NEW DIRECTIONS ON IS SECURITY METHODS

The process view

MIKKO T. SIPONEN

University of Oulu, Department of Information Processing Science, Linnanmaa, P.O.BOX 3000, FIN-90014 Oulu university, FINLAND. E-mail:Mikko.T.Siponen@oulu.fi

Abstract: Information systems (IS) security (ISS) methods can be broken down into process (what steps developers should take) to develop IS and modeling notation (how to model reality). Of these, modeling notation has attracted a lot of attention. Whereas process has been largely ignored. As a step towards remedying the situation, this paper first analyzes the limits of the processes of the existing ISS approaches in the light of the analytical framework. Second, the implications of the results for research are suggested.

Key words: ISS methods

1. INTRODUCTION

Several ISS methods have been nominated (Dhillon & Backhouse, 2001; Siponen, 2002). In recent years, increasing interest has been shown in making sense, comparing and exploring the underlying theoretical assumptions of these different ISS methods (Baskerville, 1993; Dhillon & Backhouse, 2001). These studies have improved our understanding of different ISS methods and their limits, and have suggested important avenues for future research on ISS.

Following Hirschheim and Klein (1992), IS development methods can be broken down into *process* (how, and in which order, IS development is carried out) and *modeling notation* (how things are presented). The waterfall model (Royce, 1970), suggesting a step-by-step process from requirement analysis to testing and maintenance, is an example of process. Object-

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35691-4_52

D. Gritzalis et al. (eds.), *Security and Privacy in the Age of Uncertainty*

oriented notations, such as UML (Fowler & Scott, 1999), are examples of modeling notation.

Malouin and Landry (1983) initiated the debate on nature of process in the area of IS development methods by showing that practitioners do not apply IS methods as prescribed by the method developers. This idea that there cannot be a universal "fit-one-fit all" method, but that IS methods need to be modified to suit different contexts, was later deployed by the method engineering (Brinkkember, 1996; Kumar & Welke, 1992) and amethodological community (e.g., Truex *et al.*, 2000).

However, although practitioners commonly modify IS methods in practice (Truex *et al.*, 2000), it is odd that considerations regarding the applicability and modifiability of processes of ISS methods have not entered the ISS area. The existing analyses on ISS methods (e.g., Baskerville, 1993; Dhillon & Backhouse, 2001) do not address this dimension. The aim of this study is to generate discussion on this issue in the area of ISS methods. To do this, we need first to analyze the assumptions underlying the process of the alternative ISS methods. Second, on the basis of this analysis, we propose requirements for future ISS methods.

The rest of the paper is organized as follows. In the second section, the theoretical framework is presented. In the third section, the alternative assumptions are analyzed. The fourth section discusses the results and implications of the study. Finally, the key results of this study are summarized in the fifth section.

2. AN OVERVIEW OF THE EXISTING ISS METHODS, AND ANALYTICAL FRAMEWORK

An overview of the existing ISS paradigms and approaches. Scholars have found alternative ISS paradigms, and their respective approaches (Baskerville, 1993; Dhillon & Backhouse, 2001; Siponen, 2001, 2002), including: *checklists*, *management standards*, *maturity criteria*, risk management, *self-reflected security mgt. cookbooks*, *information modeling approaches*, *responsibility approaches*, *business process*, *the security-modified IS development approaches* and *viable Information Systems*. These approaches are briefly described next; for details see (Siponen, 2002).

Checklists (e.g., AFIPS, 1979; Wood *et al.*, 1987) list the ideal means of protection found by practitioners, from which the developers tick off those relevant for their organizations. *Management standards* (BS7799, 1993; GASSP, 1999; Sanders *et al.*, 1996) in turn endeavours to capture the best industrial practices in standards and prescribes these standards for all organizations. *Maturity criteria* (Murine & Carpenter, 1984; SSE-CMM,

