

Manuscript Information	
Journal Acronym	TEIS
Volume and issue	
Author name	
Manuscript No. (if applicable)	_A_476860

Enterprise Information  
Systems



Typeset by  
KnowledgeWorks Global Ltd.

for



Taylor & Francis  
Taylor & Francis Group

# QUERIES: to be answered by AUTHOR

**Dear Author**

Please address all the numbered queries on this page which are clearly identified on the proof for your convenience.

Thank you for your cooperation

QUERY NO.	QUERY DETAILS
1	Kindly check whether affiliations are OK as set.
2	Please check the heading levels throughout.
3	Kindly check whether the sentence "It is argued by Zhang (2007)..." is OK as edited.
4	Please spell out "RRR, DBMS, DBA, CRM" in full at first mention.
5	Kindly update reference "Bi and Zhang 2008".
6	Kindly provide publisher details for reference "Doidge 2007".
7	Kindly provide title of proceeding for reference "Guelfi <i>et al.</i> 2008".
8	Kindly provide details so that the reader can access reference "Hamel and Valikangas 2003".
9	Kindly provide last accessed date for references "Moslehi and Kumar 2009, Zhang 2009".
10	Kindly check whether references "Ouyang <i>et al.</i> 2005, Pease 2009, Zhang 2007, 2008" are OK as set.
11	Kindly provide place of publication for reference "Pulley and Wakefield 2001".
12	Kindly provide volume number and page range for reference "Wu <i>et al.</i> 2010".

# On the principle of design of resilient systems – application to enterprise information systems

W.J. Zhang<sup>a\*</sup> and Y. Lin<sup>b</sup>

<sup>a</sup>Department of Mechanical Engineering, University of Saskatchewan, Saskatchewan, Canada;

<sup>b</sup>Department of Mechanical and Industrial Engineering, Northeastern University, Boston, Massachusetts, USA

(Received 23 February 2010; final version received 9 March 2010)

Resilience engineering is an emerging discipline. In this article, we discuss the concept of resilience and resilience engineering in light of its distinct identity. We propose the principle of design of resilient systems. Then, we outline how these design principles can be applied to the enterprise information system to make it more resilient. This leads to a proposed architecture of resilient enterprise information systems.

**Keywords:** resilient system; design principle; enterprise information system

## 1. Introduction

Resilience engineering comes to attention only recently despite the fact the notion of resilience has been known in ecological systems and biological systems for a long time. In this article, we attempt to discuss the identity of resilience (in the engineering context) and resilience engineering, as this is the foremost issue to precede any other issue regarding resilience engineering. The discussion is carried out by distinguishing resilience or resilience engineering from other similar notions such as reliability, robustness, sustainability and so on. The second question to be answered by this article is what design principles for resilient systems are, as this is the second most important issue to recognise for the practical value of resilient systems. The answer to the second question is sought by examining some truth with the biological system in relation to the notion of resilience as well as principles that govern the resilience behaviour of the biological system. Finally, we consider the enterprise information system as a kind of service systems. Based on this view, the design principles for the resilient system are specialised to ones for the resilient enterprise information system. By applying these design principles, we outline the architecture of a resilient enterprise information system.

## 2. Resilience versus other system concepts

### 2.1. Resilience versus reliability and robustness

There are a few definitions regarding resilience and resilience engineering in the current literature. Pulley and Wakefield (2001) defined resilience, in the context of

---

\*Corresponding author. Email: [chris.zhang@usask.ca](mailto:chris.zhang@usask.ca)

50 building a system, as: ‘resilience provides the ability to recover quickly from change,  
hardship or misfortune. It is associated with elasticity, buoyancy and adaptation’.  
They further stated; ‘resilient people demonstrate flexibility, durability, and attitude  
of optimism, and openness to learning. A lack of resilience is signalled by burnout,  
fatigue, malaise, depression, defensiveness, and cynicism’. Hollnagel *et al.* (2006)  
55 defined resilience, in light of its implication to safety management, as: ‘resilience is  
the ability of an organisation (system) to keep, or recover quickly to, a stable state,  
allowing it to continue operations during and after a major mishap or in the presence  
of continuous significant stresses’. Hollnagel *et al.* (2006) further stressed on the role  
of prediction of mishaps of the system as part of the scope of resilience engineering in  
60 an extended definition of resilience engineering.

