Quality Modelling and Metrics of Web-based Information Systems

Yanlong Zhang

A thesis submitted in partial fulfilment of the requirements of the Oxford Brookes University for the degree of Doctor of Philosophy

Department of Computing
School of Technology
Oxford Brookes University
United Kingdom

June 2005

Acknowledgments

The author is greatly indebted to his supervisors, Professor Hong Zhu and Dr. Sue Greenwood. The author would like to gratefully acknowledge their contributions for the completion of this PhD thesis. They have devoted invaluable guidance, advice and time to this study.

The author would like to thank his examiners, Professor Hongji Yang, De Montfort University and Professor David Duce, for providing invaluable comments on this thesis.

The author wishes to thankfully acknowledge help and support from all colleagues in the Department of Computing, Oxford Brookes University, especially John Nealon, the Head of the Department, Sue Flint, School Administrator and Qingning Huo, a PhD student.

The author wishes to express his gratitude to colleagues in Department of Computing and Mathematics, Manchester Metropolitan University, in providing respectively support for this study. Furthermore, the author would like to thank all the colleagues and students who carefully completed my questionnaires and attended interviews.

Lastly, the author's family, his mother, wife Sophie and son Howard are sources of encouragement, and part of his success.

Abstract

In recent years, the World Wide Web has become a major platform for software applications. Web-based information systems have been involved in many areas of our everyday life, such as education, entertainment, business, manufacturing, communication, etc. As web-based systems are usually distributed, multimedia, interactive and cooperative, and their production processes usually follow ad-hoc approaches, the quality of web-based systems has become a major concern.

Existing quality models and metrics do not fully satisfy the needs of quality management of Web-based systems. This study has applied and adapted software quality engineering methods and principles to address the following issues, a quality modeling method for derivation of quality models of Web-based information systems; and the development, implementation and validation of quality metrics of key quality attributes of Web-based information systems, which include navigability and timeliness.

The quality modeling method proposed in this study has the following strengths. It is more objective and rigorous than existing approaches. The quality analysis can be conducted in the early stage of system life cycle on the design. It is easy to use and can provide insight into the improvement of the design of systems. Results of case studies demonstrated that the quality modeling method is applicable and practical. Practitioners can use the modeling method to develop their own quality models.

This study is amongst the first comprehensive attempts to develop quality measurement for Web-based information systems. First, it identified the relationship between website structural complexity and navigability. Quality metrics of navigability were defined, investigated and implemented. Empirical studies were conducted to evaluate the metrics. Second, this study investigated website timeliness and attempted to find direct and indirect measures for the quality attribute. Empirical studies for validating such metrics were also conducted.

This study also suggests four areas of future research that may be fruitful.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Study aim

In recent years, the World Wide Web has become a major platform for software applications. The WWW is having phenomenal impact on business, industry, finance, education, government, entertainment, and many other sectors as well. The Web's initial scope as a way for information sharing and exchange has been significantly extended, and a whole range of new applications is emerging in the Web environment. The Web now also serves as a platform for many distributed applications. The original document-oriented model of the Web is being extended and enhanced by initiatives such as the Semantic Web and Web Services to cater for advanced applications.

Web-based information systems (WIS) have become the major research focus in information systems engineering practice (Nielsen, 1999). As the nature of Web-based information systems is usually distributed, hypermedia, autonomous and cooperative (Zhu, et al, 2000), such systems have posed a number of significant challenges to the research community, such as faster time-to-market, increased concerns for security and performance, open and evolving software architectures, dynamically composed electronic services, etc. The pervasive connectivity of the Internet and the powerful architecture of the WWW have created a tremendous opportunity for conducting business on the Internet. The terms e-commerce and e-business have been used to describe those systems and technologies that make conducting business on the Internet possible (Chan et al, 2001).

1.1.1 Problems and challenges

Generally speaking, WIS development is still mostly ad hoc, and the lack of disciplined and systematic approaches, causes concerns about their maintainability, quality and reliability (Nielsen, 1996; Powell, 1998). It often neglects appropriate established practices from other disciplines such as Hypermedia and HCI to create, manage, and reuse structures of the information space and enhance the end user experience. And in most cases, Web development suffers from its biggest potential: new technologies and devices allowing for ubiquitous use of these applications (Powell, 1998).