1998; Stacey, 1996) are similar to ISS checklists and security mgt standards, as these three are practitioner-driven aiming at improving ISS processes. However, maturity criteria have a distinct feature: they assess the maturity level of ISS. *Risk management* (e.g., Saltmarsh & Browne, 1983; Bennett & Kailay, 1992; Halliday *et al.*, 1996) calculates the occurrence of risks and the justification of controls. ISS approaches, which are 1) self-reflective (do not relate their work to existing research) and 2) do not use any research methods, form the paradigm of *self-reflected security management cookbooks* (e.g., Nitzberg, 1997; Sherwood, 1996). *Information modeling approaches* (Ellmer *et al.*, 1995; Pernul *et al.*, 1998) add modeling notations for expressing security constraints in structural and object-oriented IS development methods. *Responsibility approaches* (Backhouse & Dhillon, 1996; McDermott & Fox, 1999; Thomas & Sandhu, 1994) see the identification of workers' role responsibilities as the key question in designing ISS. *Business process security* approaches (Herrmann & Pernul, 1999; Röhm *et al.*, 1998) build security constraints into business processes. *The security-modified IS development* approaches (Baskerville, 1993; Booysen & Eloff, 1995; Hitchings, 1996; James, 1996) comprise a variety of ISS methods which have in a common interest in making use of the ideas of IS development methods. *Viable Information Systems* approaches (Hutchinson & Warren, 2000; Karyda *et al.*, 2001) suggest that the key issue is not on how to build secure, but how to create viable/survivable IS.

Analytical framework. Even though IS scholars have debated the nature of processes, they have not used any comprehensive framework, or intellectual map, which could be utilized for scrutinizing the assumptions underlying ISS processes. Hence, in this article a framework is synthesized from the Western philosophical literature.

We suggest that the debate on normative theories in philosophy is an excellent candidate for identifying the nature of ISS processes; e.g., for identifying whether a predefined universal process fitting all ISS development settings exists? Were found three intellectual assumptions in ISS methods: *monistic deontological*, *pluralistic deontological* and *consequentialism.*

According to *monistic deontology*, there is one predetermined, "engraved in stone", set of actions that fits all organizations and problems, i.e., one ISS method fits all development settings. Predetermined means that the developer applying the ISS method needs not to pay attention to the consequences of applying the method in an organization. This also means that monistic deontological methods hold that developers need not modify the methods as a result of negative consequences or experiences. In fact, such a situation is impossible for the monistic deontologist, because monistic deontological view holds that one method *per se* fits to problems. The

philosophical background of *pluralistic deontology* stems from the *prima-facie* principles developed by W. D. Ross (1877-1971). *Prima-facie* in the IS development context means that we any not bound absolutely by any ISS development principles, but rather that they generally hold. Pluralistic deontologist would agree with the monistic deontologist in the respect that ISS methods are predetermined. The difference between pluralist deontology and monist deontology is that pluralist deontology insists that one predefined method, or one predefined set of ISS actions, is not enough. Hence pluralist deontology contends that different problems need different cures, and thus we need different ISS methods for different development situations.

Consequentialism: There are no prefixed rules - the consequences matter! So, in addition to the need to ponder the applicability of methods on a case-by-case basis in different situations (as in the case of pluralistic deontology), consequentialism stresses that no predetermined ISS methods or process can exist. The concept of consequentialism was first introduced by G.E.M. Anscombe, and later made famous by the utilitarians (Bentham, Mill).

3. AN ANALYSIS OF THE ISS PARADIGMS AND METHODS

Checklists. Wood *et al.* and AFIPS take a pluralistic deontological view. Wood *et al.* recognize a shortcoming of the monistic deontological view, namely it is impossible to put forward an all-encompassing list of actions: "*no checklist will ever be truly complete*" (Wood *et al.*, 1987 p. 8). Yet, they see that "*all control ideas described here need to be adapted to the environment being examined.*" (Wood *et al.*, 1987 p. 8). These extracts have a pluralist deontological flavor, i.e., there does not exist one predetermined set of actions that would fit all ISS problems, but different problems need different cures. They also see that we should approach ISS problems with certain predefined principles in mind (pluralist deontology).

AFIPS prescribes clear security imperatives, which organizations generally should follow: "*The following tasks are necessary...* " (AFIPS, 1979 p.11) and "*it is necessary to avoid the following management fallacies...*" (AFIPS, 1979 p.11). And yet, AFIPS recognizes that the checklist is not all-encompassing: "*Not all the questions in this manual can be used directly. It is essential to apply judgment, logic, analysis, and hard thinking.*" (AFIPS, 1979 p. 11). This implies that the checklist cannot be applied blindly as given, but according to the security expert's own judgment. *Management Standards*: BS7799, GASSP and Baseline security guidelines by Sanders *et al.* subscribe to the deontological view. Underlying BS7799 are both monistic and pluralistic deontological views.

