Modelling Responsibility

Tim Storer and Russell Lock

February 17, 2009

Abstract

The purpose of this paper is to document the semantics and concrete notation of a model of responsibility employed in the InDeED Project. We have used this model in several case studies through a variety of inter-related graphical notations. This document provides a basis for consolidation and further refinement.

Contents

1 Introduction

The notion of 'responsibility' is one that is widely used in everyday discourse, but it is surprisingly difficult to establish a precise definition of the term. For the purposes of the work described here, we have established the following definition:

> *A duty, held by some agent, to achieve, maintain or avoid some given state, subject to conformance with organisational, social and cultural norms.*

The term 'duty' refers to more than simply a statement that a given task should be completed (as would be the case for the term 'goal'), it also encompasses aspects of accountability, authority. Responsibilities are rarely broken down to individual instructions (for anything but the most trivial of system this would extremely difficult), instead they represent higher level constructs encompassing a remit for initiative. Initiative is bounded by professional conduct, from an organisational perspective as well as wider social and cultural constraints.

Responsibility modelling has been advocated by a number of authors as a suitable abstraction for modelling, analysis and construction of socio-technical systems [1, 2, 3, 7]. Modelling responsibilities provides an abstraction across a system in terms of the broad duties which agents are expected to discharge. A responsibility model is a succinct denotation of the responsibilities associated with some socio-technical system, the agents which have been assigned responsibilities and the resources which are required or have been allocated for the discharge of a responsibility. Previously, we have applied responsibility modelling to a number of case studies in order to develop concepts and notation [4, 3].

This report summarises our current approach to modelling responsibilities by documenting the semantics of a model of responsibility. The paper is a revision of earlier work on modelling responsibilities described in [6]. Section 2 describes the semantics of the model which is essentially a typed entity relationship structure. We define the modelling

semantics using the Z specification language[5], using the CadiZ toolkit[8]. In section 3, a concrete graphical notation is presented along with its relationship to the underlying semantics. Finally, Section 4 summaries the research described and outlines the planned extensions for the work.

2 Semantics

This section describes the semantics of our responsibility modelling language. The language is described as a typed entity relationship notation, beginning with a CadiZ Z section:

section *ResponsibilityNotation* parents *bagkit*

and some basic types:

$[AGENT, RESOURCE, RESPONSIBILITY]$

2.1 Agents and Agent Structures

An agent is some entity capable of holding a responsibility. This definition implies that the agent is perceived to hold some autonomy in the discharge of responsibilities.

Examples: Police Officer, Web Server, Board of Directors.

2.1.1 Properties

- Technical agents are explicitly anticipated by the definition of agents, since this is a useful abstraction for modelling socio-technical systems in which technical agents are percieved to hold responsibilities.

Example: An automated boarding pass issuing system at an airport is perceived to be responsible for issuing boarding passes, and indeed, it is not clear who else (in the normal case) could be said to be responsible for this

- Agents can be members of groupings of agents termed organisations, which are themselves agents. Agents who are modelled as composed of other agents are implicitly also organisations. However, agents may be denoted as being organisations despite not being composed of other agents explicitly in a model.

  – The denotation of organisations as holding responsibilities is again a useful abstraction for our purposes.

    Example: A police constable is a member of a police force. This responsibility will in fact be discharged by individual police officers. Organisations provide a useful abstraction for modelling responsibility assignments. Organisational agents may be "virtual" in the sense that they are constructions which have no actual "real" identity. Such constructs are often useful when modelling inter-organisational responsibilities.

- Agents in the notation are a conflation of the usual agent modelling terms "actor" and "role". One agent (an actor) can thus act as another (a role). Implicitly, an agent is an actor, unless one or more other agents act as it.

  Example: Bob Smith can act as a Police Constable. A Police Sergeant can act as a Evacuation Team Leader.

- Agents who are members of a common organisation can be arranged in subordinate structures. This is useful when analysing the transfer of responsibilities.

Example: A police constable is subordinate to a police sergeant.