It is worth briefly reviewing the 'software crisis' forty years ago (Gilb, 1988). Traditionally, software applications were largely hand-crafted. Tools were focused around implementation and performance issues. Almost no process modelling or management existed. Metrics were primitive or non-existent. As software grew in scope and complexity, this early approach broke down. Many applications failed to meet the specifications and were completely unmaintainable. The phenomenon was called the 'software crisis'. These issues have been addressed through the development of software engineering discipline, including development of methodologies, process modelling and management and quality assurance techniques. Although it is still too early to say that all the problems have been solved, software has a much higher quality today.

Unfortunately, WIS development is currently at the stage software development was forty years ago (Lowe and Hall, 1999). Most hypermedia applications are developed using an ad hoc approach. There is little understanding of development process, application quality and project management. The focus is much on the technical issues such as interface implementation. However the quality of the end-product is rarely seriously considered and clearly understood.

Current approaches to developing websites are, in many cases, failing to deliver applications with acceptable quality, especially in terms of information access and usability. As WIS grow in scope and complexity, current practice will need to develop or risk failing to deliver. The problems that developers are liable to suffer if this crisis is not addressed include:

- failure of satisfying users' requirements
- failure of providing the access to the desired information

- failure of assisting in the use of this information
- poor maintaining of the systems
- poor cost-effectiveness in the production of WIS

Indeed, the problems parallel those of the software crisis.

WIS has imposed a great challenge to the current theories and methods of software quality management (Zhu, et al, 2000). This may cause the 'web-based information systems crisis' like the 'software crisis' forty years ago. The main reason is the lack of management methodologies and techniques. To the best of the author's knowledge, there is not a widely accepted quality model for WIS, and even few efforts have been made on understanding and measuring WIS quality.

1.1.2 Research context

Before setting the research objectives of the thesis, it is necessary to discuss the related research areas which constitute the context of the study.

1.1.2.1 WIS engineering

Engineering is concerned with producing artefacts that are correct in respect of their requirements and specifications, which work reliably, to a stated cost, to a stated level of quality, and where delivery is complete by a stated time. Software engineering is concerned with using engineering methods in the production of software in order to achieve these aims (Sommerville, 2001).

The differences between software engineering and WIS engineering exist (Powell, 1998; Lowe and Hall, 1999). First, as WIS will continue to be document-oriented, they will continue to be more content-driven. Static documents will not disappear when Java or other web programming technologies become commonplace. The creation of document content does not have the same requirements as common software applications. Second, WIS will continue to be time sensitive. Many websites are driven by marketing goals. Short time to market and timely information provision have become two main lifelines of a website. Thirdly, WIS will continue to be style driven. Websites will continue to be very focused on look and feel. The emphasis on the creative acceptability of websites is generally not well considered by the software developers, except possibly by game developers. Finally, and more importantly, WIS is software but not traditional software. The platform of the Web is not as well understood as the traditional software platform. While many developers may feel that WIS development is nothing more than glorified client/server, the Web is generally more unpredictable. It has structural problems, such as outside influences and user perceptions that make development difficult. This is probably the most significant difference between WIS development and traditional software development.

1.1.2.2 Quality systems engineering

Quality systems engineering, where quality modelling and measurement play the central role, is an essential part of systems engineering practice.

The International Organisation for Standardisation (ISO) defined quality as "The totality of features and characteristics of a product or service that bear on its ability to satisfy specified or implied needs." (ISO, 1986). This definition provides a specific and objective view of the notion of quality. It is the key to being able to measure quality. It is based on two points:

- Quality is the ability of the product or service to fulfil its function
- Quality is achieved through the features and characteristics of the product or service.

In other words, quality is to meet specific requirements, and is associated with having the required range of attributes and achieving satisfactory performance within each attribute. The definition has been widely accepted and considered applicable to software quality and information systems' quality.

In a general theory of quality management, Garvin (1984) suggested the following five different views of quality:

- The user-based view

- The product-based view
- The manufacture's view
- The value-based view
- The transcendent view

In order to compare quality in different situations, both qualitatively and quantitatively, it is necessary to develop a model of quality. A quality model is the basis of measurement. Without a proper model, a measure will become a collection of meaningless numbers. By quality model it means a model of a type of products or services by which the quality of a particular product or service of the type can be understood, analysed, measured, predicted and compared with other products or services of the same type. In this project, WIS form the type of products under consideration.