As a whole in its philosophy for securing organizations BS7799 has adopted a very typical deontological position. It holds that there exist a predetermined set of ISS actions (controls, etc) that fits all organizations. However, BS7799 states that "*some controls are not applicable to every IT environment and should be used selectively, according to local circumstances*" (BS7799, 1993 p. 1). On the one hand, this passage indicates that BS7799 does not subscribe to monistic deontology. On the other hand, BS7799 maintains that "*however, most of the controls documented are widely accepted by large, experienced organizations as recommended good practice for all situations...*" (ibid p. 1). This latter passage ("*for all situations*") stresses monistic deontological thinking. This view is expressed more clearly in the concept of key controls, which "*apply to all organizations and environments*" (BS7799, 1993 p. 2). Thus the key controls are seen in monistic deontological terms; they prescribe a predetermined set of controls, which are universally applicable – every organization should use them.

Similar to BS7799, the approach taken by Sanders *et al.* (1996) is deontological as the guidelines are predetermined. This approach includes elements which draw on both monistic and pluralistic deontology. With respect to monistic deontology, the guideline states that "*the security guideline...should be applied in any European Health care Establishment...*" (Sanders et al., 1996 p. 87). The guideline provides two hints that it takes a pluralistic deontological view. First, it states that its aim is to provide a minimal protection (Sanders *et al.*, 1996 pp. 82, 84). Second, it declares that "*some of the guidelines may not be totally suited to all systems*" (Sanders *et al.*, 1996 p. 87). These two aspects suggest that the guideline entails pluralistic deontological thinking.

The overall approach of GASSP is deontological: an objective authority (GASPP) predefines the principles. This reflects the main motive of GASSP: to build an authoritative standard (GASSP, 1999 pp. 29-33). We further interpret GASSP as monistic deontological. The overall idea of three level ISS principles and the three-stage process are valid for every organization. Thus, we label the view of GASSP on the nature of processes as monistic deontological.

Maturity criteria. All three maturity criteria (SSE-CMM, Stacey's and Murine-Carpenter maturity criteria) engage in monistic deontology. SSE-CMM holds that there is one objective maturity criterion (SSE-CMM, 1998), which is able to indicate the maturity of all IS. The nature of the processes of this criterion is clearly seen from a monistic deontological perspective. SSE-CMM suggests that certain predetermined processes need to be in place, and which of those processes are in place or are missing in organizations,

determines the ISS maturity level of that organization (monistic deontological view).

Stacey's process is monistic deontological: the maturity stages are prefixed and universal, i.e., argued to be valid for assessing the maturity of all organizations (Stacey, 1996 p. 22). Also the list of "musts" (cf., ibid pp. 31-33) for each maturity level supports the conclusion that this approach is monistic deontological (there are predefined and objective (= monistic deontology) lists of "musts" for each maturity level).

The Murine-Carpenter maturity criterion is monistic deontological in the sense that their criterion fits all organizations. However, even though they have adopted monistic deontology at the highest level of abstraction (the criterion itself), they have favoured pluralistic deontology at the lower levels of abstraction. This conclusion is drawn as one can freely select any technique to measure the quality of ISS security (cf., Murine-Carpenter, 1984 p. 213-214).

Self-reflected security management cookbooks. TIPS (Nitzberg, 1997) consists of six stages (1 security policy, 2 risk assessment, 3 security evaluation, 4 security awareness, 5 re-evaluation, 6 crisis management). The TIPS process is a typical waterfall one. According to TIPS, one should start by making a "security policy" and then follow the rest of the stages in order. After reaching the final stages, "re-evaluation" and "crisis management", one can start again or make iterations. The structured waterfall orientation with its fixed stages suggests that the TIPS process is monistic deontological: the stages and their order in development are predetermined.

SALSA engages in monistic deontology: "*the model is generic... for any organization*" (Sherwood, 1996 p. 501) and "*it is universally applicable*" (ibid p. 506). The expressions "*generic*" and "*universally*" suggest that the process of SALSA is applicable as such to every organization.

