Role Logic and its Application to the Analysis of Process Control Systems from the Socio—Technical System Perspective

Professor Andrew John Charles Blyth, PhD.
Information Security Research Group
Faculty of Computing, Engineering and Science
University of South Wales
Treforest, RCT, CF37 1DL
UK
andrew.blyth@southwales.co.uk

Security requirements for process control systems can be viewed as a social construct derived from the culture and society within which the requirement is said to exist. To capture and understand these requirements we need to make use of a formal reasoning system that supports a rigorous deductive process. Socio—Technical Systems thinking offers us the ability to express the wider socio—context within which an ICT system can be said to operate. In this paper we will extend the π-calculus model of actions via the creation of role logic. Then via the application of responsibility modelling and role logic we will demonstrate how a model of a Socio—Technical process control system can be created and analysed so as to identify critical dependancies.

Socio—Technial Systems, Role Theory, Role Logic, Process Control Systems

1. INTRODUCTION

Socio—Technical Systems analysis and design is the study of the interactions that coexist between complex infrastructures and human/business behaviour Baxter (2011). It focuses upon the analysis of the nature of the discourse between human personnel and technology in a work environment in order to improve performance and working life conditions, Baxter (2011). By socio—technical systems we mean people (individuals, groups, roles and organisations), physical equipment (buildings, surroundings, etc.), hardware and software, laws and regulations that accompany the organisations (e.g laws for the protection of privacy), data (what data are kept, in which formats, who has access to them, where they are kept) and procedures (official and unofficial processes, data flows, relationships, in general anything that describes how things work, or how they could work better) Trist (1978); Whitworth (2006).

Socio—technical systems focus on the working units of interaction that are capable of either linear "cause—effect" relationships, or non—linear relationships that are more complex and unpredictable Mumford (1983). They are adaptable to the constantly changing environment and the complexity that lies in the heart of most organisations. The concept of tasks, their owners, their meaningfulness and the entire responsibility modelling, as well as the dependencies, are also a big part of this theory, Mumford (2006), and thus we assert that Socio—technical systems analysis and design provides a valuable tool for the analysis of process control systems. In this paper we treat people and systems as agents that perform roles and interact with each other. Along with the Socio—Technical Systems design approach, we will use role theory and π-Calculus, Milner (1993). Role theory Biddle (1966) emphasises the fact that roles are basically sets of behaviours or expected behaviours and norms of interaction, while π-Calculus,Sangiorgi and Walker (2001) allows us to explore the dynamic nature of the interactions between roles. A process control system can thus be viewed as a set of interacting roles which may be said to be critical to the organisation. Examples of such systems include safety critical systems, utility management systems and financial systems.

2. RELATED WORK

Traditional approaches to modelling Socio—Technical Systems Mumford (1983) and responsibilities Dewsbury and Dobson (2011) view the system as a set of interacting activities/processes. Agent based modelling approaches Mavee (2012); Dam (2013) view a system as a series of interacting agents and do not place the interaction in the context of a responsibility structure for the delivery of a service. Attack and threat, modelling techniques Larson (Nilsson and Jonsson2007) of Critical Information Infrastructures and Process Control Systems take a very technology—focused view of the infrastructures, and do not address the people issues. Fundamentally, a Critical Information Infrastructure is fundamentally a Socio—Technical System. While formal models of accountability Feigenbaum, Jaggard and Wright (2011); Franz and Wappler (2005) have been developed they have not been set in the context of a Socio—Technical System.

3. EXPRESSING AND MODELLING ROLES AND RESPONSIBILITIES

Responsibility modelling is the analysis and design of the responsibilities within an organisation with the purpose of exploring the internal structure and the dependabilities in the Socio—Technical System Dewsbury and Dobson (2011). Such responsibilities are said to be mediated via a Critical Information Infrastructure. It is one way of exploring the relationship amongst personnel, technical infrastructure, resources and business processes. Hence, the risk associated with any deviation from the expected behaviour can be explored. In the event of an unanticipated change, a before and after analysis can determine what effect the event had on the socio—technical system. In this situation, applying vulnerability analysis will help illustrate the system's strengths and weaknesses and reveal the associated risks.

According to dictionary definitions, responsibility has two meanings:

- The state of having a duty to deal with a certain state of affairs.
- The state of being accountable or to blame for a certain state of affairs.