Because quality is really a composite of many characteristics, the notion of quality is usually captured in a model that depicts the composite characteristics and their relationships. There are many models proposed for information systems and software systems. Most of them are hierarchical in nature. These models are useful in articulating what people think is important, and in identifying the commonalties of the view. It is suggested that a complete quality model should include the following (Eriksson and Torn, 1991):

- A structure of quality characteristics
- Measuring attainment of quality goals
- Deciding on quality procedures
- Quality assurance
- Management of the quality model

The structure of quality characteristics and measurement issues lay the theoretical foundation for the other aspects.

1.1.2.3 Metric and measure

The terms metric and measure are not used consistently by different authors, and there is some overlap in usage. According to National Institute of Standards and Technology (NIST, 2004), the term *measure* is usually used for 'more concrete or objective attributes' and *metric* for 'more abstract, higher-level, or subjective attributes'. It also provides some examples. Lines of code (LOC) is objective and concrete, and thus a measure. Robustness and quality are hard to define objectively, therefore these are metrics.

In (Ragland, 1995), the difference between measure and metric is illustrated using an interesting example, where 'body temperature' can be called a measure and 'health' a metric. It backs NIST's statements, as 'body temperature' is an objective attribute and 'health' a subjective one.

In (Wikipedia, 2005), a software metric is defined as 'a measure of some property of a piece of software or its specification'. IEEE (1990) gives a more formal definition of a metric as 'a calculated or composite indicator based on two or more measures. A quantified measure of the degree to which a system, component or possesses a given attribute. Suppose we wish to assess the quality of a C compiler. We decide to assess the quality in terms of 'compiling speed' and 'resource needed'. For measuring compiling speed, we choose to use LOC per second, and we measure the LOC and compiling time. For measuring resource, we choose the bytes of RAM required. It is easily seen that LOC, time and bytes of RAM are objective attributes and thus they are all measures. However, when we propose the quality measurement, it is a metric, because our notion of quality defined in terms of two sub-attributes compiling speed and resources needed are subjective. It is also difficult to decide how to decide the weights of each sub-attribute.

This thesis uses the terms measure and metric in the senses defined by NIST above, though quotations from other authors are direct quotations and do not necessarily always follow these senses. This study will focus on quality measurement of two attributes of WIS: Web navigability and timeliness. Navigability is a more subjective attribute, because it reflects users' feeling, and different users (age, gender, computing background, etc.) may perceive the same site's navigability differently. Thus it is a metric. According to the definition of Website Timeliness (see Chapter 5), timeliness is the time difference between an event occurs and it is published. It is a more objective attribute, and thus a measure.

1.1.3 Aims and objectives

This research is trying to apply quality systems engineering principles to WIS. The objectives of the research are:

- to develop a quality modelling method which can be used to construct quality models of WIS.
- to develop and evaluate navigability metrics and timeliness measures.

1.2 Research methodology

This section describes how this study is carried out and the research methods used in this study.

Quality is defined as a list of quality attributes or characteristics that are associated with certain attributes of software. On one hand, it is of prime importance to define or identify quality attributes and their relationships, as quality cannot be built into a product, or measured until it is properly defined. On the other hand, measuring attributes in precise quantitative terms has been recognised as a crucial stage for enhancing understanding (Dhyani, et al, 2002). Statements like 'the systems looks good' or 'that system may not be very good' are too vague to be of much value. Quantifying quality evaluations, or developing quality metrics, answers such concerns.

This study therefore consists of two interrelated parts: the development of quality models of WIS (Chapter 3) and the development of navigability metrics (Chapter 4) and timeliness measures (Chapter 5). These two parts are closely related and neither can be neglected.

First, quality models are the basis of quality metrics and measures. Only after the question 'what to measure' is correctly answered, can the question 'how to measure' become meaningful. Without a correct model, any measurement results would be useless collections of numbers (Fenton, 1994). Second, quality measurement can significantly improve the applicability and usability of quality models. Quality models can be regarded as a set of quality attributes and their interrelationships, and quality measurements provide qualifying rating of each attribute. All the results can then be used for evaluation or prediction purposes (Shepperd, 1995).