Risk Management. The Saltmar-Browne generic risk management approach, which consists of seven stages, subscribes to both monistic and pluralistic deontology. Monistic deontology can be seen in the injunction that the seven stages should be carried out in sequential order. The approach is also pluralistic deontological as it suggests the use of a predetermined list of assets in risk mgt. But this list is only valid in general sense, and can be modified, if regarded as irrelevant or unnecessary. Halliday *et al.* present a few possible ways (processes) of using their risk management techniques. Thus, these ways are not the only possibilities, but rather *prima-facie* type suggestions (hence pluralistic deontology). Monistic deontology is behind the Bennett-Kailay risk analysis: they put forward a prefixed seven-stage process valid for all organizations. The stages seem to be carried out in a sequential order from 1 to 7 (cf., Bennett & Kailay, 1992 p. 68) – hence the label monistic deontology.

Information modeling. Only one information modeling approach by Ellmer *et al.* presents a process. This process is monistic deontological as it proposes predetermined stages for ISS development. As the other information modeling approaches did not concern themselves with processes, one may object to our conclusion by arguing that other approaches may equally regard the process as irrelevant or consequentialistic. Our counter-argument would be that because the existing information modeling approaches are expounded by the same research group, we have reason to belief that they hold the very same view as expressed in (Ellmer *et al.*, 1995).

Responsibility approaches. Strens-Dobson (1993) and Backhouse-Dhillon (1996) do not propose any process. As they do not present any predetermined stages we speculate that they might regard the process guiding the use of modeling techniques/notations as consequentialistic, i.e., left to the developers to decide. Thomas-Sandhu's (1994) process is clearly deontological: it consists of five predefined steps. Yet, we see, on the basis of their design methodology stages (Thomas & Sandhu, 1994 Figure 6), that the stages should be carried out in a predetermined sequential order.

McDermott-Fox (1999) present a prefixed (deontological) process for identifying the abuse cases. It is impossible to ascertain whether this process is monistic deontological (i.e., are these only possible stages for sketching abuse cases), or pluralistic deontological. Nevertheless, the deontological five stage includes sub-rules of thumb, which are pluralistic deontological. This is the case, as for each stage McDermot-Fox (1999 p. 59-60) do not provide an explicit list of principles, but rather loose rules of thumb.

Business process. Similar to certain approaches within the responsibility paradigm, the business process approaches concentrate solely on modeling techniques without giving any consideration to the process guiding this notation. For these reasons, our interpretation of the nature of the process in the business process approaches is that they are consequentialistic.

The security-modified IS development approaches. Their descriptions of the nature of processes range from monistic deontological (Baskerville, Booysen-Eloff, Hitchings), monistic deontological and consequentialistic (James). Baskerville proposes five predetermined design phases, which he calls "*the five security design phases*" (Baskerville, 1988 p. 93). We understand that these stages should be followed in the order of 1 through 5. Yet, we interpret his approach to mean that the five phases should be followed in all cases. For these reasons, we conclude that the approach is monistic deontological.

Hitchings' approach is a deontological one. It proposes seven predetermined phases for ISS development. Although the process of virtual methodology is described in a step-by-step manner (Hitchings, 1995 p. 371,

372 and 374), the stages can be undertaken in a free order. Yet, these seven stages are valid for all organizations: "*VM* [virtual methodology] *has been designed to be applied to any IS within any type of organization.*" (Hitchings, 1995 p. 382). This gives reason to suggest that the core process of virtual methodology (consisting of seven stages) has adopted monistic deontological thinking. However, Hitchings (1995 p. 369) also states that the methodology can be adapted by different organizations. But it is not clear whether the term 'adapt' (understood as adjustment to different environments) refers to the models of this method, or to the whole process of virtual methodology, and to what extent one can adjust the virtual methodology.