The first case has a causal connotation meaning the agent has the responsibility for doing something - making an event happen. The second case has a connotation of blame between the action and its result, but does not necessarily imply causality for the agent held accountable, for example, the

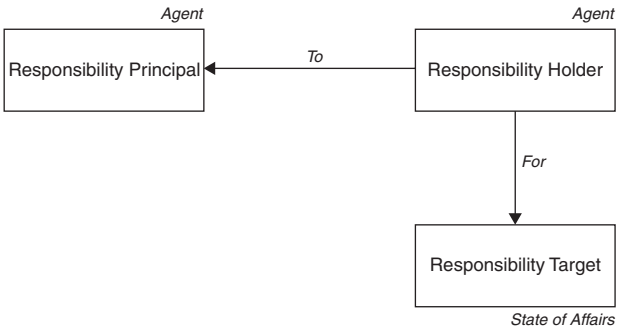

Figure 1: *A Responsibility Model*

parents are held responsible for the actions of their children. As a result, two types of responsibilities can be distinguished, a causal responsibility and a consequential responsibility Dewsbury and Dobson (2011). For instance, each member of a crew on a ship or a plane is causally responsible for the performance of certain tasks, but the captain/pilot is always consequentially responsible for the state of the ship or plane. Within the context of a responsibility we can explore the following:

- Role Assignment: All roles must be executed by an agent and that agent must have had the role assigned to them.
- Role Capability: All agents assigned a role must have the capability to fulfil that role.
- Role Authorisation: An agent must be authorised by another agent to take on a specific role.
- Role Rights Assignment: An agent can only engage in the usage of a right/mode in the context of a role that they are authorised to take.

Responsibility is associated with agents, resources and tasks as defined in the Ontology for a Socio-Technical Systems Model (See Figure 1), and is defined as the duty from one agent (the responsible) to another (the authority or principal) for the accomplishment of a state of affairs, whether this is the execution, maintenance or avoidance of certain tasks, subject to conformance with the organisational culture (Figure 1). Thus the characteristics of a responsibility consist of: who is responsible to whom, for what state of affairs, which are the obligations of the responsibility holder in order to fulfil his responsibility and what type of responsibility it isDewsbury and Dobson (2011).

- Causal responsibility lies effectively between one agent and a state of affairs, while the consequential responsibility is a three way relationship between two agents and a state of

affairs. In this case, the agent who holds the responsibility can be held accountable, culpable or liable to the *authority* agent as seen in Figure 1. Apparently, the most important part of the diagram for the consequential responsibility is the relationship between two agents as the most important question to be answered is *Who is responsible to whom and in what respect?*. On the other hand, for the causal responsibility the most important part is the relationship between the agent and the task as the most important question to be answered is *who is responsible for this action?*. Causal responsibility is a dynamic behavioural relationship between an agent and a state of affairs.

- Consequential responsibility indicates the organisational relationships within organisations and their objectives. Due to its nature, consequential responsibility may be held by more than one agent - it could rest upon an entire organisation - whereas the causal usually lies upon one agent. However, whereas the former can also be delegated from one agent to another, causal responsibility is normally not capable of that although it can be transferred. Types of consequential responsibility include:
  - *Accountability*. This is where a responsibility holder is required by the responsibility principal to give an account of the actions through which the agent has failed to discharge a causal responsibility. This account can take various forms such as a verbal or written statement.
  - *Liability*. This is where a responsibility holder is required by the responsibility principal to be liable for some form of recompense with regard to the failure of causal responsibility. The liability typically takes the form of some resource, such as money, that is given to the responsibility agent that the failure of the responsibility holder has impacted upon.
  - *Culpability*. This is where a responsibility holder is required by the responsibility principal to be culpable of the actions through which the agent has failed to discharge a causal responsibility. Culpability typically takes the form of blameworthiness and results in the withdrawal of some right from the responsibility holder. Typical forms of culpability result in the imprisonment of the responsibility holder, or the withdrawal of some form of liberty.

From a formal perspective we can construct a set of primitives N_r that allow us to define the nature of the responsibility.

$$N_r = \{causal, accountable, liable, culpable\} \quad (1)$$

From this simple set we can construct a 4-tuple that defines a responsibility α within a Socio—Technical System as modal action logic operator.

$$\alpha = R_h, R_p, T, S_t \quad (2)$$

Where:

- R_h is the agent within the Socio—Technical System that holds the responsibility and is thus referred to as the *Responsibility Holder*. Let RH be the set of all *Responsibility Holders* within a socio—technical system and $R_h \in RH$.
- R_p is the agent within the Socio—Technical System to whom the responsibility holder agent R_h is responsible and is referred to as the *Responsibility Principal*. Let RP be the set of all *Responsibility Principlals* within a Socio—Technical System and $R_p \in RP$.
- *T* is the nature of the responsibility between the *Responsibility Holder* agent R_h and the *Responsibility Principal* agent R_p. The nature of the relationship is defined such that: $T \in N_r$.
- S_t is the state of affairs within the Socio—Technical System to which the responsibility holder agent R_h is responsible. Let W_{STS} be the set of state of affairs within a Socio—Technical System and $S_t \in W_{STS}$.

