.: Model-based Auditing Using REA. Int. Journal of Accounting Information Systems (2012)
[8] Weigand, H, et al.: Management services - a framework for design. Proc. CAiSE 2011, LNCS pp. 582-596 (2011)
[9] Weigand, H., Johannesson, P., Andersson, B. A Petri Net Specification of the REA Business Ontology. CAiSE Forum 2016: 49-56