

The Soft Systems Methodology Based Analysis Model in the Development of Selfmanaging Information Systems

Sa'Adah Hassan
Department of Information Systems
Faculty of Computer Science and Information Technology,
University Putra Malaysia, 43400 UPM Serdang,
Selangor, Malaysia

Abstract: Problem statement: In order to be able to manage its own information within its operating environment with minimal human intervention, a selfmanaging information system ought to identify and make use of information from the resources available in its environment. The development of requirements for selfmanaging information systems should start with an appropriate analysis model that can explicitly show the collaborating entities in the environment. The traditional forms of analysis in systems development approaches have not focused on computing systems environments and ways to identify environment resources. **Approach:** This study discusses the analysis activities in the development of selfmanaging information systems. **Results:** We propose an SSM based analysis model, which is able to examine the requirements for selfmanaging information systems. We describe the analysis of one particular system, the inventory management system and illustrate how this system fulfils certain desired selfmanaging properties. **Conclusion:** The SSM based analysis model is able to address the actuation capabilities of the systems and considers internal and external resources in the environment. The analysis model not only takes into account the information from the environment but is also able to provide support in determining the requirements for selfmanaging properties.

Key words: Information systems, soft systems methodology, selfmanaging system, analysis model

INTRODUCTION

Today's Information Technology (IT) has enabled information systems to evolve from 'simple' data processing into a selfmanaging system, in a way previously unimagined. A selfmanaging information system is defined as a system that is able to manage its own information within its operating environment with minimal human intervention (Horn, 2001; Sterritt and Bustard, 2003; Kephart and Chess, 2003; Sterritt, 2005). The growth of IT has also offered a wide range of information available in the computing systems environment to be benefited from by the selfmanaging systems. Therefore, a selfmanaging system ought to identify and make use of information and resources available in its environment. Furthermore, if the provided information is only based on local information sources or databases, the ability to make accurate decisions based on the provided information may be limited. Thus, selfmanaging information systems should be able to continuously monitor changes in the environment,

detect the latest contextual data from the environment, update and provide the necessary information accordingly with minimal intervention by the enduser. Previous study has also shown that knowledge of available resources in the computing system environment is helpful to effectively guide user recovery from service failures (Bustard *et al.*, 2006; 2007; McSherry *et al.*, 2008). It helps minimise the effects of service failures in the computer system by knowing what is happening in its surrounding, thus helping to improve the effectiveness of its services.

Based on its selfmanaging characteristics, the development of selfmanaging information is unlike the conventional system, as the selfmanaging system requires a deeper analysis of relationships in the environment. In particular, the development of requirements should start with an appropriate analysis model that can explicitly show the collaborating entities in the environment. The analysis model should also provide a general understanding of what constitutes the environment of the system (Bustard *et al.*, 2005).

Previous study (Bustard *et al.*, 2006; 2007; McSherry *et al.*, 2008; Hassan *et al.*, 2006; Hassan, 2010) has shown that the relevant environment knowledge needed to support autonomic behaviour can be identified from Soft Systems Methodology (SSM) models. The SSM (Checkland, 1986; Checkland and Scholes, 1990) is useful in providing a model for visualizing environment knowledge that influences its activities (Bustard *et al.*, 2006; 2007; Hassan, 2010). The SSM is based on holistic thinking, a well-established approach to understanding problem situations and 'systems thinking'. A conceptual distinction is made between what a system is (its properties) and what it does (its activities and relationships) and a practical distinction is made between defining the system and modelling its activities as an integrated entity. This study has taken the step forward to explore how the SSM analysis model can provide support in determining the requirements of selfmanaging information systems.

In this study, a brief introduction to selfmanaging information systems and the background of the SSM are presented. The proposed approach is explained with illustration of an analysis of an inventory management system that supports a printing service.

MATERIALS AND METHODS

In this study, we explore how the SSM can be used to identify the requirements of a selfmanaging information system. The following subsection presents its background and the SSM approach to the analysis of selfmanaging information systems. The proposed approach is illustrated by an analysis of an inventory management system that will be discussed in detail in the subsequent section.