We can express that concept of an agent R_h fulfilling a responsibility α within the π-calculus reasoning framework as:

$$\|\alpha\|_{R_h} \quad (3)$$

For a responsibility to be well founded, the agent fulfilling the responsibility must be the responsibility holder of the responsibility, such that:

$$\|R_h, R_p, T, S_t\|_{agent} \quad (4)$$

Should $agent \neq R_h$, then the responsibility is judged to be not well-founded as the agent responsible for responsibility is not the agent fulfilling the responsibility. Via Deontic Logic the concept of responsibility as an obligation construct has been expressed in functional terms relating permissions to agency/agent, Horty (2009). At its simplest level the concept of a role is used to define behaviour in terms of norms and behaviours that an agent has to face and fulfil Biddle (1966). Expanding this concept with the model of a socio—technical system and a process control system, we may say that:

- A role defines the division of labour and takes the form of a series of interactions.
- A role always stands in relation with another roles and it is through this relationship that interactions flow.
- A role is both a descriptive and a normative concept that can be used to represent many different organisational realities from the structured to the unstructured.

A role is to be treated as the basic building block that make it possible to move between organisational requirements and the requirements of individual agents. (e.g. from the organisation's role in a project, to the way these responsibilities devolve, to the roles of members of the project team). A role defines norms of behaviour and thereby system requirements. Roles include "appropriate" and "permitted" forms of behaviour, guided by organisational norms, which are commonly known and hence determine expectations and requirements. A role defines the relationships between role holders and the behaviour they expect of one another, which in turn defines many environmental requirements.

In terms of responsibility modelling we can apply logical disjunction and conjunction inference rules to responsibilities as follows:

$$\left\|R_h, R_p, T, S_{t_1} \vee S_{t_2}\right\|_{agent} \leftrightarrow \left\|R_h, R_p, T, S_{t_1}\right\|_{agent} \vee \left\|R_h, R_p, T, S_{t_2}\right\|_{agent} \tag{5}$$

$$\left\|R_h, R_p, T, S_{t_1} \wedge S_{t_2}\right\|_{agent} \leftrightarrow \left\|R_h, R_p, T, S_{t_1}\right\|_{agent} \wedge \left\|R_h, R_p, T, S_{t_2}\right\|_{agent} \tag{6}$$

An agent fulfils a responsibility via the performance of a set of roles in a logical order, we will refer to this ordering as a role program RLP. We can use π-calculus to express and reason about a role program so as to identify security requirements for a *Process Control System*. A role program allows us to express the concept that in order to fulfil a responsibility, an agent performs a set of roles. The execution sequence of a set of roles is defined via the use of π-calculus operators. The operators all have the property of atomicity.

- $Role_1 + Role_2$ means that $Role_2$ is performed after $Role_1$, that $Role_2$ follows $Role_1$. Thus the $+$ operator is used to denote sequencing. This operator is referred to as the $Sequential$—AND operator.
- $Role_1 \oplus Role_2$ means that either $Role_2$ is performed or $Role_1$ is performed. This operator is referred to as the $Sequential$—OR operator. In logical, truth table, terms the $Sequential$—OR operator function as an Exclusive OR.
- $Role_1 \mid Role_2$ means that both $Role_1$ and $Role_2$ are performed in parallel in a concurrent manner.

The following allows us to express the idea of an agent performing a specific role.

$$\left\|Role_1\right\|_{agent_1} \tag{7}$$

We can also define the predicate logic operator $\neg$ for roles as the following:

$$\left\|Role_1\right\|_{agent_1} \vee \neg\left\|Role_1\right\|_{agent_1} = \top \tag{8}$$

$$\left\|Role_1\right\|_{agent_1} \wedge \neg\left\|Role_1\right\|_{agent_1} = \bot \tag{9}$$

We can use the concepts of logical disjunction and conjunction inference rules to define how roles can be constructed and deconstructed, if and only if the same agent is performing both roles.

$$\left\|Role_1 + Role_2\right\|_{agent_1} \leftrightarrow \left\|Role_1\right\|_{agent_1} + \left\|Role_2\right\|_{agent_1} \tag{10}$$

$$\left\|Role_1 \oplus Role_2\right\|_{agent_1} \leftrightarrow \left\|Role_1\right\|_{agent_1} \oplus \left\|Role_2\right\|_{agent_1} \tag{11}$$

