Modeling the responsibility relationship in the REA Business Ontology using Petri Nets

Hans Weigand\textsuperscript{1}, Paul Johannesson\textsuperscript{2}, Birger Andersson\textsuperscript{2}

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

Abstract. In this paper, we adopt a Petri Net specification of REA for ontology-based process analysis. The focus is on the representation of the responsibility relationship. The Petri Net specification makes it possible to express more than the traditional static specification.

1 Introduction

The Resource-Event-Agent (REA) ontology was first formulated in 1982 by McCarthy and developed further in subsequent papers. It has been shown recently that REA structures also provide a basis for auditing. 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 [McCarthy,1982], 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 this paper, we analyze how the REA responsibility relationship can be specified with Petri Nets.

Definition Petri Net

A Petri Net is a triple (P,T,F) where P is a finite set of places, T is a finite set of transitions and F \( \subseteq \) (P×T) \( \cup \) (T×P) is a set of arcs (flow relation). The state or marking M of a Petri Net is a distribution of tokens over places, that is, \( M \in P \rightarrow N \). We can write M as a vector. A Petri Net state changes by the firing of enabled transitions, which basically means that tokens from the input places of a transition are removed and tokens are added in the output places. The transitions F can be expressed as a matrix that maps a state \( (m_1, \ldots, m_n) \) to a state \( (m'_1, \ldots, m'_n) \).

In a Coloured Petri Net (P,T,F,\Sigma,C), the tokens are of a certain colour (token type) from the colour set \( \Sigma \). A colour has a domain of values. The values can be composite, like a data record, where each field \( i \) is of a domain D. In the following, we assume...
that some of the fields represent a quantitative metric; these are called valuation functions. \( C \) is a function that gives the colour of a place, that is, \( C \in P \rightarrow \Sigma \). The transition details can depend on the token properties and are typically represented by means of arc inscriptions and guard expressions.

1.1 A Petri Net specification of REA

Let us start with a REA conversion process. The example in Fig. 1 shows two decrement events (outflow) for two kinds of raw material (resource). It contains one produce event (inflow) that produces the finished product (resource). The duality (white diamond) links N outflow to M inflow events. Each event has a providing and receiving agent.

![Fig. 1. Sample model for a REA conversion process](image)

There are two ways in which this process can be mapped to Petri Nets. It depends on whether we emphasize the stock flows or the processes. The most compact mapping that highlights the process is given in Fig. 2a.

![Fig. 2. (a) Compact Petri Net representation (b) Alternative representation](image)

Once there is a token (black dot) in place1 and place2, the transition can fire and an event is executed. The tokens in place1 and place2 are consumed and a token in place \( P \) is produced. The two consumptions in the Petri Net correspond with the outflow events in the REA model, and the token production corresponds with the inflow event. The transition itself corresponds with the duality in REA. The place
corresponds to a REA agent, that is, the organizational role controlling the resource instances. This is in line with the definition of REA agent as “some entity transferring or receiving the control of economic resources”; this is what places do with respect to tokens. We can say that a token T in a place P corresponds to agent A (P) controlling resource R (T). The resource type is the colour of the place. For labeling the places, we can use the resource type/colour or the controlling agent, or both (as a convention, we put the resource in square brackets then). In REA practice, the name of the agent role is often derived from the resource he is controlling, so then the resource type label is sufficient. The allocation of REA agents to subjects in the organization is not included in the model as such, as it is also not in the REA ontology. If needed – for instance, for analyzing Segregation of Duties – we can annotate the model with agent allocation in the form of a function P \rightarrow Subjects.

In Fig. 1 there are four agents, whereas there are only three in the compact Petri Net (Fig. 2a). The Production Manager receiving the raw material and providing the product is not included because he is only an intermediary. The role is represented by a place in Fig. 2b – this representation corresponds directly to the event template of [Laurier&Poels]. In this case, the transitions correspond to stock flows, whereas in Fig. 2a, the transition corresponds to the conversion process. Formally, the two are equivalent in the sense that one can be mapped to the other. In the following, we will use the compact process-oriented version of Fig. 2a. In our interpretation, the intermediary role gets a place at the transaction level (not included in this paper).

