

THEORETICAL ARTICLE

Open Access

# Quality assurance in crime scripting

Hervé Borrion

## Abstract

**Background:** With the growing interest in the use of crime scripts and attack scenarios for the development of control measures comes the need for more systematic scripting methods. Information about those sequences of actions that offenders carry out to commit a given type of crime can be extremely valuable to designers as control measures may be designed to influence the possibility to actualise criminal plans. However, there exists limited guidance as to what qualities crime scripts should possess in order to support the creation of suitable requirements, and how they should be handled in a design framework.

**Discussion:** This theoretical work contributes to the production and sharing of scientifically robust, useful and usable crime scripts. Drawing a link with the main application considered in this paper, it details the ways in which scripts can contribute to the development of functional requirements for control measures. It presents a list of defects commonly encountered with requirements specifications, and identifies those that could originate from poorly constructed scripts. This section adopts a modelling approach to identify and discuss the sought qualities of crime scripts, but the results apply to all scripts developed for the purpose of reducing crime.

**Summary:** The author presents a list of twelve quality criteria that could be used to evaluate crime scripts. These were identified by considering the common defects of requirements specifications, and tracing back their potential causes within crime scripts. The criteria relate to the following modelling aspects: typology, traceability, transparency, consistency, context, completeness, parsimony, precision, uncertainty, usability, ambiguity and accuracy. A checklist is also provided to facilitate comparison between scripts, contribute to their utility, and ensure that the information required by designers of security systems is available within the functional requirements to be developed for innovative designs. Ultimately, this first investigation of quality assurance in crime scripting opens an important avenue towards further research on the construction and evaluation of crime scripts, their verification and validation.

**Keywords:** Crime script; Modus operandi; Verification; Risk; Design; Security; Control measure; Use case; Perpetrator techniques

## Background

### Engineering crime out

#### *Control measures and systems requirements*

Crime control is commonly achieved through the implementation of control measures in the forms of security policies, security procedures, security personnel and security products (e.g. Gill 2006; Stowell and Rebovich 2007; Byrne and Marx 2011). Their effectiveness in terms of crime prevention is both context and time dependent, and for this reason it is normally assessed when they are implemented in a given ecosystem by means of randomised control trials or quasi-experiments (Pawson and Tilley 1997). It is not necessary though to wait until they appear

in the field to make a rough estimate of their effectiveness. Unlike biological systems, control measures normally result from goal-driven design processes that confer on them specific characteristics deemed conducive to securing objects, places, services or individuals, hence their designation (INCOSE 2010).

From the beginning, control measures are normally envisaged as candidate solutions to specific security problems (Asnar and Giorgini 2006, Taylor et al 2013), and according to most systems engineering frameworks (SEF) their development should be driven by stakeholders' needs (INCOSE 2010). Similar approaches can be found in problem-solving frameworks such as Scanning, Analysis, Response, and Assessment (SARA) (Clarke and Eck 2003) and the Security Function Framework (Ekblom 2012a, 2012b).

Correspondence: h.borrion@ucl.ac.uk  
UCL Department of Security and Crime Science, 35 Tavistock Square, London WC1H 9EZ, UK

Top-down development methods encourage developers of security measures to specify the behaviours that those should exhibit across a range of operational conditions (Meyer & Ekblom 2012). For this a number of requirements engineering (RE) frameworks were developed. They include KAOS (Dardenne et al. 1993), i\*(Yu 1995), Tropos (Bresciani et al. 2004) and Common Criteria (CCSO 2006). An example of a functional requirement is given here for a museum security system: 'the system should raise an alarm when the frame no longer touches the wall'. This information would normally feature in a requirements document (RD) that specifies (amongst other things) what impact the security measure-to-be is required to have on its environment without over-specifying how it should achieve it. As explained by Van Lamsweerde (2009), requirements can be divided into two categories: Functional requirements that define 'the functional effect that the [crime control] system-to-be is required to have on its environment' and non-functional requirements that define 'constraints on the way the system-to-be should satisfy its requirements, or on the way it should be developed'. In a RD, non-functional requirements typically address the following aspects: time, space, cost, accuracy, reliability, safety, security, user interaction, compliance, installation, development, and maintainability (IEEE 2008, ISO 2011).

#### Process models for requirements engineering

In a security context, crime control measures are designed to influence the possibility to actualise crime choices (Felson and Clarke 1998). To prescribe their functions, a common approach is to refer to the way a crime is likely to occur (Kaplan and Garrick 1981; Cornish and Clarke 1986). For this, a model describing the actions leading to, and following from, a crime event can be extremely useful (Cornish 1994a).

In the field of information security, attack models are used to address software security issues, including attack graph, attack tree, misuse case, abuse case, and attack-defence tree. In an *attack graph*, 'nodes identify a state of an attack, [and] edges represent actions taken by the offender or his victim/unwilling assistant. [...] Each edge has conditions on the users and/or machine. If all the conditions are met, the attack succeeds with a given probability and/or cost' (Phillips and Swiler 1998). With *attack trees*, 'attacks against a system are represented in a tree structure with the goal as the root node and different ways of achieving that goal as leaf node' (Schneier 1999). Sindre and Opdahl (2000) proposed an approach based on *misuse cases*. Unlike normal use cases that represent the interaction between a system and a 'normal user' (Carroll 1995), misuse cases are concerned with harmful actors. This is similar to the approach proposed by McDermott and Fox (1999) with *abuse cases*. Firesmith

(2003) wrote that misuse cases are useful to analyse threats but are insufficient to develop security requirements. To fill in this gap, he subsequently introduced the concept of *security use case* that describes the system actions and system interactions that should occur at the different stages of a misuse case. Considering that existing models 'do not take into account the effects of potential defensive measures', Kordy et al. (2012) introduced the concept of attack defense tree (ADTrees), 'a node labeled rooted tree describing the measures an attacker might take to attack a system and the defenses that a defender can employ to protect the system'. ADTrees have nodes of two opposite types: 'attack nodes and defense nodes, which correspond to an attacker's and defender's sub-goals respectively'. Outside the field of information security, scenarios and process models are also used for the analysis of physical security and safety risks (Ezell et al. 2010, Toubaline et al. 2012). In crime science, a process diagram that shares several modeling properties with Phillips and Swiler's attack graph was used by Brayley et al. (2011) to analyse internal child sex trafficking. The conflict between misuse cases and their corresponding security use cases (or between attacker's and defender's sub-goals) is also embodied in the term 'script clashes' that was coined by Ekblom (2007). In the following the concept of crime script at the heart of this article is further detailed before discussing its application to the design of control measures.

#### Crime scripts

##### Scripts

A script is a predetermined, stereotyped sequence of actions that define a well known situation in a particular context (Schank and Abelson 1977, p. 41). Scripts are related to the concept of schema, i.e. 'abstract cognitive representations of organised prior knowledge, extracted from experiences with specific instances' (Fiske and Linville 1980). In essence, they are a type of schema known as *event schema* proposed to explain 'how knowledge is organised about how to understand and enact behavioural processes'.