In both of the definitions above, the common emphasis seems to be such that  
disturbances are responsible for pushing the system to its capacity limit, but the  
system has high yield strength so that the system can sustain the disturbance-induced  
load and maintain its function. There is a strong sense of ‘elastic behaviour’ in  
65 ③ materials with these definitions. It is argued by Zhang (2007) that these definitions  
lack a distinction of resilience from robustness, and he further defined resilience as  
(Zhang 2007): ‘resilience is a property of the system on how the system can still  
function to a desired level when the system suffers from a partial damage’. The key  
difference of Zhang’s (2007) definition other than these others is the emphasis on  
70 partial damage with the system (or in the sense of materials, a material substance is  
broken – stress over the substance’s ultimate strength). With this definition, Zhang  
(2007) was able to distinguish resilience from robustness as well as from reliability in  
the engineering context, and these distinctions are explained with an example in the  
following (Zhang 2007).

Let us take the desk as an example. The generic function of the desk is to sustain  
an object put on the top of it. A desk has four legs and a top plate, and the legs and  
the top plate are jointed together as a commonly known notion. The desk stands up  
to function in an environment that has some disturbance to the table. *Reliability* of  
75 the desk refers to the life of the desk prior to its ceasing the function under a nominal  
loading and predictable disturbance, and the impact of reliability on the system stops  
at the point of time that the system is damaged. *Robustness* of the desk refers to the  
capability of the desk that performs its function in an acceptable performance range  
under unpredictable disturbance. Robustness has thus impact on the system prior to  
80 its damage and concerns the ‘yield’ or ‘ultimate’ strength of the system or in a more  
general sense, the insensitiveness of the system’s functionality to ‘uncontrolled’  
disturbances (Zhang 2009). It may be clear that the key difference of Zhang’s  
definition of resilience from the others such as Pulley and Wakefield (2001) is that  
Zhang’s definition regards resilience as a system’s post-damage property – i.e. the  
85 system’s ability to recover its function from some damage. Putting the three concepts  
of reliability, robustness and resilience into the engineering context, Table 1 lists their  
semantics in each of general knowledge categories, respectively, for an impression of  
their differences.

In the context of enterprises, Guelfi *et al.* (2008) defined the resilience as the  
capacity of a business process to recover and reinforce itself when facing changes.  
Further, in the context of businesses, Hamel and Valikangas (2003) considered the  
resilience as an ability of firms to change them from failure to success. These  
95 definitions are at the level where the difference between resilience and reliability or  
robustness is not quite apparent; otherwise, they have not made explicit about the

Table 1. Knowledge categorisation of reliability, robustness, resilience.

Knowledge category	Reliability	Robustness	Resilience	
Science	Mechanism of the life of a system	Mechanism of insensitiveness of the structure with respect to disturbances to the system	Mechanism of damage of the system with respect to mishap to the system	100
Analysis	Life of the system	Effect of disturbances on the system function	Function loss due to the damage	105
Synthesis	System concept for a prescribed life	System concept for insensitiveness to a prescribed disturbance environment	System concept for recovery of the lost function	110
Manufacturing	System builder for a prescribed life	System builder for a desired insensitiveness	System builder for a desired recovery capability	115
Management	Operation of the system for a prescribed life	Operation of the system for a desired insensitiveness	Operation of the system for a desired recovery capability	115

difference, despite the fact that the words ‘recover’ and ‘failures’ may indeed imply that a system has something wrong but the wrong is not necessary to the partial damage of the system.