In the study, it is found impractical to define a generic quality model for all kinds of WIS as they are quite different in terms of diversity of information and services provided. Developing a quality modelling method to enable software engineers to define their own quality models according to the application and its design of the system turned out to be a solution. The existing quality modelling methods are mostly based on the skills and experiences of quality model developers, thus subjective and difficult to validate. Based on Dormey (1995)'s research, in which the quality carrying property and quality attribute should be closely related, this study develops a new quality modelling method, which consists of WIS architectural analysis and system hazard analysis. The rational and details are in Section 3.3.

There are a number of quality attributes for WIS. In Chapter 3, a quality model for e-commerce systems is developed utilising the quality modelling method. More than ten quality attributes are identified. Unfortunately it is impossible to cover the measurement of all the attributes in this study. Therefore, this study focuses on two interesting aspects, navigability and timelines. These quality attributes are of particular importance in many WIS, but they have not been thoroughly investigated in the literature. Measures and metrics of other attributes will be described in Chapter 6, suggestions for future work. As discussed in Section 1.1.2.3, navigability is a more subjective attribute, whereas timeliness is a more objective one. To distinguish the terms, this study will use navigability metrics and timeliness measures accordingly.

WIS navigability is concerned with the users' feelings and perceptions of websites. It is thus a subjective concept and the design of such measurements will involve human-centered evaluation methods, such as questionnaire methods. As the author is working at a university, it seems much easier to recruit university students as the subjects. Thus university websites navigability measurement is investigated.

WIS navigability measurement consists of two steps. First, based on the hypothesis of a strong correlation between website structural complexity and navigability, this study develops some structural complexity metrics, which are then evaluated using adapted Weyuker's axioms. Second, empirical studies of the navigability of university websites were conducted using a questionnaire method to further evaluate the correlationships between website structural complexity and navigability. To test the reliability of the results, the questionnaire test was repeated with different subjects at different websites. The choice of university websites as the particular type of WIS for empirical studies of navigability was based on the following considerations.

(a) The students of the university where the author of the thesis is working at were available as the subjects. They had relatively similar computing skills and background knowledge and familiarity with the websites.

(b) These websites were not changing as frequently as some other types of WIS such as web-based media. Therefore, the experiments were not unnecessarily complicated due to the changes on the contents and structures of the websites during experiments. Repeated experiments to eliminate bias were hence possible.

(c) Browsing is the prime method of finding information in such websites. Therefore, navigability is of particular importance in the design of such websites. The navigation is also non-trivial due to the great amount of information and diversity in contents and formats contained in such websites.

(d) Finally, such websites are usually developed with professional and careful designs rather that students projects. They are representative in terms of high quality at a similar level. They provided a set of good samples to test whether the metrics are good enough to detect the differences between them accurately.

It is worth noting that the empirical study of navigability metrics does not use any particular properties of the university websites, such as their structure and contents.

Therefore, the results obtained in this study should be equally applicable to other types of WIS.

WIS timeliness is another very important quality attribute of media-based websites. Based on the timeliness concept in Chapter 5, three timeliness measures are developed and their properties were studied. Empirical evaluations of the timeliness measures are also conducted. Since some WIS update with a very low frequency, even remaining unchanged for several years (Fleming, 1998), it is thus important to choose the right kind of sites as the objects to study in order to evaluate the measures effectively and efficiently. Web-based media websites were chosen based on the following considerations:

(a) Media-based websites have been proven to update frequently from our experiments. Timeliness was believed to be an important quality attribute of this particular type of WIS. It is not a trivial issue for such websites.

(b) Rival web-based media tend to compete on reporting news events in a timely manner. Groups of rival media of similar background provide samples of subtle differences in terms of timeliness so that repeated experiments can be performed with minimal bias. They are good subjects to test whether timeliness measures are good enough to distinguish them accurately.

(c) Such websites are run by well-organised organisations and developed by professionals. The techniques used in the implementations of the websites are mature and usually applied systematically. Therefore, such samples are stable and representative for empirical study. Their timeliness attributes are not random values. Unnecessary bias and complexity can be easily eliminated in the empirical studies.

1.3 Main contributions

1.3.1 Quality modelling method

This study discuses the major drawbacks of the existing quality models (for detailed discussions see Chapter 2). It also concludes that it is infeasible to develop a generic quality model suitable for all types of WIS. For example, security is a very important quality attribute for e-commerce systems, but not that important for a student's personal homepage. Different users often have their own quality concerns and decide which attributes are the most important for them. A new quality modelling method is developed in this study so that a quality model for specific WIS can be derived from the architecture of the systems. The case study shows that the quality modelling method is objective, applicable, and adoptable in early design phases. Users can apply the method to develop their own quality models.