We can now start to construct a role based program through the execution of which an agent fulfils a responsibility. For example, the following states that for the agent $agent_1$ to fulfil the responsibility $Responsibility_1$, then role $Role_1$ must be performed by agent $agent_1$, sequentially followed by agent $agent_1$ performing role $Role_2$. The operator $\triangleright$ says that the left hand side of the operator is fulfilled by the performance/execution of the right-hand side of the operator. The right-hand side of the operator is referred to as a role program RLP.

$$\left\|Responsibility_1\right\|_{agent_1} \triangleright \left[\left\|Role_1\right\|_{agent_1} + \left\|Role_2\right\|_{agent_1}\right] \tag{12}$$

It is judged that the responsibility $Responsibility_1$ is well formed if and only if the agent fulfilling the responsibility is the same agent $agent_1$ performing all of the roles associated with the responsibility. We can use the same logical π-calculus operators to deconstruct a role down into a sequence of acts, where each act is a speech act. In a Socio—Technical Systems model the interaction between two role holders defines how, when, where and under what circumstances responsibilities are established, flow through the organisation and are finally discharged or fulfilled. By modelling the life cycle of responsibilities we may attempt to answer a number of types of question:

- The first type of question allows for the examination of the possible conflicts that could arise for any given role holder. The term

conflict is used to denote a situation where a role holder is either obliged or responsible to perform an action, or bring about some state of affairs, whilst at the same time being obligated or responsible not to perform the action, or not to bring about some state of affairs.

- The second type of question is concerned with the elucidation of the conditions under which an agent cannot fulfil a responsibility.
- The third type of question is concerned with the elucidation of which objects act as tokens of responsibilities.
- The fourth type of question is concerned with the delineation of the valid accesses to objects that act as tokens of either responsibilities.
- The fifth type of question is concerned with the examination and comprehension of the correct creation and deletion of the objects that act as tokens of the various types of responsibilities.

From a modal/temporal logic perspective we can apply the following operators to a role program R_{LP}. The operators are as follows:

$$\left\|Responsibility_1\right\|_{agent_1} \triangleright \bigcirc R_{LP} \tag{13}$$

The above modal action operators ($\bigcirc$) shows that the role program (R_{LP}) holds at the next state. Hence the above statement may be read as the responsibility $Responsibility_1$ is fulfilled by, in the next state, the role program (R_{LP}) being executed/performed.

$$\left\|Responsibility_1\right\|_{agent_1} \triangleright \Diamond R_{LP} \tag{14}$$

The above modal action operators ($\Diamond$) shows that the role program (R_{LP}) holds at some point in the future. Hence the above statement may be read as the responsibility $Responsibility_1$ is fulfilled at some point in the future by the role program (R_{LP}) being executed/performed.

$$\left\|Responsibility_1\right\|_{agent_1} \triangleright \Box R_{LP} \tag{15}$$

The above modal action operators ($\Box$) shows that the role program (R_{LP}) holds on the entire subsequent path. Hence the above statement may be read as the responsibility $Responsibility_1$ is fulfilled by the role program (R_{LP}) being executed/performed on the entire subsequent path. Within the context of a process control system we can now start to construct detailed Socio—Technical System models that reflect the flow of abstraction from the roles performed by the technical components of a Process Control System into the human elements of a Process Control System. For example, within a engine control system environment, the chief engineer is accountable to the captain of the ship for the good performance of the engines. One of the functions that the chief engineer has to perform to fulfil the responsibility is to monitor the engine status to ensure that the fuel is getting to the engine. We can express this as following:

- The chief engineer has to read the engine status. The status of the engine is provided to the chief engineer by an engine management system, and the engine management system reads the status of the engine from a variety of sensors.
- The chief engineer has to make a decision based on the information provided on what action to make as a result of the information provided. The chief engineer can either do nothing or take some remedial action.

Using role logic and the modal action operators developed in this paper we can express this responsibility as follows:

- Let C_E represent the agent $chief\ engineer$.
- Let C_A represent the agent $captain$.
- Let GPE represent the state of affairs $good\ engine\ status$.

We can now express the responsibility α between the chief engineer and the captain as:

$$\alpha = C_E, C_A, Accountable, GPE \tag{16}$$

We can decompose the responsibility for the chief engineer down into the following roles:

- Let R_e represent the role $read\ engine\ status$.
- Let R_n represent the role $report\ nothing$.
- Let P_{ra} represent the role $perform\ remedial\ action$.