Selfmanaging information system: A computing system that involves not only the computer itself, but also a variety of entities, such as devices, software, hardware, servers and databases, the number of which has increased as technology continues to grow. The environment of a computing system is the set of entities that act on or are acted upon by the system but are not part of the system (Ekholm, 1994). The entities outside the system's boundary that are accessible, possibly via connected networks, such as mobile devices, also encourage the sharing of knowledge on available resources in the computing systems environment. Therefore, there is a potential for information systems

to benefit from the available entities in its environment to carry out their tasks effectively. Consequently, a selfmanaging information system should be able to monitor and analyse information from the environment and adjust this information as necessary to ensure that the information provided to endusers is useful, current and accurate.

SSM for selfmanaging information systems:

Understanding the environment is part of the process of determining the requirements of any computing system prior to its creation (Jackson, 1995). In addition to activities such as analyzing, designing, implementing and validating, a systems development approach should also be concerned with the system's interactions with its environment (Sommerville, 2008). However, traditional development approaches do not address or explicitly model the environment to have an effect on the system should any change occur (SEPO, 2005). In contrast, selfmanaging information systems have to be able to sense and recognise changes in the operating environment and to determine what actions to take in response to such changes. Thus, the environment of computing systems must be modelled so that the systems can have knowledge of their environment (Intel, 2005). Apart from that, the analysis model should also take into account the dynamic context when dealing with ubiquitous and heterogeneous resources in the computing environment (Mainsah, 2002). In order to do this, it requires an appropriate analysis methodology to support the creation of a model for an information system and its environment.

The Soft Systems Methodology (SSM) (Checkland, 1986; Checkland and Scholes, 1990) is a systemic approach. The systemic view recognises that the parts and internal relations of a system are dependent on its environment. A system can never be studied in isolation, but must be seen in relation to its environment. The systemic approach provides conceptual tools for modelling the system as well as the environment in which the system is located. The systemic approach depicts the system's environment as being composed of various interveners and active processes that influence the system (Avenierm, 2002). The potential benefits of the SSM as a basis for developing an environment model for an autonomic computing system has also been mentioned in other work (Hassan, 2010; Bustard *et al.*, 2006). The SSM is not only used to clarify objectives in complex and dynamic problem situations but can also be used to explore actual and future needs.

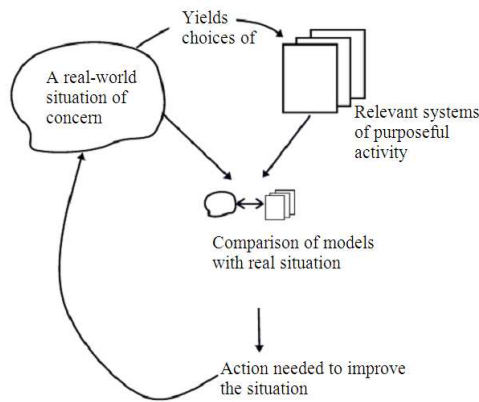


Fig. 1: Basic Shape of SSM (adapted from (Checkland and Scholes, 1990))

The four main activities in an SSM (Checkland and Scholes, 1990), as depicted in Fig. 1, are:

- Analysing a problem situation by drawing a rich picture
- Building purposeful activity models by formulating relevant descriptions of activities as root definitions and conceptual models. The elements of the root definition, as defined below, are often referred to by the mnemonic CATWOE:

Customer:	A beneficiary of the system or anyone who is affected by the system
Actor:	Person who performs the transformation
Transformation:	The conversion of input to output activity carried out by the system
Weltanschauung:	The ‘world view’ which makes the transformation meaningful
Owner:	The person or group with the authority to modify the system
Environment:	Elements outside the system that influence or constrain its activities

- Using the models for debating the situation and seeking from that debate both changes for improvements and action to improve to be taken
- Taking action to improve the situation

RESULTS AND DISCUSSION

The initial analysis provides an understanding of the computing environment and insights into how systems can interact with their changing environment. The first two activities of the SSM are involved in this

initial analysis. The results of the initial analysis, the rich picture and conceptual model of the SSM, are used to provide an understanding of the processes, working scenarios and, potentially, the major information flows likely to affect the system. An inventory management system for a printing service is used to demonstrate the use of the SSM analysis to model the system and its environment.