1.3.2 Quality measurement of WIS

There are many areas where measures are not available, and even they are, they are not always validated. This is also quite true for WIS. Therefore, quality metrics of WIS need to be developed for either predictive or comparative evaluation purposes. In this study, navigability metrics of university websites and timelines measures of media-based websites are identified and defined. They are also evaluated against formal and/or empirical studies.

1.4 Organisation of the thesis

The remainder of the thesis is organised as follows:

Chapter 2 Literature Review

This chapter reviews related research on quality models and measurement. In the literature, numerous quality models have been proposed and investigated. In recent years, with the rapid growth of the World-Wide Web and the Internet, new quality requirements are imposed on information systems. As a result, WIS quality models are emerging. This chapter contains the description, comparison and discussion of the most well known and influential works in the area.

Measurement is central to all engineering disciplines. In this chapter, theoretical foundations of measurements are reviewed in the context of design, implementation and validation of quality metrics. A number of software metrics, related to this study, are reviewed.

Chapter 3 Quality Modelling of Web-based Information Systems

This chapter consists of two parts. It first presents a method for modelling the quality of WIS. It then applies the method to develop a quality model of e-business systems through a case study.

Chapter 4 Measuring website navigability

One of the biggest issues of website usage is how to help the users find the required information efficiently and effectively. Website navigation design is therefore crucial for website usability. It is identified as one of the quality attributes of WIS. This chapter presents some website structural complexity metrics, which can be used to measure indirectly website navigability. Software tools are developed to implement metrics for the measurement of a website's navigability. Empirical studies are also conducted to validate the metrics.

Chapter 5 Measuring website timeliness

Timeliness is another important quality attribute for websites. Few examples of research into timeliness measurement can be found in the literature. This chapter presents an automated approach to measuring timeliness. A prototype system has been designed and implemented to realise the metrics. Empirical studies are also conducted and results reported.

Chapter 6 Conclusion

This chapter concludes the thesis with a summary of the findings and discussions of the directions for further research.

Chapter 2 Literature Review

2.1 Introduction

One of the principal objectives of software engineering is to improve the quality of software products (Fenton and Pfleeger, 1997). The abstract notion of software quality can only be measured when it is broken down into a number of specific measurable software attributes. These quality attributes are not independent of each other. A number of software quality models have been proposed to establish frameworks for classifying software quality attributes and to understand the relationships between them. The most well-known software and information system quality models include McCall's model (McCall, et al, 1977), Boehm's model (Boehm, et al, 1978), the ISO 9126 standard quality model (ISO, 1991), and the SOLE model (Eriksson and Torn, 1991). The interrelationships between quality attributes have also been investigated in Perry's Model (Perry, 1987) and Gillies's Model (Gillies, 1992; 1997). Metrics for quantitative measurement of many of the identified quality attributes have also been established.

In recent years, the World Wide Web has become a major platform for software applications. Although WIS also share some common characteristics with conventional program-based software systems, some special features have been identified in the literature, see e.g. (Zhu et al, 2000; Hearst, 1998; Hall, 1999). The differences between the WIS and the conventional software systems are discussed in (Lindroos, 1997) from the perspective of software quality. Therefore, it is widely recognised that the existing quality models and metrics for quality attributes cannot fully satisfy the requirements for measuring quality of web-based systems. Efforts have been made to develop new quality models for WIS (Olsina, et al, 1999; Miller, 2000; Brajnik, 2001).

Quality measurement is usually expressed in terms of metrics. Abundant software quality metrics can be found in the literature. For software complexity metrics, the most important ones are Halstead's software science (Halstead, 1977), Cyclomatic complexity metric (McCabe, 1976), system positioning measures (Belady and Evangelisti, 1981), information flow metrics (Henry and Kafura, 1981) and stability measures (Yau and Collofello, 1980).

Dhyani, et al (2002) surveyed the existing research on Web metrics and developed a taxonomy of Web metrics. In the graph properties approach, researchers use a Web graph model to reflect the structural organisation of the hypertext. Web metrics are derived based on graph theory, such as centrality, global and local measure (Botafogo et al, 1992).