We can now express the relationship between the responsibility and a set of roles that fulfils the responsibility as follows, where the right hand side of the $\triangleright$ operator represents a role program.

$$\left\|\alpha_1\right\|_{C_E} \triangleright \left[\left\|R_e\right\|_{C_E} + \left\|R_n \oplus P_{ra}\right\|_{C_E}\right] \tag{17}$$

Via the application of the modelled operators we assert the following for the socio—technical system that corresponds to the process control system.

$$\left\|\alpha_1\right\|_{C_E} \triangleright \Diamond \left[\left\|R_e\right\|_{C_E} + \left[\left\|R_n\right\|_{C_E} \oplus \left\|P_{ra}\right\|_{C_E}\right]\right] \tag{18}$$

4. A STATE OF AFFAIRS

A state of affairs is the universe of discourse for a responsibility and defines a possible world for a Socio—Technical System. An agent fulfils a responsibility by having a role, or sequence of roles, performed. A role is performed via the execution of a set of acts and acts interact with the state of affairs via the creation, modification or destruction of objects within a state of affairs. Within the context of a Process Control System a state of affairs is used to express the combination of circumstances applying within a Socio—Technical System at a particular, or possibly future, time. This view of a process control system as a possible world for a Socio—Technical System allows us to express a variety of assertions, in particular:

1. True propositions are those that are true in the actual world.
2. False propositions are those that are false in the actual world.
3. Possible propositions are those that are true in at least one possible world.
4. Impossible propositions (or necessarily false propositions) are those that are true in no possible world.
5. Necessarily true propositions (often simply called necessary propositions) are those that are true in all possible worlds.
6. Contingent propositions are those that are true in some possible worlds and false in others.

5. CONCLUSIONS

All process control systems form part of a Socio—Technical System as criticality is a humanistic construct. Hence constructs like responsibilities allow us to export criticality from a Socio—Technical perspective. In this paper we have derived the concept of role logic from π-calculus, and have shown how this type of logic can be applied to the analysis of Critical Information Systems and Critical Information Infrastructures that form a Process Control System from a Socio—Technical perspective. We achieve this through the concept of agents performing speech acts. Through logical constructs of a formula being $well|formed$ we can explore criticality of role and the agents required to perform a role.

REFERENCES

Baxter G, and Sommerville I. (2011), *Socio-technical systems: From design methods to systems engineering*, Interacting with Computers, Vol. 23, No. 1.

Biddle BJ. and Thomas EJ. (Eds) (1966), *Role Theory: Concepts and Research*, (1st edn). John Wiley & Sons.

Dam KH van., Nikloic I. and Lukszo Z. (Eds) (2013), *Agent-Based Modelling of Socio-Technical Systems*, (1st edn). Springer-Verlag.

Dewsbury G. and Dobson J. (Eds) (2011) *Responsibility and Dependable Systems*, (1st edn). Springer-Verlag.

Feigebaum J, Jaggard AD, Wright RN, *Towards a Formal Model of Accountability*,New Security Paradigm Workshop, ACM.

Franz E. and Wappler U. (2005), *Tailored Reaposonsibility within Component Based Systems* Proceedings of the 8th International Conference on Component-Based Software Engineering, Spinger-Verlag.

Horty J. (2009), *Agency and Deontic Logic*, OUP USA.

Larson UE., Nilsson DK. and Jonsson E (2007), *A General Model and Guidelines for Attack Manifestation Generation*, 2nd International Workshop on Critical Information Infrastructure Security.

Mavee SMA. (2012) *A Multi-agent Immunologically-inspired Model for Critical Information Infrastructure Protection – An Immunologically-inspired Conceptual Model for Security on the Power Grid* Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE Press

Milner R. (1993) *Elements of Interaction. Turing Award Lecture*, Communications of the ACM, 36(1).

Mumford E. (1983) *Designing human systems for new technology: the ETHICS method*, Manchester: Manchester Business School.

Mumford E. (2006) *The story of socio-technical design: reflections in its successes, failures and potential*, Information Systems Journal, Vol 16, pp. 317-342.

Sangiorgi D. and Walker D. (2001) *The π - Calculus: A Theory of Mobile Processes*, Cambridge University Press.

Stirling C. (2001) *Modal And Temporal Properties of Processes*, Springer.

Trist EL. (1978) *On socio-technical systems*, Sociotechnical systems: A sourcebook. San Diego, CA.: University Associates.

Whitworth B. (2006) *Socio-technical systems*, Encyclopaedia of human computer interaction.