Early work on scripts (Abelson 1976, 1981; Schank and Abelson 1977) was situated at the intersection between linguistics, psychology and artificial intelligence. It was an attempt to simulate human cognitive structures and processes involved in understanding text. Scripts were hypothesised as knowledge structures whose function is to represent 'psychological and physical events occupying the mental life of individuals' (ibid).

Instantiated from someone's perspective, a script describes the relation between *casts* (also called actors or roles), *props* and *locations* in a sequence of *actions*, so to

characterise routines occurring in specific scenes. In a script, casts and props represent the individuals and objects involved in the behavioural process considered.

In (Schank and Abelson 1977, p. 42) the script-theoretic approach is illustrated using an example commonly known as the Restaurant script. This script describes a dining procedure from the perspective of the customer. It covers four successive scenes: entering, ordering, eating and exiting, with each scene comprising a sequence of actions that makes explicit the steps involved in it. For example, the first scene (entering) is described as follows: the customer enters in the restaurant, looks at tables, decides where to sit, moves to the selected table, and sits.

'Scripts provide a cognitive representation of how an individual believes a sequence of events will occur' (Abelson 1981), and 'individuals rely on their scripts to guide attention and behaviour, especially in new situation (Fiske and Taylor 1991). For this reason, scripts also represent an individual's interpretation of cultural norms (Bowleg et al. 2004; Simon and Gagnon 1986).

The first references to *crime scripts* can be found in (Schank and Abelson 1977) with situations involving an individual stealing coats in a restaurant (ibid, p. 56), a customer leaving a restaurant without paying (ibid, p. 57), and someone carrying out a robbery in a liquor store (ibid, p. 73). However, crime scripting only really found its place in the crime science toolbox a couple of decades later when Cornish (1994a) proposed to use it to support situational crime prevention in his seminal article on the procedural analysis of offending.

### **Crime scripts**

'Crime scripts are [...] simply a way of highlighting the procedural aspects of crimes' (Cornish 1994a, p. 175). The development of Cornish's crime scripting approach originated from the adaptation and application of Abelson's work to represent the commission of crime. In a crime script, casts are identified as the offender and the (human) target, props as weapons, and actions as those atomic activities carried out by offenders during the commission of specific crimes.

The value of crime scripting as a crime analysis technique is perceived to lie in its potential to assist designers of situational prevention measures. It can 'provide a way of eliciting offender's subjective account of crime-commission and provide a framework for constructing more comprehensive and objective accounts of crime-commission synthesised from offender account and other source of information' (ibid).

'When the behavior associated with a script has been used repeatedly and successfully in the past, it will be activated more readily. If strong enough, an activated script will be followed by the scripted action, unless

there are strong inhibitory factors present' (Tedeschi and Felson 1994, p. 181).

### **Levels of abstraction**

Scripts, as a general word for an account of a procedural sequence, can be instantiated at different levels of abstraction. Cornish identified four such levels of specificity: *tracks*, *scripts*, *protoscripts* and *metascripts*. A *track* is defined at the lowest classification level and corresponds to a specific series of actions in specific circumstances. It represents the level at which situational crime prevention is commonly practiced (Leclerc et al., 2011). Cornish (Cornish 1994a, 1998) also provided specific examples of scripts for each level. Examples of scripts at *track level* include subway mugging, and sexual abuse of male children in a particular institutional setting. At the next level (*script level*), the crime is generalised. The scripts for the above examples become 'robbery from the person', 'sexual abuse of male children', respectively. At the next level, further generalization is conducted, and the scripts become 'sexual abuse against children' and 'robbery'. Finally, reaching the metascript level, only remains the direct subgroups of an offense, with 'theft of property', and 'sexual offending'.

In addition to these four levels of abstraction, the components of a universal script were laid down, with the view to represent the generic structure of scripts (Cornish, 1994a). These include *Preparation*, *Entry*, *Precondition*, *Instrumental precondition*, *Instrumental initiation*, *Instrumental actualization*, *Doing*, *Post-condition*, and *Exit*. Tompson and Chainey (2011) recently propose an alternative model with four steps only: Preparation, preactivity, activity and post-activity.

### **Personal, instrumental and situational scripts**

Three types of scripts were identified by Shank and Abelson (1977): instrumental scripts, situational scripts and personal scripts. Both instrumental scripts and situational scripts describe a sequence of actions. However, an instrumental script is a prescriptive model (i.e. recipe) with limited actors, whereas a situational script is a descriptive model with potentially a large number of actors. The third type of script is said to exist only in the mind of the main actor, and consists of a series of possible actions that the latter knows by having conducted them repeatedly. Personal scripts can be goal-oriented, or performed as a ritual, or as a reaction to a situational outcome. Because scripts alone were not sufficient to explain how people deal with new situations, they are defined within a broader epistemic framework comprising goals and plans. The entire set of instruments is based upon the idea that goals are met through the manifestation of events caused by chains of mechanisms (ibid).

**Potential, planned and performed scripts**

To apply the concept of script to real world problems, it seems useful to differentiate between potential scripts, planned scripts and performed scripts.

- Potential scripts describe hypothetical sequences of actions. They are closely related to Heuer and Pherson’s definition of scenarios (2011): “*plausible and provocative stories about how the future might unfold*” and to misuses cases. For example, an attack scenario against a nuclear power plant can be build from a potential crime script that represents the various tracks that a terrorist could follow to break into the plant. This approach can be very risky as the scripts may not be sufficiently realistic or specific to develop effective measures with limited resources. However, it may be the only option available to derive risk scenarios for rare crime events.
- Planned scripts are a subset of potential scripts, and represent sequences of actions that someone intends to execute. Typically they are generated from information obtained through the collection of intelligence. An example of planned script may be an instrumental script that has been selected for a specific operation, e.g. a bank robbery. Planned scripts can be established on the basis of incomplete, uncertain, and sometimes incorrect contextual information. If the context has changed since the planning phase, the actualization of a planned script may result in a (performed) script that only shares a few similarities with the planned script. This would be the case if, for example, covert control measures successfully influence the course of a robbery.
- Performed scripts concern sequences of actions that actually occurred. Most of the crime scripts published in the crime science literature are developed based on empirical data and can be regarded as performed scripts. However, when they are used to develop new control measures, they are then interpreted as a model of what may happen in the future, and consequently also constitute potential scripts.

For those crimes recurrently occurring with the same modus operandi, developing performed scripts can yield relatively realistic potential scripts. However, replication of performed scripts also has limitations, and contextual changes should also be taken into account before using a script as misuse case.

**Application of crime scripts**

Table 1 provides an example of performed crime script with a single track. It describes a ‘crime [that] involved a

**Table 1 Computer input-fraud script (Willison, 2006)**

	SCRIPT ACTION
A <sub>1</sub>	Deliberately gaining access to the organization
A <sub>2</sub>	Already authorized as employee
A <sub>3</sub>	Wait for employees’ absence from offices.
A <sub>4</sub>	Access colleagues’ computers
A <sub>5</sub>	Access programmes
A <sub>6</sub>	False customer account construction
A <sub>7</sub>	Authorization of fictitious invoices
A <sub>8</sub>	Exit programmes
A <sub>9</sub>	Exit the system
A <sub>10</sub>	Spend the transferred Money

local council employee who committed computer fraud. Taking advantage of poor access security (colleagues failed to lock their computers when leaving the office for a substantial period of time), the employee would wait until other members of staff had vacated the office. He would then access their computers to process the fraud. In total £15,000 was embezzled, through the setting-up, inputting and authorization of fictitious invoices’. (Willison 2006, Commission 1998).