2 Responsibility

Responsibility is defined in REA as “a static hierarchical reporting and assignment relationship between agents”. The concept plays a minor role in the REA literature. Still, it is quite essential from an accounting ontology point of view. The basic goal of accounting systems and auditing is to provide reliable information about the activities and the use of resources. Ultimately, this information is relevant for the owners of the resources. Companies typically use a vertical delegation system where information moves from lower levels to higher levels in more and more aggregated form.

Delegation is usually described as a relationship between a principal and an agent where the principal gives the agent some task and provides him with resources. To avoid opportunistic behavior of the agent and, in general, to optimize the value creation of the task, there is a need for monitoring. On the basis of signals that the principal receives he can provide compensation to the agent and execute managerial control. In this paper, we focus on the control. The compensation of the agent is represented in REA as an exchange transaction between the company and the employee who provides his human resources.

A principal within the organization needs to monitor his agents for managerial control. This control is not a private goal but something he is supposed to do to fill in his own responsibility to the next echelon, because delegation of tasks does not take away his own responsibility.

According to [Selznick,1948], “delegation is the primordial organizational act, a precarious venture which requires the continuous elaboration of formal mechanisms of coordination and control”. According to Ijiri (1975), cited in [McCarthy, 1984], “In
general, an entity’s power to control resources is provided by someone else, who in turn demands that the entity account for the resources under its control. Therefore accountability and control may be regarded as two sides of the same coin”. According to [Grossi et al, 2004], there are two basic principles: “Organizations structure should see to it that each agent is always aware of its duties (an “ought” implies “know” principle) and organization structure should see to it that its objectives are met (“successful performance” principle).” But: “Successful performance cannot be guaranteed when agents are autonomous and subject to failure”. By consequence, there is an obligation to monitor the performance, as well as an obligation on the side of the agent to inform.

The above-mentioned principles of delegation are captured by the following rules:

R1 For each participation of an agent in an event, the agent must have an authorization from another agent (= principal)
R2 For each participation of the agent in an event, the principal must receive a report, eventually. The report discharges the accountability of the agent
R3 Reporting must be based on records (primary observations) stemming from the participation in the event.

R1 and R2 follow directly from the above. We have formulated them in general wordings. There is much more to say about how the principal receives the recording, for instance, and how this procedure ensures reliability and efficiency. However, we consider these as internal control concerns that are not critical for the definition. They are the means to ensure the goal of R2.

We added R3 in order to make the link between the event and the report more explicit. The principal is typically NOT participating in the event so the report is not provided to the principal in the event directly. Reporting is an independent event. However, the production of the report must have some input. We can use the notion of “signal” as it is used in agency theory for this link, taken in a very general sense. The event must produce a signal: a self-recording by the agent, or recording by another agent (e.g. the counter agent in the transactions of that agent), or an independent observer, e.g. an automated sensor system. The “must produce” is normative. We assume that every event produces signals, but the organization must see to it that some signal is taken up. The signal, or a combination of signals, is the basis for the reporting. In Ijiri’s accountability framework, we talk about records rather than signals. One of the tasks of auditor is to check the reliability of the recording infrastructure and the way the records are being transformed into reports.

The axioms refer to concepts not included in the REA ontology yet, in particular the concepts “authorization”, “record” and “report”. Following [Weigand et al, 2012] we conceive of these as intentional resources, and the corresponding events of authorizing and reporting as intentional events. Characteristic of intentional events is that they are speech acts with a provider and a receiver (Speaker and Addressee) and that they refer to other (typically economic) events. We can take over the event ontology of REA for economic resources (events, resources, agents), but now for intentional resources. Note that these are typically NOT regarded as economic resources, and we do not require economic duality between these events. However, they are very relevant for the accounting information system. In fact, the accounting
information system is a collection of intentional resources. In REA, authorizations are “policies” [Geerts, 2002]. In our conceptualization, policies are intentional resources.