This chapter will survey the existing research in quality models and quality metrics. The rest of the chapter is organised as follows: Section 2.2 surveys and summarises the conventional software quality models. Sections 2.3 reviews and assesses the information systems quality models. Section 2.4 surveys and discusses the WIS quality models. Section 2.5 reviews measurement theory which will be used in the study. Section 2.6 surveys and discusses software metrics and Section 2.7 will focus on the website metrics review. Section 2.8 will summarise the chapter and pose open problems to be addressed in this study.

2.2 Software quality models

As an important part of modern information systems, many researchers have studied software's quality. It is apparent that without that one cannot address the issue of development efforts without addressing the quality of the software at the same time. A poorly designed system, although it can be put together quickly to process an acceptable test correctly, may cost more in the long run because of additional costs of maintenance. Hence, improving the quality is a major goal of research in software engineering. Since 1970s, there are a number of quality models of software systems.

2.2.1 Hierarchical models

Hierarchical models decompose software quality attributes into a number of classes and then subclasses, then list these quality attributes in a hierarchical structure. The earliest hierarchical quality models are Boehm's model (1976) and McCall's model (1977).

2.2.1.1 Boehm's model

In an important paper by Boehm (1976), an attempt was made to define software quality in terms of some high-level software characteristics. These characteristics are:

- Reliability
- Portability
- Efficiency
- Human engineering
- Testability
- Understandability
- Modifiability

In 1978, Boehm improved his work and developed his quality model, showed in Figure 2.1 (Boehm, 1978).

Boehm's model is hierarchical in nature, but the hierarchy is extended so that quality criteria are subdivided. The first division is made according to the uses made of the system. These are classified as 'general' or 'as is' utility, where the 'as is' utilities are a subtype of the general utilities. There are two levels of actual quality criteria; the intermediate level is further broken into primitive characteristics that are amenable to measurement.

MISSING PAGES REMOVED ON INSTRUCTION FROM THE UNIVERSITY

- *Portability* is the effort required to transfer a program from one environment to another.
- *Reusability* is the ease of reusing software in a different context.
- *Interoperability* is the effort required to couple the system to another system.

Boehm's and McCall's models form the basis for much quality work. Their models are very influential in the study of software quality that ISO has revised and adapted as an international standard (ISO 9126). However, these models were developed in the era of batch processing and mainframe computers. As organisations are moving towards networked systems, new quality attributes should be added.

2.2.1.3 Watts' model

Watts' model (Watts, 1987) was based upon McCall's quality model. Watts' model also used the same divisions – product operation, product revision and product transition – proposed by McCall, but it redefined some of the quality criteria, for example, the following are the definitions for integrity and interoperability.

- *Integrity*: there are two senses in which integrity is required; one is that the data or program needs to be protected against unauthorised access; the other is that the environment, i.e. other program and data, needs to be protected against the program.
- *Interoperability*: in McCall's model this characteristic was particularly related to the interaction of programs in computer networks. In any installation, however, there can be problems in data commonality and communication between systems.

Watts' model provided a more recent assessment of software metrics, which are associated with criteria outlined in McCall's model.

2.2.1.4 ISO 9126 quality model

In 1991, ISO issued an international software standard, called "Software production evaluation: quality characteristics and guidelines for their use." (ISO, 1991). In the standard, quality is decomposed into six factors:

- Functionality
- Reliability
- Efficiency
- Usability
- Maintainability
- Portability

The standard claims that these six quality factors are comprehensive; any component of software quality can be described in terms of one or more of the six factors. It also claims that each factor can be refined through multiple levels of sub-characteristics. An annex of ISO 9126 contains examples of possible definitions of sub-characteristics at the first level. However, this annex is not an official part of the International Standard. Attributes at the second level of refinement are left undefined. Hence, ISO 9126 does not provide a conceptual framework for software quality measurement.

2.2.2 Relational models

The major shortcoming of hierarchical models is that they can only provide positive relationships between the quality attributes. Relational models identify and justify the relationships, which are positive, negative or neutral. Thus as far as the relationships are concerned, relational models provide insights into the understanding of quality. The first well-known relational model was Perry's relational model (Perry, 1987). This direction of work was further developed by researchers, such as Gillies (1992).

2.2.2.1 Perry's model

Perry (1987)'s model contains three types of relationships between the quality attributes in McCall's model. The *direct* relationship between two quality attributes means that a software system that is good at one attribute implies that it should also be good at the other attributes. The *inverse* relationship means that a software system that is good at one attribute will not be good at the other attribute. The *neutral* relationship means that the qualities on two attributes are normally independent of each other.