Over the years, a number of crime scripts (or similar models) were produced to describe and analyse different types of crime. These include robbery, vandalism and auto theft (Cornish 1994a, 1994b), resale of stolen vehicle (Tremblay et al. 2001; Morselli and Roy 2008), burglary (Nee and Meenaghan 2006), check forgery (Lacoste and Tremblay 2003), employee computer crime (Willison and Siponen (2009), crimes in public transport (Smith and Cornish 2006), explosive attacks (Clarke and Newman 2006; Le Sage et al. 2013), corruption (Zanella 2013), organised crime (Hancock and Laycock 2010), kidnapping (Yang et al. 2007), sex offenses (Cornish 1998; Beauregard et al. 2007a, 2007b; Deslauriers-Varin and Beauregard 2010; Brayley et al. 2011; Leclerc et al. 2011; Savona et al. 2013), drug manufacturing and drug dealing (Chiu et al. 2011; Jacques and Bernasco 2013), illegal waste activities (Tompson and Chainey 2011), stalking (Yanowitz and Yanowitz 2012; Leclerc 2013) and wildlife poaching (Hill et al. 2014). In addition, many crime scripts were created by practitioners and scholars over the years without necessarily being published.

**Verification and validation**

The number and diversity of published crime scripts illustrate the growing interest that exists in this modeling approach amongst the crime science community. However, as Brayley et al. (2011) pointed out, limited guidance can be found as to how crime scripts should be developed, and in particular what information should (not) be included in a script.

Overall, many questions remain to be answered in this field:

- What information should be collected for the creation of crime scripts?
- How crime scripts should be developed from secondary data?
- How crime scripts should be visualized?
- How crime scripts should be verified?
- How crime scripts should be validated?

One may appreciate that there exist multiple degrees of freedom in crime scripting. Scripts can be modeled to represent the actions of different protagonists, can be based on a single or several instances of the same crime, and can be based on one or several accounts. For example, Willison (2006) developed a crime script corresponding to a single instance of crime based on a single account. In contrast Brayley et al. (2011) developed a script that represents on the same diagram five tracks based on multiple accounts, but unlike Savona et al. (2013) they did not explicitly 'consider the actions undertaken by the victims'.

Faced with the task of constructing a script to analyse illegal waste disposal, Tompson and Chainey (2011) felt that, 'for the purpose of greater utility, [...] many of the processes involved in script analysis needed to be streamlined', and contributed to improve the state of the art in crime scripting by proposing a more systematic scripting method as well as a framework for the development of control measures.

There exist dependencies between crime scripts and crime control measures. For example, new ideas for control measures should be based on the information contained in the script used, but equally the information selected for the script should be relevant to the developer of control measures. Recognizing this, Tompson and Chainey (ibid) modelled the problem in terms of prerequisites, facilitators, responsibility, and legislation and regulations. Whilst the scope of the article was limited to 'altering the choice-structuring conditions which give rise to offenders' decision-making', the list still provides a useful basis to generalize the analysis of more preventive options for reducing harm. It was also used more recently by Zanella (2013) to analyse corruption in public procurement.

Emphasizing the practical aspects of situational crime prevention, Clarke and Cornish (1985) recommend working with 'good enough' models, i.e. basic models that serve their purpose. Without a list of qualities to evaluate crime scripts, there is a risk that scripts are *always* considered good enough. In order to avoid institutionalized complacency, it seems wise to ask ourselves why so few crime scripts were eventually used to create innovative control measures. Arguably, a hypothesis might be that crime

scripts are actually unfit for that purpose, and that an improvement of crime scripting practices is needed to (i) increase our understanding of criminal procedures, and (ii) encourage the development of innovative control measures. The following present six arguments supporting this hypothesis.

(1) Comparison is a method commonly used to identify inaccuracy or gaps in data. At present there is no agreed method and criteria that can be used to compare two different scripts representing the same crime event. The term 'script' has different meanings ranging from a type of schema existing only in the head of the perpetrator (e.g. Abelson 1981) to an *a posteriori* description of the actions they actually carried out (e.g. Cornish 1994b). Scripts can also have different scopes. For example, some scripts are limited to one scene whereas, for others, the timeline stretches across several scenes, before and/or after the criminal event. A suitable basis for script comparison and improvement appears to be missing from the literature, and robust quality assurance rules could provide ground for this.

(2) Crime scripts can be considered as a form of ecological models. The application of scientific principles to their development should therefore involve the implementation of a verification stage and a validation stage (Rykiel 1996). In practice, very few articles explicitly report the results of script verification and validation, and confidence in the produced scripts is often almost entirely based on the analyst's credibility.

- In the context of simulation modeling, 'verification is a demonstration that the modeling formalism is correct' (Fishman and Kiviat 1968). For crime scripting, the set of rules and forms that analysts should adhere to is very limited.
- 'Validation is a demonstration that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model' (e.g. Sargent 1984, Curry et al. 1989). However, for many, crime scripts are often provided without information about their application.

There are few cases of scripts that were used to specify new control measures, and even fewer where those measures were eventually implemented and evaluated. Consequently it is difficult to assess whether published crime scripts are actually good enough.

(3) In a field where quantitative techniques are extensively applied to calibrate models of crime patterns and evaluate their statistical significance,

verification and validation of crime scripts should be normal practice. This was recognized by Tompson and Chainey (2011) when stating that 'issues of sampling and representativeness are important limitations to acknowledge'. However, many of the published scripts do not address this point.

- (4) Verification and validation stages are critical to building credible models, i.e. models in which 'a user has sufficient confidence to base scientific and management decisions' (Holling 1978, Sargent 1984). Without the ability to test and verify them, the application of crime scripts to the development of requirements is very limited, as most SEFs based on the V model (a development model with successive phases of test and integration) require systems requirements to be verified and traceable (e.g. INCOSE 2010, ISO 2011; p34-36).
- (5) If a script is to be used for the top-down design of control measures, elicitation and selection of information about the crime procedure should be driven by designers' needs. However, it is unclear how many analysts really understand the needs of designers. This is best illustrated by the SARA model, an example of a semi-structured problem-solving framework, whose main design step gives remarkably little insight into the design process: 'brainstorming for new interventions'. (POP 2013).
- (6) Crime scripts can reflect both cognitive and physical processes. They certainly have the potential to offer additional capabilities over the attack models presented above. However, crime scripts should also be rich enough to include the range of information needed by designers to devise physical control measures. The use of intuitive methods to model this information is perhaps not the most appropriate one, and crime scientists should perhaps look at the field of intelligence analysis where structured methods have been developed to handle some of the tasks (Heuer and Pherson 2011).