## 2.2. Resilience versus sustainability and healing

Furthermore, resilience may seem to be close to sustainability and healing; but it differs from them. *Sustainability* refers to a system’s property that measures the balanced generation and consumption of the system resource. If we tie a ‘supply’ function to the resource – i.e. the resource plays a role in supplying, we shall view the consumption of the resource as a sort of function ‘loss’. From this view, sustainability merges with resilience in such a manner that the sustainability property measures ‘recovery’ of the lost function, i.e. regeneration of resources. The significant difference between resilience and sustainability also lies in this view; indeed, this view makes, in essence, the sustainable system play only one function – i.e. to supply the resource. The relationship between resilience and sustainability allows the two to learn from each other in developing theories and methodologies for the resilient system and sustainable system, the topic of which is out of the scope of the present article.

*Healing* refers to recovery of a lost function of the system (after damage) by means of the external resource instead of a system’s own resource. Therefore, healing, in the context of engineering, involves replacement or remediation of damaged components (Moslehi and Kumar 2009, Pease 2009, Wu *et al.* 2010). Self-healing is much closer to resilience than healing. However, the self-healing system is more restricted to materials engineering – especially surface engineering. For instance, when a solid surface is broken, a chemical fluid bursts out underneath a damaged surface, forming a ‘new’ layer of solid (Pease 2009). For the system with the architecture that consists of engineered discrete components, which may be

called the macro system, self-healing is relevant to adaptive systems (Bi and Zhang 2008) in such a way that reconfiguration of ‘remaining’ components of a handicapped or partially damaged system to an one that can still function to a desired function tolerance. The adaptive property here is applied to a so-called ‘reduced’ system, where the ‘reduced’ system is the system excluding damaged components.

### 2.3. Resilience versus fault tolerance, safety and dependability

Finally, resilience distinguishes from dependability, fault tolerance and safety. Dependability is a system notion in software engineering, and it is defined as the ability to deliver service that can justifiably be trusted (Avizienis *et al.* 2004). Fault tolerance is a classic notion in software engineering, and it is defined as the ability to deliver service in the presence of faults (Avizienis *et al.* 2004). Fault tolerance is part of the means to achieve the dependability. Fault tolerance relies on two technologies: error detection or diagnosis and recovery, according to Avizienis *et al.* (2004). In fault tolerance, fault does not mean component damage but means errors made at the software development phase and/or errors in the system input at the software operation phase. The absence of component damage is because of the nature of software, which essentially misses the notion of aging and wear or friction. However, software is now also subject to attack, which makes sense for logical damage of components (i.e., codes). The ability to function after software component damage under attack is a kind of the fault tolerance in software and is the resilience of a software system.

The resilient system, especially the resilient software system, will be more dependable. When a system is less dependable or non-dependable and when this may create threat or compromise to human health and life, the notion of safety occurs. Hollnagel *et al.* (2006) considered resilience as a new paradigm for safety. However, Zhang (2007) pointed out that safety is a concept associated with both the system and human. Based on this view, Zhang (2007) proposed a triad (reliability, robustness and resilience) as a new paradigm (RRR for short) for safety, as all of the three are associated with a system’s failure that may further threaten the human’s health and life. Elaboration of the RRR paradigm for safety engineering is out of the scope of the present article.

### 2.4. Summary of the identity of resilience and resilience engineering

In conclusion, resilience has its unique identity or signature. Resilience is a system property that has been less known in system design, manufacturing and operation management. The resilience property is critical to the functioning of an engineered system including enterprise information systems. The next section puts forward some of our thoughts on general principles that may be applied to designing an engineered system to be more resilient.

## 3. Principles of design of resilient systems

We shall explore the principles of design of resilient systems. Our strategy is to learn from the biological system. We conclude four axioms related to the resilience of the engineered artefact system, learned from the biological system.

- Axiom 1:* A damaged component of the artifact system can never be fully recovered to its original one without external intervention.
- Axiom 2:* A component or a part of the component of the artifact system for function-A may be trained to do function-B of another component.
- Axiom 3:* There will always be a distinct identity of the embodiment of the artifact system that includes a control system and physical entity at one time in one place to perform one distinct function.
- Axiom 4:* A system's failure is an emergent consequence of the system's internal vulnerability and system's external mishap.