Booysen-Eloff (1995) suggests that there are four quadrants (cf., spiral model) and 14 stages for developing secure IS. The 14 stages they put forth are predetermined; hence, they hold a deontological view. Booysen-Eloff (1995) do not state explicitly if these stages are the only relevant way of doing things. However, we understand that the stages are pretty much the sa as the stages for ISS development. This interpretation was made as they justified each stage using the formula "it [the stage in question] is necessary because...". Such statements give a strong hint that all 14 stages are necessary. In the light of this interpretation, the approach of Booysen and Eloff is monistic deontological. However, they make one statement, which suggests a different kind of thinking. When they describe the stages they state "*a typical example of the security spiral*" (Booysen and Eloff, 1995 p. 260). If "a typical" means that this is an example of one possible way of doing things, the Booysen-Eloff approach entails a pluralistic deontological view.

Viable Information Systems. Karyda *et al.* (2001) propose a three-stage process: Diagnosis, Re-design and Transformation. The nature of the processes in these stages seems to be monistic deontological. Although this process is iterative, the authors imply that the order of the stages is fixed: first, the stage of diagnosis; second, the stage of re-design and third, the stage of diagnosis (Karyda *et al.*, 2001 p. 459). Yet, they also give four-stage sub-processes for the re-design stage (they do not do the same for the other stages). This four-stage sub-process also hints at a deontological view (Karyda *et al.*, 2001 p. 464). As to the Hutchinson-Warren approach, we did not find any mention of the process. This led us to guess that they subscribe to consequentialism.

4. DISCUSSION AND IMPLICATIONS FOR RESEARCH

This paper scrutinized the assumptions underpinning the processes of ISS methods.

Monistic deontology. The strength of monistic deontology lies in the step-by-step process given to ISS developers. The ideal situation would be one where an optimal universal process, which fits all ISS development problems perfectly, existed.

The foundational weakness of monistic deontology stems from a belief in the uniformity of all ISS development situations. Clearly, several conflicts arise on the conceptual-theoretical and empirical level. To start with the former, let us assume that two ISS methods subscribing to a monistic deontological view propose two different processes. A conflict arises, given that both monistic deontological methods provide different "fits-one-fits all ISS problems", as to which one is the right one. The situation regarding different maturity criteria is a case in point. All ISS maturity criteria (SSE-CMM, Stacey's and Murine-Carpenter maturity criteria) differ – yet they all propose objective, predefined universal criteria. For the sake of argument, imagine for a moment that there are monistic deontological maturity criteria capable of determining the ISS maturity of every organization. It would follow that the 'right one' will be one of these three (and the other two false measured in terms of objectivity), or will be none of these, but one yet-to-be-found. In the former case, the problem that remains is how we recognize which one of these is the only and right one. In the latter case, it would mean that none of the existing ISS maturity criteria could label themselves as objective (or monistic deontological), and anyone seeking the objective criterion would need to wait for the one-to-be-discovered.

Another argument against monistic deontology is to claim that for certain predetermined and all applicable (monistic deontological) processes to exist, all development situations need to be predetermined. This argument is associated with the dispute between 'determinism' and 'indeterminism'. A critique of determinism (e.g., Peirce, 1935; Popper, 1985) may start from the claim that as we cannot exactly predict phenomena, unexpected situations may arise in which the existing methods may not be of any help, or be irrelevant. The critique would maintain that the relevance of processes are dependant on the context and situation in which they are used. And the situations they are used in are so multifarious that a predetermined method fitting all cases is either impossible to formulate or is irrelevant. Nevertheless, even if we were able to sketch a fit-one-fit-all list of ISS actions covering all situations, the list would become far too long to be of use (cf., Hare, 1981). Moreover, the monistic deontological ISS method, by

prescribing certain hard-and-fast processes, shows less respect for developers' freedom, autonomy, creativity and competence.

There seems to be no empirical evidence for the relevance of monistic deontology (i.e., one prefixed process fits all problems) as it bears on ISS methods. However, there is empirical evidence to show that IS development methods, at least certain IS methods, are not *per se* applicable to each and every situation (Hardy *et al.*, 1995; Malouin & Landry, 1983).

Despite the fact that it is hard to believe in the utility of monistic deontology, this way of thinking might have an educative purpose – in addition to learning the weakness of the idea. Monistic deontological ISS methods may be good candidates for consideration as the first method students will learn about in ISS education (cf., Checkland & Holwell, 1998). Clearly, empirical studies are needed to assess the relevance of this idea further.