## Analysis

The above arguments show there are reasons of both academic and practical natures to improve quality assurance in crime scripting. As a first step toward improving crime scripting methods, this article seeks to identify the qualities that crime scripts should possess in order to offer the quality and credibility required for their application to developing functional requirements. The analysis is detailed in the following section, and a discussion on the application and limitations of the method also follows.

### Method used for the elicitation of qualities

This second section describes the method used to derive the qualities that crime scripts should possess to yield

high quality systems requirements. The range of research designs considered was limited by the paucity of published cases where crime scripts were used for the development of control measures. For this reason, the work reported in this article adopted a theoretical approach based on a backward and deductive analytical method similar to fault tree analysis (e.g. Bedford and Cooke 2001).

- Stage 1: modelling the process by which requirements can be created from crime scripts,
- Stage 2: identifying the type of defects encountered in RDs,
- Stage 3: using the model to deduce the defects in scripts that could cause those in RDs.

NB: The analytical approach used to develop the argument, and several of the terms used in the following section pertain to the fields of risk analysis and systems engineering. In addition, certain aspects of the models discussed are closely linked to computer simulation. For this reason, some readers may feel that the matter discussed here is only relevant to the development of computer simulation. It should be clear, however, that the adoption of an object-oriented approach and the use of structured models borrowed from probabilistic risk analysis to investigate the problem only represents a research design choice, and not a prescribed domain of application.

### Stage 1 – requirements development framework

The following summarizes the method used in a recent EU security project funded under the framework programme 7 to improve Resilience of Infrastructure and Building Security, to elicit functional requirements for new counter-terrorism measures (RIBS 2013).

- Step 1 – Eliciting the high level security requirements from the stakeholders.
- Step 2 – Modelling the scenarios relevant to those requirements.
- Step 3 – Modelling the decision, initiation and completion stages for each activity.
- Step 4 – Carrying out a sensitivity analysis of the selected activities.
- Step 5 – Specifying aspects of the proposed strategies, control principles and mechanisms.
- Step 6 – Determining the response of the various entities, and updating the scenarios.
- Step 7 – Reiterating the previous steps as appropriate.
- Step 8 – Comparing the different alternatives and selecting the most suitable ones.
- Step 9 – Specifying the effects the measure-to-be should have on the selected factors.

- Step 10 – Verifying and validating the systems functional requirements.

Following this framework, the high level security requirements are elicited from the main stakeholders (Asnar et al. 2011). In order to identify the set of risk events and their causal chains, scenarios relevant to the requirements are then modelled and perpetrator scripts developed using an action model that makes the sub-activities explicit, e.g. decision, initiation and completion of activities. For the design of intervention the variables to which the scripts' sub-activities are most sensitive are then identified. Those factors can be decomposed into the individual's intent, motivation and capabilities (Ezell et al. 2010), and may correspond to prerequisites and facilitators mentioned by Tompson and Chainey (2011). The performance of these activities is subsequently characterized as a function of the factors through sensitivity analysis. In order for designers to detail the functional requirements, at least one intervention strategy must be specified that indicates which activities of the crime script should be influenced by the measure-to-be. Domain knowledge, e.g. situational crime prevention (SCP) principles, is then employed to prescribe the control principles and mechanisms that should be employed to alter the factors (Ekblom 1994; Ekblom and Hirshfield 2013, Roach et al. 2005). The scenarios are then adapted to account for the dynamic nature of the entities present in the ecosystem. The process described above is not linear and several iterations of the previous stages typically take place before the (almost) final set of scenarios and requirements is obtained. Once done, the requirements can then be evaluated and prioritized. Inconsistency and conflicts between requirements are identified and addressed. The most appropriate alternatives are then selected, and the effect that the control measures should have on the environment formally specified (Jackson and Zave 1995). Finally, verification and validation of the functional requirements are carried out before being used by a systems architect or designer in a systems engineering framework.

### **Stage 2 – defects in requirements documents**

Van Lamsweerde (2009) identified fourteen common types of defects in requirements specification: Omission; Contradiction; Inadequacy; Ambiguity; Opacity; Noise; Unintelligibility, Unmeasurability; Overspecification; Unfeasibility; Poor structuring; Forward Reference, Remorse and Poor modifiability. The first eight defects are considered strongly dependent upon the way misuse cases (and consequently crime scripts) are modelled, and are detailed below. The other defects relate to the way the information is presented in a RD, or to the introduction of additional requirements, and are therefore less affected by

the content and presentation of crime scripts. For this reason the following focuses on the first eight defects:

'Omission: Problem world feature (PWF) not stated by any RD item. Contradiction: RD items defining a PWF in an incompatible way. Inadequacy: RD item not adequately stating a PWF. Ambiguity: RD item allowing a PWF to be interpreted in different ways. Opacity: RD item whose rationale, authoring or dependencies are invisible. Noise: RD item yielding no information on any PWF. Unintelligibility: RD item stated in an incomprehensible way for those who need to use it. Unmeasurability: RD item stating a PWF in a way that cannot be precisely compared with alternative options, or cannot be tested or verified in [designed] solutions'. (Van Lamsweerde 2009)

### **Stage 3 – analysis**

The final stage of this analysis attempts to link defects in requirements to the quality of crime scripts. As a starting point it is considered that Omission, Contradiction, Inadequacy, Ambiguity, Opacity, Noise, Unintelligibility, and Unmeasurability are observed during the tenth step. Going back through the various steps of the above requirements specification framework, it was possible to identify some of the possible causes at the script level. These qualities are highlighted in the text using italic font, and compiled in the section titled Results.

- (1) To avoid undesirable omission, scenarios and alternatives should ideally be developed to ensure that no suitable alternative (amongst those considered) is discarded. This implies that, in Step 6, the scenarios used to assess the alternatives should be *unambiguous*, as *accurate* as possible, and sufficiently *precise* to evaluate the requirements against the selected set of selection criteria. If the *uncertainty* is too large, it may not be possible to differentiate between the different alternatives, which could also result in discarding suitable ones.
- (2) To avoid any contradiction between requirements, the set of selected scenarios should also provide the information needed by an analyst to reject any requirement that conflicts with the non-functional requirements of the organisation, i.e. it should be as *complete* and *usable* as possible. For example, it should be possible to notice that a measure leading to forced containment does not represent a feasible control principle if people's right to self determination features as one of the stakeholders' requirements (Borrion et al. 2012, Taylor et al. 2013).
- (3) Scenarios are used to articulate security risks and enable usable and intelligible requirements to be elicited. However, a list of detailed scenarios, even a very large one, cannot embody all the various

eventualities that could occur in a complex ecosystem. There is therefore a trade-off to find between the *precision* (granularity) of the scripts and their *completeness* in terms of information represented.