Several remarks follow these axioms. Axiom 1 implies that when a component is damaged, it can never recover to its original physical embodiment on its own. This situation is different from the situation of the biological system. In the biological system, cell can be divided into other cells, and these other cells may grow to be the same as the original cell. In that sense, the biological entity may be said to be able to recover to its original through the cell division principle. Some work on self-healing in the context of material engineering (Pease 2009) can be thought to mimic this principle. However, according to Pfeifer *et al.* (2007), the true replication of systems has not been achieved for engineered dynamic systems such as robots. In this article, we do not include the cell division principle.

Axiom 2 is learned from the biological system such as brain plasticity (Doidge 2007). Brain plasticity refers to a property of the human brain that the elements in the brain have a many-to-many relation to its functions (Figure 1a) and furthermore, such relation may be altered by training the brain elements (Figure 1b). Further, putting together Axiom 2 and Axiom 3, we imply the following scenario where a component or system may have one part to perform one particular function while have another part to perform another function; however, the two parts are distinct to each other (though they may partially share some structure). In the biological system, in particular taking a human as entire entity, we see that the human performs multi-functions in one place at one time. However, such multi-functionality is not beyond the distinct identity of entities to perform multiple distinct functions.

From the four axioms as discussed above, we shall propose five principles of design of resilient (artifact) systems, and they are discussed in the following.

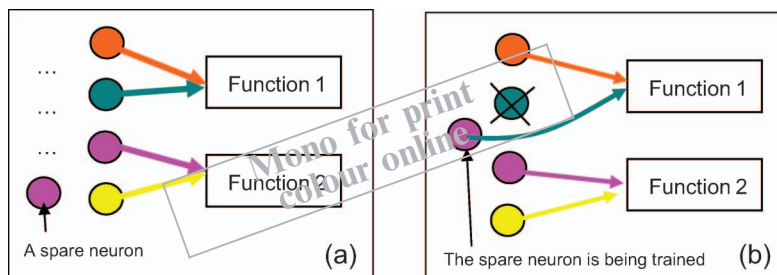


Figure 1. The neuron system and brain plasticity ((a) neurons play function 1 and function 2; (b) one neuron is damaged, a spare neuron is trained together with the other one to play function 1 which is lost due to the damage of the neuron).

*Principle I:* A resilient system should be designed to have a certain degree of redundancy preferably functional redundancy. The more redundancy the system has and the higher degree of resilience.

*Principle II:* A resilient system should be designed to have a special controller which is responsible for (1) redundancy management and (2) function learning.

*Principle III:* A resilient system should be designed to have a sensor which is responsible for both timely and spatially (1) monitoring the system's function and performance, (2) monitoring the utilisation of the system's capacity and (3) monitoring the system's demand.

*Principle IV:* A resilient system should be designed to have a predictor which is responsible for (1) predicting potential threats to the system and (2) analysing potential vulnerabilities of the system.

*Principle V:* A resilient system should be designed to have an 'actuator' along with a 'physical' entity which is responsible for (1) implementation of changes of the system both in cognitive and physical domains and (2) implementation of trainings of one component or sub-system of the system to perform a new function.

Several remarks regarding the design principles are worth discussion. In Principle I, function redundancy is a system's property that a particular function of the system or its subsystem or its component can be performed with more than one system configuration, and physical redundancy is a specialised functional redundancy in the way that physical redundancy is the duplicate of components. Configuration describes a set of component instances and their connection instances (Bi and Zhang 2008). Components are of both hardware and software. The functionality will be viewed from two different angles: one is generic in the sense that performing a particular function is regardless of a system's demand or task load and the other is specialised in the sense that performing a particular function is associated with the context of a particular task load and a particular environment. We shall revisit the desk example as we previously discussed to illustrate the concepts regarding the generic and specialised function. Figure 2 shows two different configurations of the desk: one with four legs (Figure 2a) and the other with three legs (Figure 2b). The four-leg desk possesses a degree of function redundancy; if one leg is broken in the four-leg configuration (Figure 2a), the reconfiguration of the desk into a three-leg desk (Figure 2b) may make the partially damaged desk still function (Figure 2b) – i.e. to perform its generic function regardless of task load and its specialised function with regards to task load. From the equilibrium principle in mechanics, it is easily seen that for the three-leg configuration, when the load is sufficiently heavy, the whole desk may tilt about the line AB (Figure 2b) and thus does not perform its specialised function, a function under a particular load and in a particular location, at all.