Pluralistic deontology. It is common for ISS approaches subscribing to a pluralistic deontological view that, while acknowledging that every approach needs to be tailored to different organizations and settings, they do not provide any guidance on how this may be achieved. The different "method engineering" approaches (Brinkkember, 1996) to developing IS have attempted to address this problem. However, similar approaches have been not presented in the arena of ISS methods. Thus, following the lead of "method engineering" is a possible avenue for ISS methods that adhere to pluralistic deontological thinking. In this case, the ultimate challenge arises: to what extent are our previous ISS development knowledge and skills are and valid in new ISS development situations?

Consequentialism. The strengths of the consequentialistic ISS approaches rest on the freedom and autonomy they provide for developers. The main weakness of the present consequentialistic ISS approaches is that they leave developers without any guidance on the ISS process. This may mean that consequentialistic approaches require more skill from developers than other approaches.

5. CONCLUSIONS

ISS methods can be divided into process and modeling. Modeling has gained a lot of attention in ISS literature, while process has not. To fill this gap, this paper analyzed and compared the assumptions underlying the processes of ISS methods in the light of a framework drawn from the philosophical literature. Three alternative views - monistic deontological, pluralistic deontological and consequentialism - were found. The implications of the findings for research on ISS methods were presented.

REFERENCES

AFIPS, (1979), Security: Checklist for Computer Center Self-Audits. AFIPS, USA.

Backhouse, J., and Dhillon, G., (1996), Structures of responsibilities and security of information systems. European Journal of Information Systems. 5(1): 2-10.

Baskerville, R., (1993), Information Systems Security Design Methods: Implications for Information Systems Development. Computing Surveys 25, (4) December, pp. 375-414.

Bennett, S. P., and Kailay, M. P., (1992), An application of qualitative risk analysis to computer security for the commercial sector. Proceedings of the Eighth ACM Annual Computer Security Applications conference.

Booysen, H.A.S., & Eloff, J.H.P. (1995). A Methodology for the development of secure Application Systems. Proceeding of the 11th IFIP TC11 international conference on information security.

Brinkkember, S., (1996), Method engineering: engineering of information systems development methods and tools. Information and software technology, 38(4): 275-280.

Checkland, P., and Holwell, S., (1998), Information, systems and information systems: making sense of the field. Wiley, cop.

BS7799, Code of Practice for Information Security Management, (1993), Department of Trade and Industry. DISC PD003. British Standard Institution, London, UK.

Ellmer, E., Pernul, G., Kappel, G., (1995), Object-Oriented Modeling of Security Semantics. In: Proceedings of the 11th Annual Computer Society Applications Conference.

Dhillon, G. & Backhouse, J. (2001) Current directions in IS security research: toward socio-organizational perspectives". Information Systems Journal, 11(2): 129-156.

Fowler, M. and Scott, K., (1999), UML Distilled: A Brief Guide to the Standard Object Modeling Language. Second edition, The Addison-Wesley.

GASSP, (1999), Generally Accepted System Security Principles (GASSP). Version 2.0. Information Systems Security. June, vol. 8, no. 3.

Halliday, S., Badenhorst, K., and von Solms, R., (1996), A business approach to effective information technology risk analysis and management. Information Management and Computer Security Vol. 4, No, 1, pp. 19-31

Hardy, C.J., Thompson, J.B., Edwards, H.M., (1995), the use, limitations and customization of structured systems development methods in the United Kingdom. Information and software technology, vol. 37, no, 9, pp. 467-477.

Hare R.M. Moral Thinking, Its Levels, Method and Point. Oxford University Press, 1981.

Herrmann, G., Pernul, G., (1999), Viewing Business-Process Security from Different Perspectives. International Journal of electronic Commerce, Vol. 3, No. 3, pp. 89-103.

Hirschheim, R. and Klein, H., (1992), A Research Agenda for Future Information Systems Development Methodologies. In: in W.W. Cotterman and J.A. Senn (eds): Challenges and Strategies for research in systems development, pp. 235-269.

Hirschheim, R., Klein, H. K., and Lyytinen, K., (1995), Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations. Cambridge University Press, UK.

Hitchings, J. (1995). Achieving an Integrated Design: The Way forward for Information Security. Proceedings of the IFIP TC11 11th international conference on information security.