- (4) Even with empirical data about past events, potential scenarios can only be postulated within a design framework. The *uncertainty* associated with those scenarios should therefore be specified so that probability and consequences can be taken into account when assessing the various alternatives in Step 8.
- (5) The selected strategies and control principles selected in Step 5 should also be sufficiently *unambiguous*, as *accurate as possible*, and sufficiently *precise* to evaluate the requirements against the set of selection criteria. For example, a requirement for a control measure may relate to the introduction of an obstacle that would prevent an offender from carrying out a specific action. However, if the location of the obstacle is imprecisely prescribed, or if specific information about the *context* is omitted, the designed obstacle may be utterly ineffective.
- (6) Likewise, if the *uncertainty* about an alternative is too large, the assessment in Step 8 may not be sufficiently insightful to decide whether to select or reject it, and to measure whether a candidate solution is satisfactory.
- (7) Sensitivity analysis of the selected activities is performed in Step 4. In practice this typically involves characterizing the outcomes of the script's activities (and their likelihood) as the states of the various entities are modified. Such an analysis can reveal the factors (i.e. requisite and facilitating/hindering conditions) that could be altered to support or disrupt certain activities. In order to ensure that appropriate strategies and control principles are selected in Step 5, the activities described in the script should therefore be sufficiently *unambiguous*, as *accurate as possible*, and sufficiently *precise* to allow analyst to identify the main factors of performance.
- (8) *Consistency, transparency and traceability* can also support the development of feasible and adequate requirements. Both *consistency* and *transparency* about the syntax and method used to represent the script are essential to identify the key factors of performance in an effective and efficient manner. The same applies to *traceability* of the relationships between entities (their states and actions) and the high-level requirements. In addition, to reduce noise, the script should ideally be as *parsimonious* as possible with the identified dependencies between

the activities, states, factors and requirements explicitly indicated.

- (9) In Step 3 the crime script should provide sufficient information to allow the breaking down of activities into three stages: decision, initiation and completion. This implies that the properties of the entities selected for the scripts should be those that influence these three sub-activities. For example, most scripts provide some information about the property 'location' of the offender. This is justified because their ability to see a given physical resource and to use it to commit the crime depends on sensorial accessibility (e.g. visibility) and physical accessibility (e.g. being able to make direct physical contact with the object), and therefore on the distance between the offender and the resource. To develop a rich set of alternatives, it is thus essential to consider not just the most obvious and visible factors (e.g. location) but also the more abstract ones, e.g. trust.
- (10) Finally, for a given script, their *type* should be clearly indicated so to facilitate comparison between scripts as well as their application. Clarifying whether a script is a potential script, planned script, or performed script would allow *uncertainty* to be better considered, and would ensure (where possible) that new designs of control measures are based on probable scripts developed using empirical data, rather than anecdotal or purely hypothetical scripts.

## Summary

### Quality criteria for crime scripts

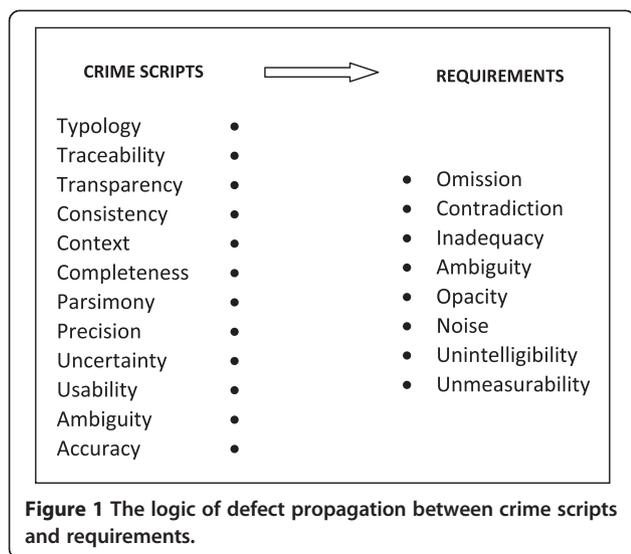
To encourage the development of effective control measures and reduce the presence of defects in RDs, the above analysis recommends that crime scripts should have certain qualities. These relate to the following properties of crime scripts: typology, traceability, transparency, consistency, context, completeness, parsimony, precision, uncertainty, usability, ambiguity and accuracy.

Figure 1 represents the identified links between the defects at the requirements level and the qualities of the crime scripts used to create the requirements.

### Checklist

A checklist is proposed for the development and evaluation of crime scripts:

1. **Typology:** The type of the script should be clearly indicated: potential script, planned script or performed script; perpetrator script, victim script, control script, etc.
2. **Traceability:** All items of information should be explicitly connected to the objectives of the design



problem. Dependencies between the states of the entities and activities should be clearly visible.

3. **Transparency:** The syntax and method adopted for the creation of a script should be clearly communicated, along with the data used for its generation. The criteria used for the development of the model and its calibration should also be made explicit. When multiple scripts are combined, the syntax and method of integration should be provided.
4. **Consistency:** The syntax and method adopted for the creation of a script and the integration of existing scripts should be consistently applied throughout the entire scripting process. Consistency also applies to scripts represented in a diagrammatic form.
5. **Context:** Crime and crime control are both context sensitive. A mention of the context should be added alongside the script so to allow more accurate understanding of the constraints and conditions that could impact on the effectiveness of control measures.
6. **Completeness:** Scripts should include relevant information about the elements that significantly influence the probability distribution of the consequences. Whilst it is understood that ecological models are always incomplete, the main factors of performance should be described for all modelled activities, including physical and psychological ones.
7. **Parsimony:** Scripts should not include any information about those elements that are not relevant to the stakeholders' high level requirements.
8. **Precision:** The precision and resolution of the information included in a script should be based on

the sensitivity of the control measures, and allow effective evaluation of requirements.

9. **Uncertainty:** The uncertainty about the commission of crime and its impact according to the stakeholders' criteria should be explicitly detailed. For example, one should distinguish between sensed, inferred or speculated actions. Equally, when the scripts is based on multiple instances of crime and contain several tracks, the likelihood and statistical significance of each path should ideally be indicated.
10. **Usability:** Scripts should be comprehensive to those expected to use them. When scripts are represented using activity diagrams, both the text and symbols used should be intelligible.
11. **Ambiguity:** It should not be possible to interpret the information forming a crime script in more than one way.
12. **Accuracy:** The intrinsic and relational properties of the elements represented in a crime script should be accurately characterised.

## Discussion

In this article, twelve quality criteria for crime scripts were derived, and a checklist was proposed to support the specification, verification and validation of functional requirements for control measures. This is the first list of its kind for crime scripts, and its application to review the crime scripts mentioned in the background section already provides some insight into the areas that need improvement.

Accuracy, precision and uncertainty are related to model substance. For scripts based on empirical data, the question of accuracy can be divided in two sub-questions: does the data accurately represent the crime phenomenon of interest? and does the script accurately describe the data? Systematic methods for answering these questions remain to be developed. However, it is evident that quantitative methods such as those used by Beauregard et al. (2007a) can provide greater confidence in the results. Another interesting observation concerns the selection of information and its precision. In many cases, the granularity of the scripts was such that the requirements that could be formulated to produce new control measures would probably not be specific enough to be useful to a designer. Finally, very few authors considered or reported the uncertainty present in crime scripts. Given that the script correspond to a potential risk scenario, an indication of the relative likelihood of the different tracks occurring would be useful to allocate resources effectively and prioritize the requirements.