Analysing the problem situation: The first step in the SSM is to analyse the current problematical situation. A rich picture diagram (Checkland, 1986; Checkland and Scholes, 1990) is often useful for depicting a shared understanding of the problem under investigation and identifying significant external dependencies, issues and possible innovations.

The scenario presented begins with an analysis of an inventory management system that supports a printing service. In general, the main entities of an inventory management system might involve users, suppliers and the administration department (e.g., inventory clerk). The administration department is responsible for managing inventory to ensure enough stock (e.g., study and toner) for all the printers. Figure 2 shows that users and administration personnel may sometimes experience difficulties, for example when the administrator isn’t aware that there is no toner in stock and at the same time, the user is facing the problem of the printer being low in toner. The situation becomes critical if processing the order takes time. A rich picture, as shown in Fig. 2, is developed to illustrate the problem situation in the system.

Conceptual modelling for selfmanaging information systems: The SSM’s rich picture diagram helps stakeholders identify possible viewpoints about the relevant system. For example, one of the possible viewpoints revealed from Fig. 2 is “a system to ensure availability of stock for printer users”. In essence, an SSM root definition captures a relevant system viewpoint. A CATWOE analysis can be used to understand system elements and help formulate the elements of a root definition. A root definition based on the CATWOE elements for the viewpoint of “a system to ensure availability of stock for printer users” is:

- C : Users
- A : Inventory clerk or administrator
- T : Keep excess stock to a minimum
- W : Ensure enough stock on hand to fill anticipated requests
- O : Administration department
- E : Resources in the system’s environment

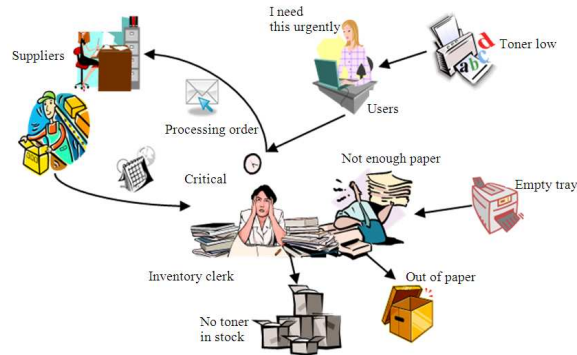


Fig. 2: Problem situation in inventory management system

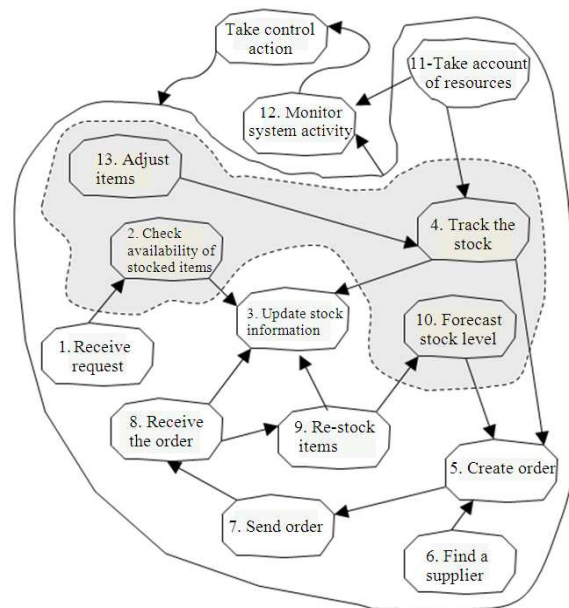


Fig. 3: Preliminary conceptual model of the inventory management system

The elements are combined into a simple statement, namely:

“The administration department owns the system for ensuring that enough stock is on hand to fill anticipated requests from users by assigning an inventory clerk/administrator to keep excess stock to a minimum while taking account of resources in the system’s environment”

The SSM’s conceptual or activity models are built from the root definitions. In the conceptual model, the minimum necessary activities are identified in order to meet the requirements of the root definition. Each is used to identify the activities necessary for the system to meet the purpose specified, as well as relationships between the activities involved. The diagram in Fig. 3 is a conceptual model of the inventory management

system created from the root definition. The conceptual model contains set of related activities needed to carry out the transformation process. The Transformation (T) of the system is to “keep excess stock to a minimum”, which implies the need of information from resources outside the environment that may result in stock problems. Accompanying this transformation is the Weltanschauung (W) or world view, “ensure enough stock on hand to fill anticipated requests”, which defines the purpose of the transformation. The Environment (E) consists of information that can affect the system or the elements that define the limits of the system. The model also includes a set of monitoring and controlling activities and thus the system is thought of as being adaptive. The monitor comprises a detailed description of the monitoring

activities by using the measures of performance from above. In the control action, it is decided whether or not control action is necessary.