2.1.2 Z Notation

Agents

$$agents : \mathbb{P}\, AGENT$$

$$organisations : \mathbb{P}\, AGENT$$

$$composedOfAgt : AGENT \rightarrow \mathbb{P}\, AGENT$$

$$canActAs : AGENT \rightarrow \mathbb{P}\, AGENT$$

$$subordinateTo : AGENT \times AGENT \rightarrow \mathbb{P}\, AGENT$$

$$organisations \subseteq agents$$

$$\forall\, agent : AGENT \bullet composedOfAgt(agent) \neq \varnothing$$

$$\quad \Rightarrow agent \in organisations$$

$$\forall\, agt, org : AGENT \bullet$$

$$\quad agt \in composedOf(org) \Rightarrow org \notin composedOf(agt)$$

$$\forall\, actor, role : AGENT \bullet$$

$$\quad actor \in canActAs(role) \Rightarrow role \notin canActAs(actor)$$

$$\forall\, actor, role : AGENT \bullet$$

$$\quad actor \in canActAs(role) \land (actor \in organisations \lor role \in organisations)$$

$$\quad\quad \Rightarrow role \in organisations \land actor \in organisations$$

$$\forall\, sub, sup, org : AGENT \bullet$$

$$\quad sup \in subordinateTo(sub, org) \Rightarrow$$

$$\quad\quad sup \in composedOf(org) \land sub \in composedOf(org) \land$$

$$\quad\quad sub \notin subordinateTo(sup, org)$$

2.2 Resources

Resources are passive items which may be provided or expected to be required by the discharge of responsibilities.

Examples Articles: work tools, power supply, water.

2.2.1 Properties

- Resources are either information that an agent needs to know to discharge a responsibility, or articles that are to be utilised during the discharge of a responsibility.
- Required articles may be either permanently consumed, or consumed during the discharge a responsibility.

  Example: Water reserves are consumed in the extinguishing of a fire. A hose will be available to extinguish a fire once the current fire has been extinguished.
- Resources may be composed of other resources. Information resources may be composed of other information resources, and articles may be composed of other articles or information.

  Example: A weather warning contains a number of sections describing different regions. A web server is composed of the hardware and software platform as well as the information provided to users.

Resources

$articles : RESOURCE$

$information : RESOURCE$

$composedOfRce : RESOURCE \rightarrow \mathbb{P}\, RESOURCE$

$\forall\, res1, res2 : RESOURCE \bullet$

$res2 \in composedOf(res1) \Rightarrow res1 \notin composedOf(res2)$

$\forall\, res1, res2 : RESOURCE \bullet$

$res2 \in composedOf(res1) \land$

$res2 \in information \Rightarrow res1 \in information$

2.3 Responsibilities

Responsibilities are the duties to be discharged by agents as described in the introduction.

Example Maintain law and order, save patient's life, extinguish fire, order processing.

2.3.1 Properties

- Responsibilities may be composed of other responsibilities.
- If two responsibilities include each other in their decomposition they are mutually dependent.

  Example: Search & Rescue and Evacuation during a flooding incident.

- All agents hold at least one responsibility (the agent's default responsibility), which is the composition of all other responsibilities held by the agent.
- Resources may be denoted as being required[1] in the discharge of particular responsibilities.
- Where a responsibility is composed of other responsibilities, the resources associated with it will at least be all those resources associated with the sub-responsibilities. The agent's overall responsibility is associated with all the resources that the agent will require in order to discharge their responsibility.
- An agent may be denoted as having a resource, which implies that the agent requires the resource to discharge some responsibility.
- If a role is denoted as requiring a resource, then any actor in that role will also require that resource.
- If a role is denoted as having a resource, then any actor in that role is assumed to also have the resource.

[1]Shorthand for "expected" to be required" - the responsibility may still be discharged despite the resource not being available