Redundancy management in Principle II includes: (1) decision making for the system to be reconfigured in a particular manner to fulfil a system's lost function and (2) decision making for the system to be trained to work with a new configuration. Learning is meant that a component may need training to fulfil a new role.

The five principles are related to each other and they together guide the design process towards a full resilient system. While Principle I leads that a system has a capability of functional redundancy, Principle II leads that a system knows how to

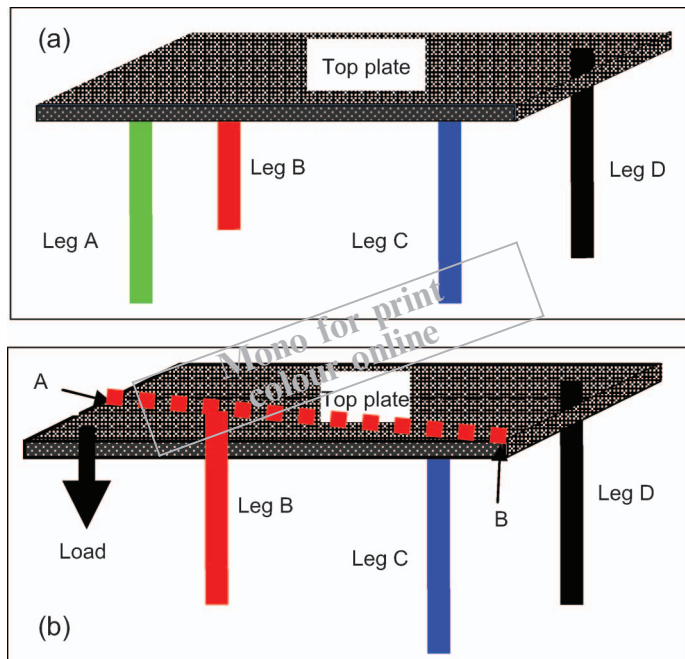


Figure 2. Reconfiguration to perform functions ((a) four legs; (b) one leg is damaged but the system is reconfigured to have three-leg desk to function).

make use of this capability. When a system is in operation, the system performs a function or task with a particular configuration. Principle III leads that the system knows what is required, how well the system performs and what reconfiguration capacity the system possesses. All this information is necessary for Principle II to work. In Principle IV, the idea of putting together two concepts, namely external threat and internal vulnerability, is derived from Axiom 4. Principle IV enables the implementation of a preventive strategy for design and a predictive strategy for operation management. Principle V leads that the system can take action to realise changes from one configuration to another one and to exercise training of a component to perform a new function.

The axioms and definition of resilience engineering are the source to derive these design principles. Principle I has a root from Axiom 1 and Axiom 2 and Axiom 3. Principle II has a root from the definition of resilience – in particular self-management. Principle III has a root from the definition of resilience – in particular self-diagnosis. Principle IV has a root from Axiom 4. Principle V has a root from the definition of resilience – in particular self-organisation. Figure 3 shows the relationship among the principles and the relationship between the principles and axioms.

A full resilient system should be designed and managed by having the structure and operation management based on the proposed five principles. Generally, a full resilient system must have software to follow Principle II and Principle IV. Together with hardware and software, the architecture of a resilient system is of a network with further characteristics being modular, adjustable and sensible in system hardware and energy or resource. In short, a resilient system must be ready for both

its structure and resource to change or reconfigure, temporally and spatially, to perform the required function with a function tolerance.