The SSM's conceptual model demonstrates an ideal view of the activities and their logical dependencies as described by the root definition. The activities identified in the conceptual model can become a source of ideas for change and improvement. For example, it can be used to enquire about the requirements of the selfmanaging information system in the problem situation. Based on the conceptual model, it is possible to ask what the system needs to know from its environment in order to be able to support each activity. Thus, for each activity in the conceptual model, the requirements of a selfmanaging system could be determined and the data and/or information examined to support the requirements.

From the proposed conceptual model, the information needed can be derived from the entities available in the system's operating environment. The entities can be anything, such as systems, devices, hardware or databases. For example, it can be the printer devices, printer management system and even external systems (i.e., suppliers). Some activities are performed by printer administrators and some of it supported by the printer device.

The conceptual model in Fig. 3 shows that the CATWOE element 'Environment' is Activity 11 (taking account of resources) and represents the relationship between the system and its environment. The system must monitor and analyse information from the external resources to adjust its managed information as necessary. For example, the inventory management system knows how many printers need to be replaced with new toners and thus, adjusts its upcoming order.

Identifying requirements from the conceptual model: In this study, the SSM analysis model is used to identify information about resources or entities in the environment that are available and relevant to the selfmanaging inventory management systems. Each activity in the conceptual model will be examined to identify information from the environment needed by the activity to accomplish its function. At the lower level, each activity may become subsystems or have more than one subsystem and thus share resources or exchange information with other activities in order to complete or improve its function. For example in Fig. 3, activities in the shaded area enclosed by the dotted line can be supported by other subsystems.

Table 1 lists some of the potential functional requirements based on the activities in the conceptual

model with respect to the selfmanaging properties. The requirements also highlight the information that is possible to be used in the system.

Given below are some examples of scenarios in which the entities in the environment support the selfmanaging properties of an inventory management system.

Printer device: Printer status information (e.g., out of toner, empty tray); As an example, for Activity 1 (receive request), instead of receiving a request from the user on the toner cartridge, the system can identify the need to replace the toner cartridge from the printer device itself when the printer is out of toner. The system is aware that the printer is out of toner and needs to be replaced with a new toner cartridge. This might also involve Activity 3 (update stock information) and Activity 5 (create order). For example, to generate a notification of orders based on changes in stock level and automatically create an order in response to quantity shortages.

Printer monitoring system:

- Toner usage; system is aware which printers have high levels of toner usage (i.e., based on usage percentage in days), thus, can be used as a reference for further actions, such as forecasting stock and updating toner inventory databases. The system is able to determine the quantities needed based on the stock situation
- Remaining toner level; system knows the type of toner cartridge that needs to be ordered immediately (e.g., estimated usage time remaining is 3 days) and to be included in upcoming orders (e.g., toner types with remaining level of less than 15%). Thus, the system is able to provide suppliers with advanced notification of upcoming orders
- Number of jobs processed or number of pages printed by each printer; system is aware of overburdened printers or underutilised printers and performs an investigation or analysis of study and toner usage

Online ordering system:

- Historical or seasonal sales demand; As an example, for Activity 5 (create order), instead of waiting for a toner cartridge to reach its minimum level before ordering, the system can be informed through proactive notification from the supplier based on historical orders