Improving the aspects of crime scripts related to typology, traceability, transparency, parsimony, usability, consistency and ambiguity could greatly facilitate their application in a collaborative environment. A review of

the literature shows that some scripts can be ambiguous. For example, in the above example it is not clear whether the term 'system' refers to the organisation or the computer system. In some cases, the keys of the diagrams were created by the authors of scripts and/or inconsistently employed. Traceability is an issue found in most scripts. Because the activities of the scripts are not explicitly connected to the goals, objectives of the offender (and security team), most scripts found in the literature offer little analytical support for the development of requirements. In particular, it would be impossible to update the scenarios in Step 6 of the above framework without making any assumption about the goals and requirements underpinning the offender's actions. Le Sage et al. (2013) provide useful directions to address current traceability issues in crime scripts, but did not propose a usable presentation format. Finally all scripts were found to be parsimonious with no irrelevant information included.

Completeness should be defined with respect to the scope and limitations of the project for which it is used. Completeness does not mean however that all the relevant information must be represented in the script. However, if the control principles to be actualized by the future control measures are based on situational crime prevention, the elements that factor in the offender's perceived risk, reward, effort, provocations and excuses should feature in the script. For many of the scripts, this was unfortunately not the case. Equally limited information was provided about the context, and if so it was often in a separate section in the document. Clearly formulated situational conditions defining the context could help refining the functional requirements and developing non-functional ones.

### Limitations

The list of criteria presented in Section 2 has both theoretical and practical limitations.

Theoretical limitations – The list of criteria is most certainly incomplete as (1) the set of defects used as input in the identification process was obtained from empirical observations with no guaranty of completeness, and (2) the analytical method employed to deduce the criteria is heuristic in nature. Comparison with the list presented by Wang and Strong (1996) shows that many of the above elements are supported by the theory in the general field of data quality. Some elements in their list are not included, for example objectivity, reputation and believability. However, these qualities are not considered essential to create a step change in the field at this point in time. Empirical research examining actual security related R&D projects could provide valuable information needed to test and improve this list. At this stage the list is still relatively abstract and further work should be carried out to explore these quality criteria individually and more concretely.

Practical limitations – Some of the recommendations are context dependent and are 'getting it right' may not be straightforward. For example, there is currently no visualization system that could provide an effective means of representing the relations between the offender's goals, requirements, objectives, scripts and factors, and those of the security team. If one were to be developed, it is unlikely that it would be comprehensively represented on a static page.

### Conclusions

This article aimed at improving quality assurance in the field of crime scripting to encourage the development of innovative control measures. A list of twelve quality criteria for crime scripts was derived that could be used to improve crime scripting practices. These relate to the following properties of crime scripts: typology, traceability, transparency, consistency, context, completeness, parsimony, precision, uncertainty, usability, ambiguity and accuracy. A checklist was derived from those to support practical applications.

In the short term, researchers could improve their scripts by using the checklist to identify some of the defects that may propagate through the different stages of design frameworks, and impact on the quality of requirements for control measures. The twelve dimensions may also serve to guide the research occurring in this field by highlighting those challenges that should be tackled to improve crime scripting. Finally, this paper adds to the systematic generation of useful and usable crime scripts by articulating how crime scripts can be used to create requirements for control measures.

Two important observations were made after using this list to examine scripting practices. Whilst everyone can intuitively produce a crime script, researchers and practitioners should appreciate the differences that exist between a model that simply represents the data, and one that clearly shows the information needed by its users for a design purpose.

A review of the literature shows that the vast majority of published scripts offer very poor traceability to the objectives. Actions should be more explicitly connected to the goals and objectives of the offender and the security team. Understanding how certain activities contribute to enable other activities or prevent certain mechanisms is essential for effective information selection.

Moreover, it is not clear to what extent scripts are influenced by cognitive biases, in particular confirmation and availability biases. If information selection in crime scripting is driven by the application, wouldn't intuitive crime scripting methods just yield models that support the range of control measures analysts immediately envisage? To what extent our prior knowledge and the information we use to mentally represent and make sense

of situations influence script granularity and information selection? Would an analyst completely foreign to computers represent in details the various steps involved in the procedure of a given software, or would they tend to use a higher level of abstraction and discard valuable information? Two decades after Cornish's seminal paper on the topic, there is evidence that crime scripts should evolve to provide greater analytical support for the design of control measures. At this stage the two most promising directions seem to be: (1) to represent much richer scripts that include not just the perpetrator's actions but all the main entities that influence the criminal procedure, and (2) moving from intuitive to structured scripting methods based on, for example, object-oriented models.

#### Competing interests

The author declares that he has no competing interests.

#### Authors' information

HB is the Deputy Director of the Security Science Doctoral Training Centre at UCL.

#### Acknowledgment

The research reported in this article has received funding from the Seventh Framework Programme FP7 2007-2013 under grant agreement n°242497. The author thanks Dr Tanya Le Sage, Dr Sonia Toubaline and Prof Nick Tilley for the discussions he had with them about this work. He is also grateful to the anonymous reviewers and the editor for their comments.