Practical resilient systems may be partial in the sense that they may not have a full list of the aforementioned characteristics of the resilient system. Figure 4 shows a partial resilient machine (Zhang 2008). The machine has two servo motors, four moving components and one ground frame. The machine performs a task to follow a desired trajectory at point P. There is software with this machine system, and the software system is responsible for real-time instruction of the supply of energy (electronic current with constant voltage) to the motors to drive the four moving

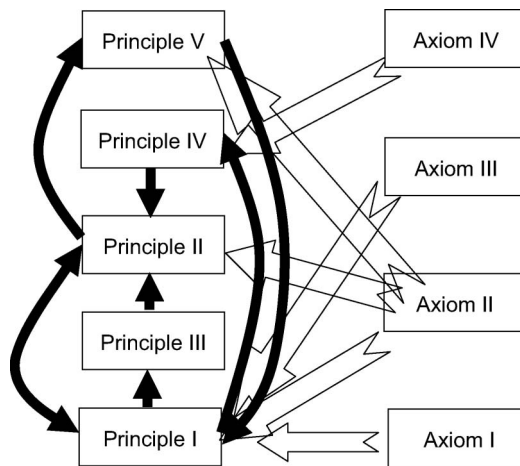


Figure 3. The relationship among the principles and the relationship between the principles and axioms.

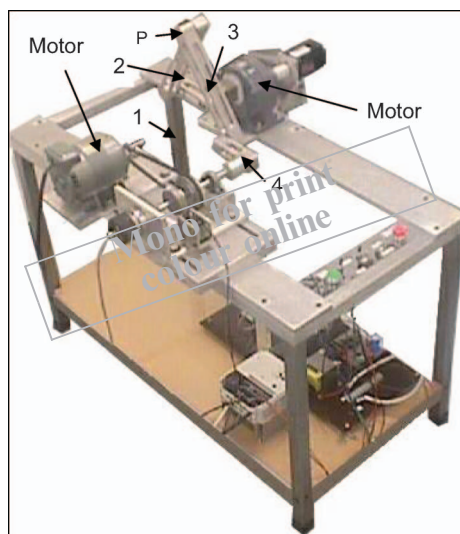


Figure 4. A resilient machine (four moving members 1,2,3,4, and two servo motors).

components to fulfil the task – i.e. to generate a desired trajectory at point P. Suppose one servo system is partially broken at one point of time; in particular the partially broken servo motor is assumed to still function as a constant velocity motor. As such, the system has one constant velocity motor and one servo motor, and such a system is called hybrid actuation system (Ouyang *et al.* 2005, Wu *et al.* 2005). It is known that the hybrid actuation system can still perform the required function of the original machine with two servo motors by reprogramming the control software of the system (Ouyang *et al.* 2005). Such software serves as redundancy management under Principle II.

#### 4. The architecture of resilient enterprise information systems

In this section, we shall propose the architecture for a resilient enterprise information system by applying the proposed principles. We first view an enterprise information system as a system that includes two subsystems: infrastructure and data. The infrastructure system includes hardware and software and human operators, where the software is not data that represent information but data that manage the information, such as DBMS. The data system includes data that represent enterprise-specific information. According to the design principles for resilient systems, the enterprise information system should have the following components.

##### 4.1. Component (1) – redundancy infrastructure system and data system

These two systems should be designed with the mind of Principle I – that is redundancy. In particular, redundancy between the infrastructure and data system may be designed. For instance, two different email management systems (webmail, outlook) are in place to deal with the same email data. When one email management system is wrong, we can switch to the other email management system. Redundancy in the data system may be designed. Information can be modularised into Infor-A, Infor-B, Infor-C, etc. Their dependency may be designed, e.g. Infor-C is dependent on Infor-A and Infor-B. The specific dependency relation may be represented or perhaps, the fact of their dependency together with the principle that governs the dependency is represented only. Redundancy among infrastructure systems may be designed with a particular feature of enterprise information systems being that human operators and machine systems may share functions – functional redundancy. For instance, a material resource planning function may be done by software but may also be done by a human planner.