**Fig. 3** Basic event ontology extended with intentional resources

Fig. 3 depicts the adapted ontology. The generic event ontology based on agents, resources and events, is shown in the top. Economic events (REA events) are events that deal with inflow or outflow of economic resources. Intentional events deal with intentional resources. The agent controls a resource in the context of an accountability relationship towards a principal. Reports account for this accountability (or one could say, they discharge accountability). The different roles and rules can be defined in terms of this ontology. It is the principal role who must provide authorizations to the agent and receive reports. It is the agent role who controls the resources. The observer is not directly involved in the accountability relationship, but is the one who provides the records on which the report is based.

### 3 Agent cycle

On the value cycle level, an REA event consumes or produces resources. However, the same event is also the performance of a delegated task. Next to the horizontal process in the value chain, there is a vertical process cycle between principal and agent.

To integrate this vertical process into the Petri Net, we model it as a subnet; in this case, the subnet is a decomposition of the economic agent (so one Petri Net place is
blown up into one agent cycle). In this agent cycle, authorizing and recording are explicit events. As we distinguish between the delegate/agent place and the principal place, the static REA responsibility relationship can be derived from this model. What is the object of authorizing and recording? It cannot be the REA event as such as it involves at least two agents. For that reason, we distinguish two micro-events - these events are not REA economic events but related to them and implicitly assumed by the REA ontology. A REA event is a stock-flow event that can only be realized by one agent providing and one agent receiving (following REA terminology). So two key micro-events that can be located within each agent are the receiving and providing. In between the receiving and providing, there is a task of stock keeping (this stock keeping is not an event but a state).

Fig. 4. Generic agent cycle

In Fig. 4 the agent cycle is modeled as a Petri Net. The principal gives an authorization (a REA policy) to the delegate (agent) who if he accepts it becomes accountable for the occurrence of the event. The agent has to perform according to it when participating in the underlying stock-flow event. The execution produces records. The report is returned to the principal, where it is typically aggregated and evaluated. The cycle is related via inflow and outflow to “horizontal” stock-flow events in the value chain. It can also be related to the contract between principal and agent: this contract is the input for the authorizing.

Authorizations are often implicit but can also be explicit in the form of an authorization table. One goal of internal control is to check whether events have actually been executed in accordance with the authorizations. The policy/authorization has a positive side, the control that is delegated to the agent with respect to some target, and a negative side, the restrictions imposed on the agent. REA does not include a policy specification language (although suggestions have been made). We don’t want to propose one in this paper. If required, policy specifications (tokens
of the colour policy/authorization) can be defined in such a way that we get a full policy language. A guard expression of the “receive” and “provide” transitions makes them dependent on compliance with the policy.

The agent cycle in Fig. 4 is generic: for each agent, the authorizations and reports will be different, but the structure of the net is the same.

The Petri Net fulfils the delegation axioms that we defined in section 2. Receive and provide micro-events require an authorization. They produce records. The transformation to reports is not direct, but for all Petri Net runnings that leave no tokens in the system, except for the money place, it must be the case that the reports have been received by the principal and processed. So under this assumption (an assumption we have to make for all axioms that have an “eventual” in them), the delegation axioms are fulfilled in the agency cycle Petri Net, and in the complete model (the Petri Net of which the agency cycle is a decomposition). What this means is that the agency cycle can be taken as normative (SOLL model). Any actual accounting system that deviates from this pattern is vulnerable to governance risks. Risks can be analyzed ex ante by modifying the net, e.g. by introducing the possibility of deleting records. If this is done by one agent, will it be detected somehow? Same question for an agent violating his authorizations.

4 Conclusion

In this paper, we have analyzed the responsibility relationship as part of the REA ontology. Recording the accountability tree is necessary for any accounting system, but not sufficient. We have identified a number of rules that should be met, for instance, that agent actions (event participation) should be properly authorized. Then we have analyzed how a Colored Petri Net can represent these rules in a dynamic way. An advantage of this representation is that it can be used for simulation and analysis. The “responsibility part” of an accounting information system (the authorization tables, the authorization structure, the rigor of the primary reporting, Segregation of Duties) are important audit concerns that currently are not served well by the REA ontology. The agency cycle is simple, but should be seen as a minimal normative model.

Colored Petri Nets provide the tools for specifying authorizations (a policy specification language). This language is not included in the paper.

References