Responsibilities

$Agents$

$Resources$

$responsibilities : RESPONSIBILITY$

$composedOfRsp : RESPONSIBILITY \rightarrow \mathbb{P}\, RESPONSIBILITY$

$mutualDep : RESPONSIBILITY \leftrightarrow RESPONSIBILITY$

$responsibilitiesOf : AGENT \rightarrow \mathbb{P}\, RESPONSIBILITY$

$requiresR : RESPONSIBILITY \rightarrow \mathbb{P}\, RESOURCE$

$requiresA : AGENT \rightarrow \mathbb{P}\, RESOURCE$

$has : AGENT \rightarrow \mathbb{P}\, RESOURCE$

$\forall\, rbty1, rbty2 : RESPONSIBILITY \bullet$

$rbty1 \in composedOfRsp(rbty2) \land rbty2 \in composedOfRsp(rbty1)$

$\Rightarrow (rbty1, rbty2) \in mutualDep$

$\forall\, rbty : RESPONSIBILITY, res : RESOURCE \bullet$

$\forall\, srbty \in composedOfRsp(rbty) \bullet$

$res \in requiresR(srbty) \Rightarrow res \in requiresR(rbty)$

$\forall\, agent : AGENT \bullet \exists\, rbty : RESPONSIBILITY \bullet$

$rbty = agentResp(agent) \land$

$\forall\, srbty : RESPONSIBILITY \bullet$

$srbty \in allocatedTo(agent) \Rightarrow srbty \in composedOfRsp(rbty)$

$$\forall\, agent : AGENT \bullet$$

$$requiresR(agentResp(agent)) = requiresA(agent)$$

$$\forall\, agent : AGENT, resource : RESOURCE \bullet$$

$$resource \in has(agent)$$

$$\Rightarrow resource \in requiresA(agent)$$

$$\forall\, actor : AGENT, role : AGENT, resource : RESOURCE \bullet$$

$$actor \in canActAs(role) \land resource \in requiresA(role)$$

$$\Rightarrow resource \in requiresA(actor)$$

$$\forall\, actor : AGENT, role : AGENT, resource : RESOURCE \bullet$$

$$actor \in canActAs(role) \land resource \in has(role)$$

$$\Rightarrow resource \in has(actor)$$

2.4 Responsibility Sharing

Two or more agents may be denoted as holding a common responsibility.

2.4.1 Properties

- **Serial** denotes that although both hold some responsibility, one agent is a backup for another agent (the primary) which is typically expected to discharge the responsibility by themselves. Specification of backups is of use when planning responsibilities. Backups thus model exceptions, when the normal discharge of a responsibility is interrupted for some reason. Backups take three possible forms.

– **Unavailable:** a backup is specified because the primary agent may not be available to discharge the responsibility.

Example A backup safety officer may be appointed in an organisation for when the primary takes holidays. Similarly, a backup web-server may be prepared should the primary suffer a failure.

– **Overloaded** a backup is specified should the primary be unable to discharge its responsibility because the workload of the duty exceeds its resources. The primary continues to discharge the responsibility, but shares this with the backup.

– **Escalation** denotes that an agent cannot discharge its responsibilities because its priviledges or authority are insufficient. In such circumstances, escalation denotes that an agent with greater authority takes over the discharge of the responsibility.

- **In parallel** denotes that both responsible agents are required to act in order discharge a responsibility. The agents will be required to cooperate in order to jointly discharge the responsibility. Cooperation implies that the responsibility will not be discharged fully if both agents do not act.

SharedResponsibilities

$$sharedWith : (AGENT \times RESPONSIBILITY) \rightarrow \mathbb{P}(AGENT \times RESPONSIBILITY)$$

3 Concrete Syntax

We have developed a concrete graphical syntax of the responsibility modelling semantics described above, essentially as a typed-entity relationship diagram. Figure 1 summarises this syntax.