Hutchinson, W. and Warren, M., (2000), Using the Viable Systems Model to Develop an understanding information system security threats to an Organisation. Proceedings of the 1st Australian Information Security Management Workshop.

James, H.L. (1996). Managing information systems security: a soft approach. Proceedings of the Information Systems Conference of New Zealand.

Karya, M., Kokolakis, S., Kiountouzis, E., (2001), Redefining Information Systems Security: Viable Information Systems. Proceedings of the IFIP TC11 16th International Conference on Information Security (IFIP/SEC'01), June 11-13, Paris, France.

Kumar, K. and Welke, R.J. "Methodology engineering: A Proposal for situation-specific Methodology construction", in W.W. Cotterman and J.A. Senn (eds): Challenges and Strategies for research in systems development, pp. 257-269, (1992).

Malouin, J.L., and Landry, M., (1983), The Miracle of Universal Methods in Systems Design, Journal of Applied Systems Analysis, 10: 47-62.

McDermott, J. & Fox, C. (1999). Using abuse case models for security requirements. Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC).

Murine, G.E. and Carpenter, C. L., (1984), Measuring Computer System Security Using Software Security Metrics. In Computer Security: A global challenge, J.H. Finch and E.G., Dougall (eds.). Elsevier Science Publisher.

Nitzberg, S.D., (1999), The Cyber Battlefield: Is This The Setting for the Ultimate World War? Proceedings of Military Communications Conference (MILCOM). Vol. 1.

Pernul, G., Tjoa A. M., and Winiwarter, W., "Modelling Data Secrecy and Integrity", Data and Knowledge Engineering. Vol. 26, pp. 291-308. North Holland (1998).

Peirce, C.S., (1935), "Collected Papers", Vol. 6, USA.

Popper, K., (1985), "Indeterminism and human freedom", In: Popper Selections (eds): D. Miller, Princeton University Press, USA, pp. 247-264.

Royce, W.W., (1970), Managing the development of large software: concepts and techniques. Proceedings of the IEEE WESTCON, Los Angeles, CA, USA.

Röhm, A.W., Pernul, G., Herrmann, G., (1998), Modeling Secure and Fair Electronic Commerce. Proceedings of the 14th Annual Computer Security Applications Conference.

Sanders, P.W., Furrell, and Warren, M.J., (1996), Baseline Security Guidelines for Health Care Management. In the SEISMED Consortium (eds): Data Security for Health Care: Volume 31: Management Guidelines, Baseline Security Guidelines for Health Care Management, pp. 82 - 107, IOS Press, The Netherlands.

Sherwood, J., (1996), SALSA: A Method for Developing Enterprise Security Architecture and Strategy. Computers and Security. Vol. 15, no. 6, pp. 501-506.

Siponen, M.T., (2001), An analysis of the recent IS security development approaches: descriptive and prescriptive implications. In: G. Dhillon (eds:) Information Security Management - Global Challenges in the Next Millennium, Idea Group (2001).

Siponen, M.T., (2002), Designing secure information systems and software: Critical evaluation of the existing approaches and a new paradigm. Academic Dissertation, Acta Universitatis Ouluensis A 387, Oulu University Press.

SSE-CMM, (1998), The Model and the Appraisal Method (v2.0). http://www.sse-cmm.org.

Stacey, T.R., (1996), Information Security Program Maturity Grid. Information Systems Security. Vol. 5, No.2., pp. 22-34.

Strens, R. and Dobson, J., (1993), How responsibility modelling leads to security requirements. Proceedings of the 1992 and 1993 ACM New Security Paradigm Workshop.

Thomas, R.K. & Sandhu. R.S. (1994). Conceptual Foundations for a Model of Task-based Authorizations. Proceedings of the 7th IEEE Computer Security Foundations Workshop.

Truex, D., Baskerville, R. & Travis, J. "Amethodical Systems Development: The Deferred Meaning of Systems Development Methods", Accounting, Management & IT, 10: 53-79, (2000).

Wood, C.C., Banks, W.W., Guarro, S.B., Garcia, A.A., Hampel, V.E., Sartorio, H.P., (1987), Computer Security: A Comprehensive controls Checklist. John Wiley and Sons.