Table 1: Selfmanaging requirements for the inventory management system

Activity	Functional requirements	Selfmanaging properties	Information needed	Information resources
Receive request	The system should be able to receive a request from the user	The system will automatically inform by sending a notification to the administrator when toner cartridges show only 10 estimated days of usage	Toner/study status Toners estimated days of usage Staff details (i.e., printer administrator)	Printer device Personnel system
Check availability of stocked items	The system should be able to check the availability of stocked items	The system will continuously update the affected subsystems, automatically check toner cartridges available in stock and notify the inventory clerk	Number of toner cartridges in stock Staff details (i.e., inventory clerk)	Inventory system Personnel system
Update stock information	The system should be able to update stock information	The system will automatically count the available toners and studys and then generate a notification if the number of toners and studys available are insufficient	Inventory status	Inventory system Ordering system
Track the stock	The system should be able to track the stock	The system will continuously monitor printers' usage information and match the stock available	Printer usage Inventory status	Printer device Inventory system
Create order	The system should be able to create an order	The system will automatically generate an order for inventory that reaches the minimum level of stock or has a shortage in quantity	Inventory status Historical sales demand/orders Number of days' supply to stock	Inventory system Ordering system
Find a supplier	The system should be able to find a supplier	The system will automatically generate a list of the potential suppliers based on their history of delivery or transaction	Suppliers History of suppliers' transactions	Supplier database
Send order	The system should be able to send the order to the supplier	The system will automatically send the generated order form to the supplier	Supplier	Supplier database
Receive the order	The system should be able to receive the order	The system will notify users when the order is received from the supplier	List of printer users	Printer device
Forecast stock level	The system should be able to forecast the stock level	The system should be able to analyse the pattern of the inventory's usage	Inventory usage	Inventory system Printer device Printer monitoring system
Take account of resources	The system should be able to take account of available resources	The system will continuously update the affected subsystems automatically and send a notification to the printer administrator for any critical status	Printer status Inventory status Staff status (i.e., printer administrator) Staff details (i.e., inventory clerk)	Printer device Personnel system Printer monitoring system
Monitor system activity	The system should be able to monitor the system activity and take appropriate proactive actions	The system will continuously monitor the system activity: if there is a shortage in inventory, the system will automatically send a notification to the affected subsystems	Inventory status Staff status (i.e., printer administrator) Staff details (i.e., inventory clerk) Enduser status	Inventory system Printer monitoring system Personnel system

- The number of days of supply to stock can be used to dynamically set reorder points and the preferred stock level for each item on an ongoing basis

Figure 4 shows an example of a printer monitoring system and information provided by the system that is useful for toner stock control, for example, in planning for upcoming orders based on estimated days left.

Therefore, the selfmanaging inventory management system should be able to maintain minimum levels of stock to the satisfaction of the administration department as well as printer users. It also encourages systems to share information and available resources in a computing systems environment in order to ensure the continuity of services (e.g., by making use of alternative resources when problems occur) (Sterritt, 2005).

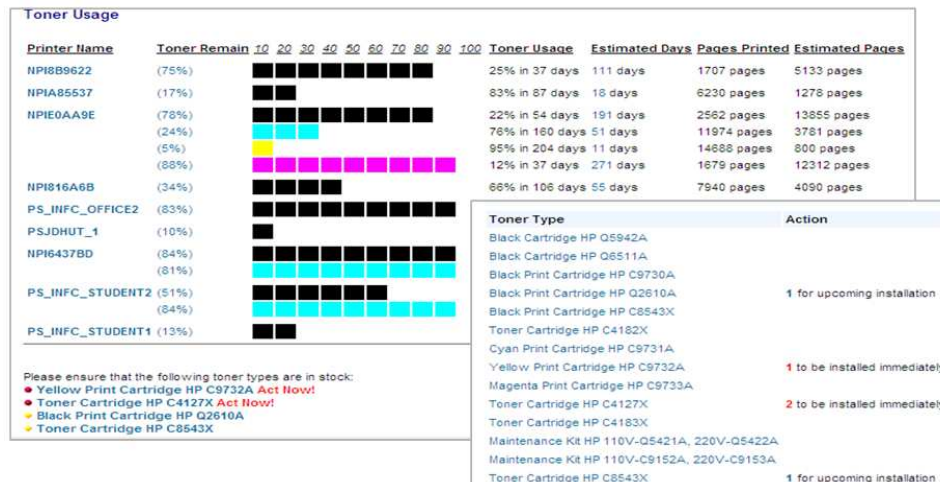


Fig. 4: Printer monitoring system

CONCLUSION

A wealth of information is available on the computing systems environment to support a selfmanaging information system. What is essential is an awareness of available information in the system's operating environment either to achieve the goal of selfmanagement or to improve the efficiency of its operations. However, systems development approaches have not focused on a computing systems environment and ways to identify environment resources in the analysis.