##### 4.2. Component (2) – redundancy management and function learning

The roles of redundancy management include: (1) maintaining a knowledge base of configurations, their functions, and functional capacities and (2) determining an optimal configuration of the enterprise information system per se. For instance, a hypothetical enterprise workplace has two email management systems (Webmail, outlook). The knowledge that these two systems can perform the same function is available to the redundancy management system. Each of them has a capacity limit which should also be known by the redundancy management system. It may happen that the outlook system may fail to work because its ‘sent-mail’ function at the local hard disk fails (e.g. over its storage capacity limit). It is noted that each time an email

is sent, a copy of it is put into the ‘sent-mail’ box. If in that particular situation, the capacity of the ‘in-mail’ box has some room, the solution to recover the function of the outlook management system is to make ‘copy’ to the ‘in-mail’ box and remove mails from the ‘sent-mail’ box; in this case, the ‘in-mail’ box serves as dual roles: receiving email and retaining a copy of the sent mail. Here, knowing the capacity of the ‘in-mail’ box is the key to the solution.

The roles of function learning include: (1) determining ‘learners’ which can be physical entities (e.g. a piece of software) or human operators, (2) determining learning targets (e.g. database manager) and (3) executing the learning procedure. For instance, a DBA is sick, and a DBA assistant can be trained to perform a part of the DBA’s role temporarily. Another example may be such that a CRM is of dysfunction; in this case, a spread-sheet program may be ‘revised’ (a kind of training) to function (partially) as a CRM (e.g. record customer’s information).

It is worth mentioning that the contemporary enterprise information system is of quite redundancy. For instance, the ERP system has been extended beyond its initial premise of business financial management to nearly all business functions such as material resource planning, customer relation management and so on. Some of them may also be available in other software systems or in a stand-alone status. However, such redundancy has not been paid attention to in the current business practice.

#### **4.3. Component (3) – monitoring the function and performance**

The scope of monitoring in the context of enterprise information systems is: (1) the function, performance and status-quo of hardware and software based on their technical specification, (2) the semantics of data per se and (3) the performance of information delivery in terms of the discrepancy between information demand and information supply. The monitoring could be done by the human operator or physical entity. For semantic-rich performance information such as scope (3), the monitoring may largely be done by the human operator.

#### **4.4. Component (4) – vulnerability analysis and demand forecast**

The scope of the predictor in the context of enterprise information systems is: (1) analysis of the vulnerability of the infrastructure system in terms of information management, (2) analysis of the vulnerability of the data system in terms of enterprise business processes, (3) analysis of external threats on both the infrastructure and data systems and (4) analysis of human errors in enterprise information systems from both perspectives: infrastructure operation and information utilisation.

#### **4.5. Component (5) – learning and training execution system**

The scope of this system includes: (1) reconfiguration of a partially damaged or supply-short-of-demand infrastructure system, (2) reconfiguration of a partially damaged or supply-short-of-demand data system and (3) training of one component to learn the function of another component; the component or subsystem here includes the human operator.

In summary, Figure 5 shows the relationship among these components. The relationship map is a specialised one of the generic one shown in Figure 3.

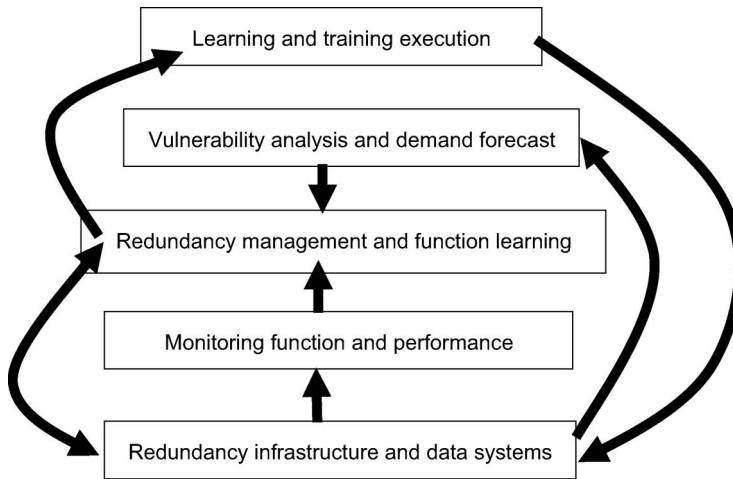


Figure 5. The relationship among the major components proposed for the resilient enterprise information systems.