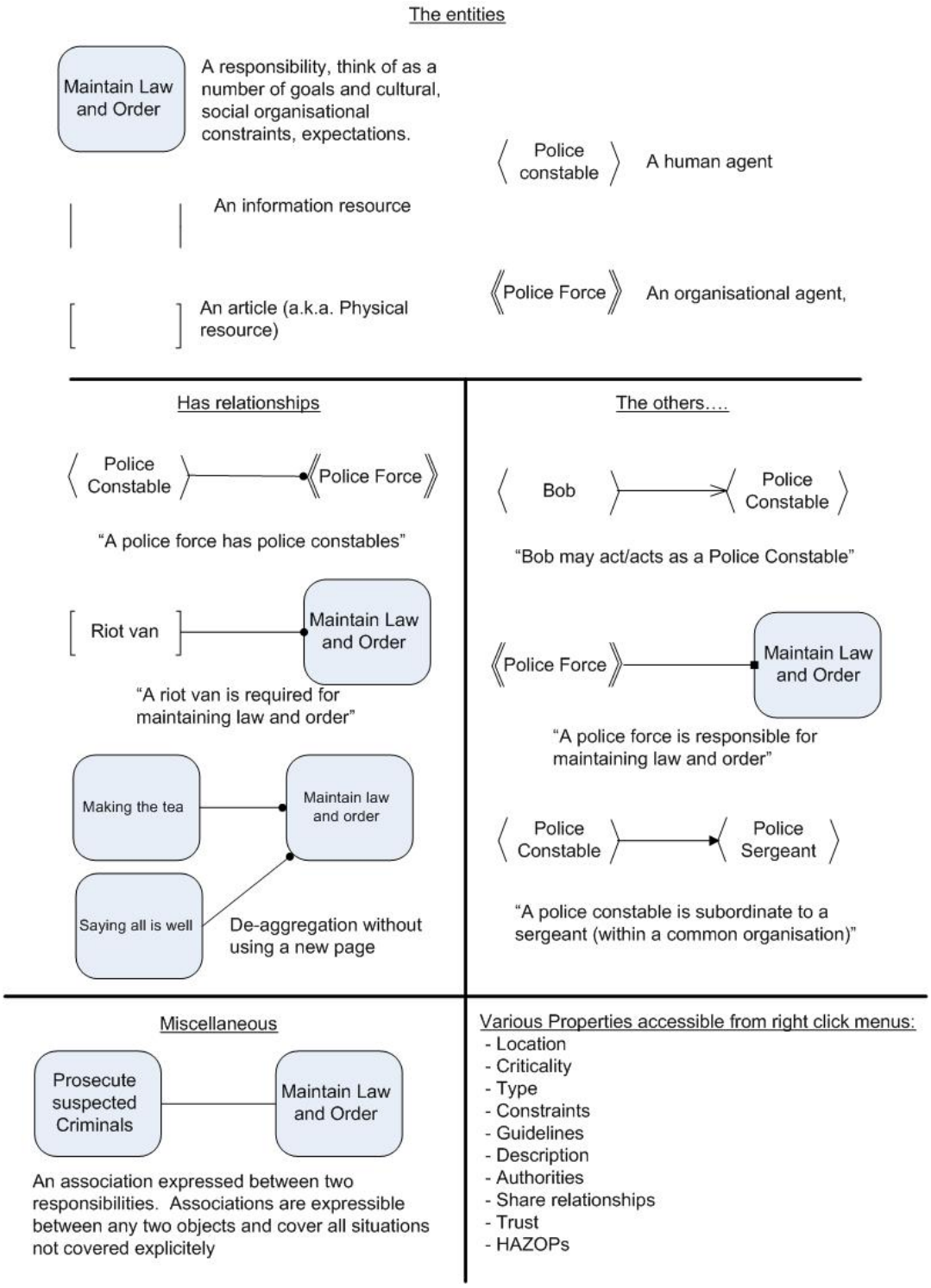

Figure 1: Concrete graphical responsiblity modelling syntax.

4 Summary

The purpose of the work described here is to denote a semantics for modelling responsibilities, the agents who hold them and the resources required to discharge them. Several areas of future work are being undertaken as a result:

4.1 Analysis

We are interested in modelling responsibility structures partly by the desire to analyse these for potential weaknesses. The development of an explicit model of responsibilities allows questions to be asked such as which agents in an organisation map to which in an organisation role. For example, who is the team leader in an evacuation team made up of police officers? Does the Constable assigned to evacuate a street know which residents have physical disabilities which need to be managed?

4.2 Dynamic Responsibility Models

The current modelling semantics and notation permits the denotation of responsibilities from a static perspective. They essentially permit a 'snapshot' of the responsibility structure of a socio-technical system to be denoted from the perspective of one stakeholder. However, responsibility structures are dynamic entities, with responsibilities re-assigned, altered an discharge in response to events in the system and environment. A development of the existing semantics and notation is required to express the changes that can occur to a responsibility structure.

References

[1] Andrew J.C. Blyth, Jarnail Chudge, John E. Dobson, and M. Ros Strens. ORDIT: A new methodology to assist in the process of eliciting and modelling organisational requirements. In S. Kaplan, editor, *Proceedings on the Conference on Organisational Computing Systems*, pages 216–227, Milpitas, California, USA, 1993. ACM Press.

[2] Guy Dewsbury and John Dobson, editors. *Responsibility and Dependable Systems*. Springer-Verlag London Ltd, June 2007.

[3] Ian Sommerville. Models for responsibility assignment. In Dewsbury and Dobson [2], chapter 8.

[4] Ian Sommerville, Tim Storer, and Russell Lock. Responsibility modelling for civil emergency planning. Working Paper 5, InDeED Project, School of Computer Science, University of St Andrews, June 2007. Submitted to the Journal of Reliability Engineering & System Safety.

[5] J.M. Spivey. *The Z Notation: A Reference Manual*. Programming Research Group, University of Oxford, Oriel College, Oxford, OX1 4EW, England, second edition, 1998. `http://spivey.oriel.ox.ac.uk/mike/zrm/index.html`.

[6] Tim Storer and Russell Lock. An integrated model of responsibility for the analysis of the dependability of socio-technical systems. Project Working Paper 1, InDeED Project, April 2007.

[7] Ros Strens and John Dobson. How responsibility modelling leads to security requirements. In *NSPW '92-93: Proceedings on the 1992-1993 workshop on New security paradigms*, pages 143–149, New York, NY, USA, 1993. ACM Press.

[8] Ian Toyn and John A. McDermid. CADiZ: An architecture for Z tools and its implementation. *Software Practice and Experience*, 25(3):305–330, March 1995.