MISSING PAGES REMOVED ON INSTRUCTION FROM THE UNIVERSITY

Perry's model although it is very helpful, suffers from two shortcomings. The first is that his analysis is based on 'common sense' and lack of hard evidence. In fact, the relationships between quality attributes can be much more complicated and often application-dependent. The second is the assumption that the relationships are commutative.

2.2.2.2 Gillies's quality model

In a study of software quality in six big organisations, Gillies (1992) developed six hierarchical quality models, in terms of the criteria used by both users and developers of software.

Gillies gave the equation of "Quality = Correctness." Correctness was seen as an umbrella property encompassing other attributes. The correctness factor was decomposed into two sub-factors: the *technical* factor and the *business* factor. A schematic model is shown in Figure 2.4.

Gillies also gave a relational model, illustrated in Figure 2.5. As many as 16 pairs of quality attributes appeared in his model. His studies demonstrated that the relationships were often not commutative, which means that although attribute A may reinforce attribute B, attribute B may not reinforce attribute A. Gillies claimed that his relational model was not project dependent.

MISSING PAGES REMOVED ON INSTRUCTION FROM THE UNIVERSITY

2.2.3 Discussion

Hierarchical quality models are successful in the identification of quality attributes and to classify them into groups so that the relationships between quality attributes become clear and easy to understand. Such models only state the positive relationships between quality attributes. They fail to identify other types of relationships between the quality attributes. This brings difficulties for organisations to improve software quality.

Relational models further investigate the different types of relationships between the quality attributes. They provide an insight and make the quality models more practical and usable. However, the relationships are simplified to be binary and stereotypes. Moreover, both hierarchical models and relational models assume that the quality attributes are of equal importance.

The software quality models identify and justify the following aspects:

- ✓ Identification of quality attributes, which decomposes abstract quality into specific elements
- ✓ Identification of the relationships between the quality attributes, which developes the structure of the models
- ✓ Process of model construction, of which validation issues are taken into consideration.
- ✓ Usage of models: what kind of software; which stage in the development processes; what activity can be helped using the model; etc.

Such models have some drawbacks. First, the models were developed by quality experts with rich experience in the field. The quality of the quality model is thus dependent mainly on the developers' expertise and understanding. Secondly, the models lack rigorous validation. As the quality models are generic ones, it is quite difficult to have sufficient empirical evidence to support the validation results. Moreover, in different situations, a quality attribute may have different weights that affect the overall quality of a product. The generic quality models cannot address such issues.

2.3 Information systems quality models

Information systems (IS) play an important role in organisations. Using IS means a large investment. Development, operation and maintenance need large financial and manpower resources. To maximise the benefits of IS within organisations, quality is of prime importance. Unfortunately, in comparison with researches on software quality, fewer efforts have been made in the research on the quality of information systems. One of the most influential works on IS quality is the SOLE model (Eriksson and Torn, 1991).

2.3.1 IS quality vs. software quality

First of all, it is worthwhile clarifying what makes the quality of information systems different from software quality. The critical point is the domain of the problems being addressed. In other words, the nature of the problem dictated what kinds of methods, techniques, tools and quality criteria are appropriate because the respective purposes of the systems are different.

The nature of the maintenance problem is different, too. Modifications arise through the users of the system as their understanding of what they require becomes better. Through their experience of using the systems, users decide what their real information needs are. The organisation in which the information system is operating is also changing from time to time, for example, because of business expanding, development of new markets, merges with other companies, etc. Another problem is that as old users are replaced by others, users' preferences may change because they view the problems differently from their predecessors.

Eriksson and Torn (1991) argued that the quality of information systems has much wider contents than software quality. Information systems' quality should concern both the business management and information management. They even suspected the feasibility of measuring the quality of information systems for the following reasons:

- There is no universally accepted quality model of information systems.
- Measuring as such is regarded as difficult or impossible.

In contrast, Dahlberg and Jarinen (1997) believe that many quality concepts, models and metrics from manufacturing industry can be used in measuring the quality of information systems. There have been some attempts to apply the ISO standards for software development guide to information systems level. However, it seems that none of them has reached a solution that can be generally accepted.