A practical enterprise information system may only be a partial resilient system owing to absence of one or more components of the five proposed components. The constraint to develop a full resilient enterprise information system is development cost.

## 5. Concluding remarks

This article elaborated on the distinct identity of resilience. Resilience is a property of the system, and it has great implication to humans in terms of safety, health and well-being. Resilience engineering goes beyond reliability engineering and robust engineering. Resilience has a root in biological and ecological systems from which engineered resilience systems can learn. This learning can lead to four axioms as proposed in this article, which can subsequently be used to derive five design principles for resilient systems, as proposed in this article. The design principles are well applicable to enterprise information systems, leading to the proposed architecture for a resilient enterprise information system. This architecture provides a guideline for further developing and implementing practical enterprise information systems to be more resilient.

## Acknowledgements

The authors acknowledge a financial support to this research received from the Natural Science and Engineering Research Council of Canada (NSERC) through a Strategic Project Grant program.

## References

- Avizienis, A., *et al.*, 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable Secure Computing*, 1 (1), 11–33.
- Bi, Z.M. and Zhang, W.J., 2008. On a general architecture of adaptive robotic systems. *International Journal of Robotics and Computer Integrated Manufacturing* (accepted in 2008).
- Doidge, N., 2007. *The brain that changes itself*. A James H. Silberman Book.

- 540 ⑦ Guelfi, N., *et al.*, 2008. *SERENE'08: Proceedings of the 2008 RISE/EFTS Joint International*  
545 ⑧ *Workshop on Software Engineering for Resilient Systems*. New York, NY, USA: ACM.
- ⑧ Hamel, G. and Valikangas, L., 2003. The quest for resilience. *Harvard Business Review*.
- Hollnagel, E., Woods, D.D., and Leveson, N., 2006. *Resilience engineering: concepts and precepts*. Aldershot, UK: Ashgate.
- Moslehi, K. and Kumar, R., 2009. *Vision for a self-healing power grid Infrastructure for a self-healing grid* [online]. Available from: [www05.abb.com/global/scot/scot271.../21-25%204M673\\_ENG72dpi.pdf](http://www05.abb.com/global/scot/scot271.../21-25%204M673_ENG72dpi.pdf).
- 545 ⑨ Ouyang, P., *et al.*, 2005. Design, modelling and control of a hybrid machine. *Mechatronics*, 14, 1197–1217.
- ⑩ Pease, R., 2009. *Self-healing rubber bounces back*. *BBC Radio Science Unit* [online]. Available from: <http://news.bbc.co.uk/2/hi/7254939.stm> [Accessed 3 January 2009].
- Pfeifer, R., Lungarella, M., and Iida, F., 2007. Self-organization, embodiment, and biologically inspired robotics. *Science*, 318, 1088–1093.
- 550 Pulley, M.L. and Wakefield, M., 2001. *Building resiliency: how to thrive in times of change*. Center for Creative Leadership.
- ⑪ Wu, F.X., *et al.*, 2005. Control of hybrid machines with 2-DOF for trajectory tracking problems. *IEEE Transactions on Control Systems Technology*, 13 (2), 338–342.
- 555 ⑫ Wu, W., *et al.*, 2010. Direct-write assembly of biomimetic microvascular networks for efficient fluid transport. *Soft Matter* 2010. DOI: 10.1039/b918436h.
- Zhang, W.J., 2007. *Is resilience a destiny of safety management paradigm?* [online]. Presented at a seminar at the East China University of Science and Technology. Available from: <http://homepage.usask.ca/~wjz485/Other%20Publication.htm> [Accessed 3 November 2009].
- Zhang, W.J., 2008. *Resilience engineering: overview* [online]. Presented at a seminar at the Chinese Natural Science Foundation. Available from: <http://homepage.usask.ca/~wjz485/Other%20Publication.htm> [Accessed 3 November 2009].
- 560 Zhang, W.J., 2009. *Robust design. Lecture note for advanced engineering design method*. Department of Mechanical Engineering, University of Saskatchewan, Canada.
- 565
- 570
- 575
- 580
- 585