Received: 1 July 2013 Accepted: 18 October 2013

Published: 31 October 2013

#### References

- Abelson, RP. (1976). Script Processing in Attitude Formation and Decision Making. In JD Carroll & J Payne (Eds.), *Cognition and Social Behavior*. Hillsdale NJ: Erlbaum.
- Abelson, RP. (1981). Psychological Status of the Script Concept. *American Psychologist*, *36*, 715–729.
- Asnar, Y, & Giorgini, P. (2006). Modelling risk and identifying countermeasure in organizations. In *Critical Information Infrastructures Security* (pp. 55–66). Heidelberg: Springer Berlin.
- Asnar, Y, Giorgini, P, & Mylopoulos, J. (2011). Goal-driven risk assessment in requirements engineering. *Requirements Engineering*, *16*(2), 101–116. London: Springer-Verlag.
- Beauregard, E, Proulx, J, Rossmo, K, LeClerc, B, & Allaire, JF. (2007a). Script analysis of the hunting process of serial sex offenders. *Criminal Justice and Behavior*, *34*(8), 1069–1084.
- Beauregard, E, Rossmo, K, & Proulx, J. (2007b). A descriptive model of the hunting process of serial sex offenders: A rational choice approach. *Journal of Family Violence*, *22*, 449–463.
- Bedford, TJ, & Cooke, RM. (2001). *Probabilistic Risk Analysis: Foundations and Methods*. Cambridge (UK): Cambridge University Press.
- Borrion, H, Mitchener-Nissen, T, Taylor, J, & Lai, KM. (2012). Countering Bioterrorism: Why smart buildings should have a code of ethics. In *Intelligence and Security Informatics Conference (EISIC)* (pp. 68–75). European: IEEE.
- Bowleg, L, Lucas, KJ, & Tschann, JM. (2004). The ball was always in his court: An exploratory analysis of relationship scripts, sexual scripts, and condom use among African-American women. *Psychology of Women Quarterly*, *28*, 70–72.
- Brayley, H, Cockbain, E, & Laycock, G. (2011). The Value of Crime Scripting: Deconstructing Internal Child Sex Trafficking. *A Journal of Policy and Practice*, *5*, 132–143.
- Bresciani, P, Perini, A, Giorgini, P, Giunchiglia, F, & Mylopoulos, J. (2004). Tropos: an agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, *8*(3), 203–236.
- Byrne, J, & Marx, G. (2011). Technological Innovations in crime prevention and policing: A review of the research on implementation and impact. *Journal of Police Studies*, *20*(3), 17–38.
- Carroll, JM (Ed.). (1995). *Scenario-based Design: Envisioning Work and Technology in System Development*. New York: John Wiley & Sons.
- Chiu, Y-N, Leclerc, B, & Townsley, M. (2011). Crime script analysis of drug manufacturing in clandestine laboratories: Implications for strategic intervention. *British journal of criminology*, *51*, 355–374.
- Clarke, RV, & Cornish, DB. (1985). Modeling offenders' decisions: A framework for research and policy. *Crime Justice*, *6*, 147–185.
- Clarke, RVG, & Eck, J. (2003). *Become a problem-solving crime analyst: In 55 small steps*. London: Jill Dando Institute of Crime Science.
- Clarke, RV, & Newman, GR. (2006). *Outsmarting the terrorists*. Westport, CT: Praeger Security International.
- Commission, A. (1998). *Ghost in the Machine: An Analysis of IT Fraud and Abuse*. London: Audit Commission Publications.
- Common Criteria Sponsoring Organizations (CCSO). (2006). Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model, Version 3.1 Rev 1. In *In Nat'l Inst Of Standards and Technology CCMB-2006-09-001*. Nat'l Inst Of Standards and Technology CCMB-2006-09-001.
- Cornish, DB. (1994a). The procedural analysis of offending and its relevance for situational prevention. In RV Clarke (Ed.), *Crime prevention studies* 3 (pp. 151–196). Monsey, NY: Criminal Justice Press.
- Cornish, DB. (1994b). *Proceedings of the International Seminar on Environmental Criminology and Crime Analysis* (pp. 30–45). Tallahassee, FL: Florida Statistical Analysis Center, Florida Criminal Justice Executive Institute and Florida Department of Law Enforcement.
- Cornish, DB. (1998). Regulating lifestyles: A rational choice approach. In M Martin (Ed.), *Environmental criminology and crime analysis: Papers of the Seventh International Seminar, Barcelona, June 1998 (Volume 1)* (pp. 165–176). Barcelona, Spain: University of Barcelona.
- Cornish, DB, & Clarke, RV. (1986). *The Reasoning Criminal*. New York: Springer.
- Curry, GL, Deuermeyer, BL, & Feldman, RM. (1989). *Discrete Simulation*. Oakland, CA: Holden-Day.
- Dardenne, A, van Lamsweerde, A, & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, *20*(1), 3–50.
- Deslauriers-Varin, N, & Beauregard, E. (2010). Victims' routine activities and sex offenders' target selection scripts: A latent class analysis. *Sexual abuse: a journal of research and treatment*, *22*(3), 315–342.
- Eklblom, P. (1994). Proximal circumstances: A mechanism-based classification of crime prevention. *Crime prevention studies*, *2*, 185–232.
- Eklblom, P. (2007). *Thinking Thief: Crime frameworks for design against crime. Presented at the Design Against Crime Research*. UK, London: Centre.
- Eklblom, P. (2012a). The Security Function Framework. In P Eklblom (Ed.), *Design against crime: crime proofing everyday objects. Crime Prevention Studies* 27. Lynne Rienner: Boulder, CO.
- Eklblom, P. (2012b). Happy returns: ideas brought back from situational crime prevention's exploration of design against crime. In G Farrell & N Tilley (Eds.), *The Reasoning Criminologist: Essays in Honour of RV Clarke*. Cullompton: Crime Science series. Willan.
- Eklblom, P, & Hirshfield, A. (2013). *An alternative formulation of SCP principles – the 11Ds (and counting)*. Philadelphia: International Seminar, Environmental Criminology and Crime Analysis, Temple University. June 2013.
- Ezell, BC, Bennett, SP, Winterfeldt, DV, Sokolowski, J, & Collins, AJ. (2010). Probabilistic Risk Analysis and Terrorism Risk. *Risk analysis: an official publication of the Society for Risk Analysis*, *30*(4), 575–589.
- Felson, M, & Clarke, RV. (1998). *Opportunity Makes the Thief. Police Research Series, Paper 98. Policing and Reducing Crime Unit, Research, Development and Statistics Directorate*. London: Home Office. <http://www.homeoffice.gov.uk/rds/prgpdfs/fprs98.pdf>. Accessed 1 June 2013.
- Firesmith, DG. (2003). Security use cases. *Journal of object technology*, *2*(3), 53–64.
- Fishman, GS, & Kiviat, PJ. (1968). The statistics of discrete-event simulation. *Simulation*, *10*(4), 185–195.
- Fiske, ST, & Linville, PW. (1980). What does the schema concept buy now? *Personality and Social Psychology Bulletin*, *6*, 543–557.
- Fiske, ST, & Taylor, SE. (1991). *Social cognition* (2nd ed.). New York: McGraw-Hill.
- Gill, M (Ed.). (2006). *The handbook of security*. Basingstoke: Palgrave Macmillan.
- Hancock, G, & Laycock, G. (2010). Organised crime and crime scripts: prospects for disruption. In K Bullock, RV Clarke, & N Tilley (Eds.), *Situational prevention of organised crimes*. Willan: Cullompton, UK.