The SSM based analysis is very different compared to traditional forms of analysis. The SSM analysis model is not only able to address the actuation capabilities of the systems but also considers internal and external resources in the environment. The SSM analysis is able to provide a model that takes into account the computing systems environment and a model for visualizing environment information or knowledge that can influence the system. The rich picture and conceptual model of the SSM are helpful in providing an understanding of the system's concerns and information likely to affect the system. Particularly, it provides an understanding of the system's environment and its interaction with its changing environment.

This study concentrates on the modelling of selfmanaging information systems and its environment that should be properly reviewed at the analysis level. Further study on systems development, which will be considered in the future, will concentrate on how to move forward from the results of the analysis.

ACKNOWLEDGMENT

This study is an extended study entitled "Using SSM to Analyzing Environment Awareness in Information Systems" that was earlier presented at the ICTS2008 in Surabaya, Indonesia. This research was supported by the University Putra Malaysia Research Grant RUGS, Project 0505080582RU.

REFERENCES

- Avenierm, M.J., 2002. Designing a collective undertaking and systemic modelling: Potentially fertile concepts.
- Bustard, D., R. Sterritt, A. TalebBendiab, A. Laws and M. Randles *et al.*, 2005. Towards a systemic approach to autonomic systems engineering. Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of ComputerBased Systems, Apr. 4-7, IEEE Xplore Press, Greenbelt, Maryland, pp: 465-472. DOI: 10.1109/ECBS.2005.71
- Bustard, D.W., R. Sterritt, A. Taleb-Bendiab and A. Laws, 2006. Autonomic system design based on the integrated use of SSM and VSM. Artificial Intell. Rev., 25: 313-327. DOI: 10.1007/s10462-007-9031-8
- Checkland, P. and J. Scholes, 1990. Soft Systems Methodology in Action. 1st Edn., Wiley, England, ISBN: 0471927686, pp: 329.
- Checkland, P., 1986. Systems Thinking, Systems Practice. 3rd Edn., John Wiley and Sons, New York, ISBN-10: 0471279110, pp: 330.

- Ekholm, A., 1994. A systemic approach to building modeling analysis of some object-oriented building product models. Lund Institute of Technology.
- Hassan, S., D. McSherry and D. Bustard, 2006. Autonomic self healing and recovery informed by environment knowledge. *Artif. Intell. Rev.*, 26: 89-101. DOI: 10.1007/s10462-007-9033-6
- Hassan, S.A., 2010. Soft systems methodology in environment-aware case-based reasoning system analysis. *Inform. Technol. J.*, 9: 467-473.
- Horn, P., 2001. Autonomic computing: IBM's perspective on the state of information technology. *Comput. Syst.*, 15: 1-40.
- Intel, 2005. Built-in manageability and proactive security for business desktop PCs. Intel vPro technology white study. Intel Corporation.
- Jackson, M., 1995. *Software Requirements and Specification: A Lexicon of Software Practice, Principles and Prejudices*. 1st Edn., ACM Press, Harlow, ISBN: 0201877120, pp: 228.
- Kephart, J.O. and D.M. Chess, 2003. The vision of autonomic computing. *Computer*, 36: 41-50. DOI: 10.1109/MC.2003.1160055
- Mainsah, E., 2002. Autonomic computing: The next Era of computing. *Elect. Commun. Eng. J.*, 14: 2-3.
- McSherry, D., S. Hassan and D. Bustard, 2008. Conversational case-based reasoning in self-healing and recovery. *Adv. Case-Based Reason.*, 5239: 340-354. DOI: 10.1007/978-3-540-85502-6_23
- SEPO, 2005. Perspectives on complex-system engineering. Collaborations, 3.
- Sommerville, I., 2008. *Software Engineering*. 1st Edn., Pearson Education India, ISBN: 8131724611, pp: 864.
- Sterritt, R. and D. Bustard, 2003. Towards an autonomic computing environment. Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03), IEEE Xplore Press, pp: 694-698. DOI: 10.1109/DEXA.2003.1232103
- Sterritt, R., 2005. Autonomic computing. *Innov. Syst. Software Eng.*, 1: 79-88. DOI: 10.1007/s11334-005-0001-5