- Heuer, RJ, & Pherson, RH. (2011). *Structured Analytical Techniques for Intelligence Analysis*. Washington, D.C.: CQ Press.
- Hill, JF, Johnson, SD, & Borrion, H. (2014). Potential uses of computer agent-based simulation modelling in the evaluation of wildlife poaching. In A Lemieux (Ed.), *Situational Prevention of Poaching. Crime Science*. London: Routledge.
- Holling, CS. (1978). *Adaptive Environmental Assessment and Management*. New York, NY: John Wiley & Sons.
- IEEE. (2008). *IEEE Guide for Developing System Requirements Specifications (IEEE Std 1233 (1998th ed))*. Piscataway, NJ: IEEE.
- INCOSE. (2010). *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, v. 3.2. INCOSE-TP-2003-002-03.2*. Seattle, WA: International Council on Systems Engineering.
- International Organization for Standardization (ISO). (2011). *ISO/IEC/IEEE 29148:2011 - Systems and software engineering — Life cycle processes — Requirements engineering*. ISO/IEC/IEEE Switzerland: ISO.
- Jackson, M, & Zave, P. (1995). Deriving specifications from requirements: an example. In *Proceedings of the 17th international conference on Software engineering* (pp. 15–24). ACM.
- Jacques, S, & Bernasco, W. (2013). Drug dealing. In B Leclerc & R Wortley (Eds.), *Cognition and Crime: Offender Decision Making and Script Analyses. Crime Science*. London: Routledge.
- Kaplan, S, & Garrick, BJ. (1981). On the quantitative definition of risk. *Risk analysis*, 1(1), 11–27.
- Kordy, B, Mauw, S, Radomirović, S, & Schweitzer, P. (2012). Attack–defense trees. *Journal of Logic and Computation*. doi: 10.1007/978-3-642-40196-1\_15.
- Lacoste, J, & Tremblay, P. (2003). Crime and innovation: a procedural analysis of patterns in check forgery. In *Theory for Practice in situational crime prevention. Crime Prevention Studies* (pp. 169–196). Monsey (NY): Criminal Justice Press.
- Le Sage, T, Borrion, H, & Toubaline, S. (2013). *An Object-Oriented Approach for Modelling Security Scenarios. Computer Modelling and Simulation*. Cambridge, UK: International Conference on - April 2013.
- Leclerc, C. (2013). Script analysis for situational crime prevention. In B Leclerc & R Wortley (Eds.), *Cognition and Crime: Offender Decision Making and Script Analyses. Crime Science*. London: Routledge.
- Leclerc, B, Wortley, R, & Smallbone, S. (2011). Getting into the script of adult child sex offenders and mapping out situational prevention measures. *Journal of Research in Crime and Delinquency*, 48(2), 209–237.
- McDermott, J, & Fox, C. (1999). *Using abuse case models for security requirements analysis. In Computer Security Applications Conference, 1999. (ACSAC'99) Proceedings. 15th Annual* (pp. 55–64). Piscataway, NJ: IEEE.
- Meyer, S, & Ekblom, P. (2012). Specifying the explosion-resistant railway carriage—a 'bench' test of the Security Function Framework. *Journal of Transportation Security*, 5(1), 69–85.
- Morselli, C, & Roy, J. (2008). Brokerage qualifications in ringing operations. *Criminology*, 46(1), 71–98.
- Nee, C, & Meenaghan, A. (2006). Expert decision making in burglars. *British Journal of Criminology*, 46, 935–949.
- Pawson, R, & Tilley, N. (1997). *Realistic Evaluation*. London: Sage Publications.
- Phillips, C, & Swiler, LP. (1998). A graph-based system for network-vulnerability analysis. In *Proc. of the 1998 Workshop on New Security Paradigm* (pp. 71–79). Resilience of Infrastructure and Building Security (RIBS). (2013). <http://www.ribs-project.eu>. Accessed 20 June 2013.
- (2013). *Center for Problem-Oriented Policing (POP)*. <http://www.popcenter.org/about/?p=sara> Accessed 20 June 2013.
- Roach, J, Ekblom, P, & Flynn, R. (2005). The conjunction of terrorist opportunity: a framework for diagnosing and preventing acts of terrorism. *Security Journal*, 18(3), 7–25.
- Rykiel, EJ, Jr. (1996). Testing ecological models: the meaning of validation. *Ecological modelling*, 90(3), 229–244.
- Sargent, RG. (1984). A tutorial on verification and validation of simulation models. In *Conference on Winter simulation, Proceedings of the 16th* (pp. 114–121). Piscataway, NJ: IEEE.
- Savona, EU, Giommoni, L, & Mancuso, M. (2013). Human trafficking for sexual exploitation in Italy. In B Leclerc & R Wortley (Eds.), *Cognition and Crime: Offender Decision Making and Script Analyses. Crime Science*. London: Routledge.
- Shank, R, & Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding*. Hilldale NJ: Lawrence Erlbaum Associates.
- Schneider, B. (1999). Attack trees. *Dr. Dobb's journal*, 24(12), 21–29.
- Simon, W, & Gagnon, JH. (1986). Sexual Script: Permanence and change. *Archives of Sexual behaviour*, 15, 97–120.
- Sindre, G, & Opdahl, L. (2000). *Eliciting security requirements by misuse cases. Proc TOOLS Pacific*, 10(1), 120–131. Secaucus, NJ: Springer-Verlag New York.
- Smith, MJ, & Cornish, DB (Eds.). (2006). *Secure and Tranquil Travel: Preventing Crime and Disorder on Public Transport*. London: University College London (UCL) Jill Dando Institute of Crime Science.
- Stowell, J, & Rebovich, D. (2007). Institutional Corrections and Hard Technology. In J Byrne (Ed.), *The New Technology of Crime, Law and Social Control* (pp. 215–286). Monsey, NY: Criminal Justice Press.
- Taylor, J, Margaritis, D, Nasir, ZA, Borrion, H, & Lai, KM. (2013). The Role of Protection Measures and their Interaction in Determining Building Vulnerability and Resilience to Bioterrorism. *Journal of Bioterrorism & Biodefense*, 4, 123.
- Tedeschi, J, & Felson, R. (1994). *Violence, aggression, & coercive actions*. Washington, DC: American Psychological Association.
- Tompson, L, & Chainey, S. (2011). Profiling illegal waste activity: using crime scripts as a data collection and analytical strategy. *European Journal of Criminal Policy and Research*, 17(3), 179–201.
- Toubaline, S, Borrion, H, & Le Sage, T. (2012). *Homeland Security (HST), 2012 IEEE Conference on Technologies* (pp. 111–116). Piscataway, NJ: IEEE.
- Tremblay, P, Talon, B, & Hurley, D. (2001). Body switching and related adaptations in the resale of stolen vehicles. Script elaborations and aggregate crime learning curves. *British Journal of Criminology*, 41(4), 561–579.
- van Lamsweerde, A. (2009). *Requirements Engineering - From System Goals to UML Models to Software Specifications*. Chichester, UK: John Wiley & Sons.
- Wang, RY, & Strong, DM. (1996). Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems*, 12(4), 5–33.
- Willison, R. (2006). Understanding the perpetration of employee computer crime in the organisational context. *Information and organization*, 16(4), 304–324.
- Willison, R, & Siponen, M. (2009). Overcoming the insider: reducing employee crime through Situational Crime Prevention. *Communications of the ACM*, 52(9), 133–137.
- Yang, SL, Wu, B, & Huang, SL. (2007). Kidnapping in Taiwan: the Significance of Geographic Proximity, Improvisation and Fluidity. *International Journal of Offender Therapy and Comparative Criminology*, 51, 324–339.
- Yanowitz, KL, & Yanowitz, JL. (2012). The role of gender in the generation of stalking scripts. *Sex Roles*, 66(5–6), 366–377.
- Yu, E. (1995). *Modelling strategic relationships for process engineering* (PhD thesis). Toronto: University of Toronto, Department of Computer Science.
- Zanella, M. (2013). Script analysis of corruption in public procurement. In B Leclerc & R Wortley (Eds.), *Cognition and Crime: Offender Decision Making and Script Analyses. Crime Science*. London: Routledge.

doi:10.1186/2193-7680-2-6

Cite this article as: Borrion: Quality assurance in crime scripting. *Crime Science* 2013 2:6.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)