Andersson and von Hellens (1997) pointed out that IS quality represents a broader perspective than software quality, involving not only the software and hardware technologies, but also IS personnel who use the IS in their practices. Von Hellens (1997) utilised three viewpoints on quality – *organisations*, *managerial* and *engineering* – to gain a better understanding of what IS quality is and the differences between the IS quality and software quality. He pointed out that software normally refers to programs whereas an information system is the organisational context in which software is used. Thus software quality should be confined to the technical characteristics. Meantime, software carries the aspects of quality that affect its performance and affect the users' opinions about its quality. User's perceptions are beyond the technical characteristics. IS development and use should be considered as an important part of business processes within an organisation. Moreover, he concluded that different perspectives do emphasise different activities. The different methods available for controlling and managing IS quality are restricted to those commonly used in different fields. More flexible methods are needed to facilitate successful IS quality improvement.

2.3.2 SOLE model

After carefully studying the different concepts of quality, Eriksson and Torn (1991) developed a hierarchical model of information systems. This SOLE model has aroused many arguments and many improvement efforts.

The SOLE model stated that quality is not an objective characteristic, but is a subjective one. Hence, in order to find all the important quality factors and attributes, all concerned groups of stakeholders and the purposes of their concerns should be listed. The model classified three different concerned groups of stakeholders: the *management*, *users* and *IS personnel*. Accordingly, it defined three main quality factors: *IS cost effectiveness* (C), *IS use quality* (U) and *IS work quality* (W). Hence, it reached the equation: Q (IS quality) = C + U + W.

Top management is responsible for and has the prime interest in the factor C. The quality factor C means the low cost and high benefits.

C = IS cost effectiveness

- C_1 = Low cost of IS
  - C_{11} = acquisition cost
  - C_{12} = updating cost
  - C_{13} = operating cost
- C_2 = high benefit from using IS
  - C_{21} = direct measurable
  - C_{22} = indirect measurable
  - C_{23} = not measurable

The cost effectiveness concept has been largely neglected in the software quality discussion. From the top management's points of view, low cost efficiency means low IS quality.

IS use quality concerns the use of the systems; it can be broken into the quality of users' requirements and the quality of the interface.

U = IS use quality

- U_1 = requirements quality
- U_2 = interface quality

The concept of requirements quality means that IS is designed to meet the users' requirements. It means that augmentability is an essential quality attribute, i.e. prepared for anticipated changes. Another essential attribute is security.

The concept of interface quality means the ease of use and ease of learning to use. IS users are responsible and interested in the IS use quality. As different users have different quality views and standards, these users should be identified in specific organisation for specific systems.

The concept of IS work quality deals with the responsibilities of IS personnel. A crude classification gives three different sub-groups: IS managers, IS designers and IS operators. Based on this division, the factor W can be further broken into the following criteria.

W = IS work quality

- W_1 = efficient IS management
  - W_{11} = production quality
    - cost-effective estimation
    - quality standards and procedures
    - project management
    - life-cycle step procedures
  - W_{12} = product quality
    - cost effectiveness
    - reusability
    - compatibility
    - portability
- W_2 = evolution quality
  - W_{21} = design quality
    - accurate, consistent and correct
    - reliable, robust, fault-tolerant
    - hierarchical, modular
    - structured

MISSING PAGES REMOVED ON INSTRUCTION FROM THE UNIVERSITY

during the software life-cycle. As all hierarchical models, three main drawbacks can be found in this model.

- It neglected to describe the relationships between the quality criteria.
- It failed to give the weights of quality factors and criteria, or the relative importance of different quality factors.
- It was based on the waterfall development process model. With the rapid development in computer sciences, information systems lifecycle may be different from the waterfall process model. Other candidates include spiral model RAD model, etc

Recently, Andersson and von Hellens (1997) further developed the concept of IS work quality in the SOLE model. Their IS work quality model provides a framework that allows the considerations of different work contexts and the specific needs of an organisation to evaluate the IS quality and the business benefits achieved through using IS. They believe that IS work quality represents a broader perspective than software quality because IS quality should involve people using the systems and their practices.

Andersson and Eriksson (1996) improved the SOLE model by combining the quality model with a method, called *input-process-output (IPO) analysis*, for adapting the general model to specific needs of an organisation and to break the quality factors into quality criteria and metrics levels.[1] Figure 2.7 shows the IPO method.

[1] They renamed the Cost Efficiency to Business quality.

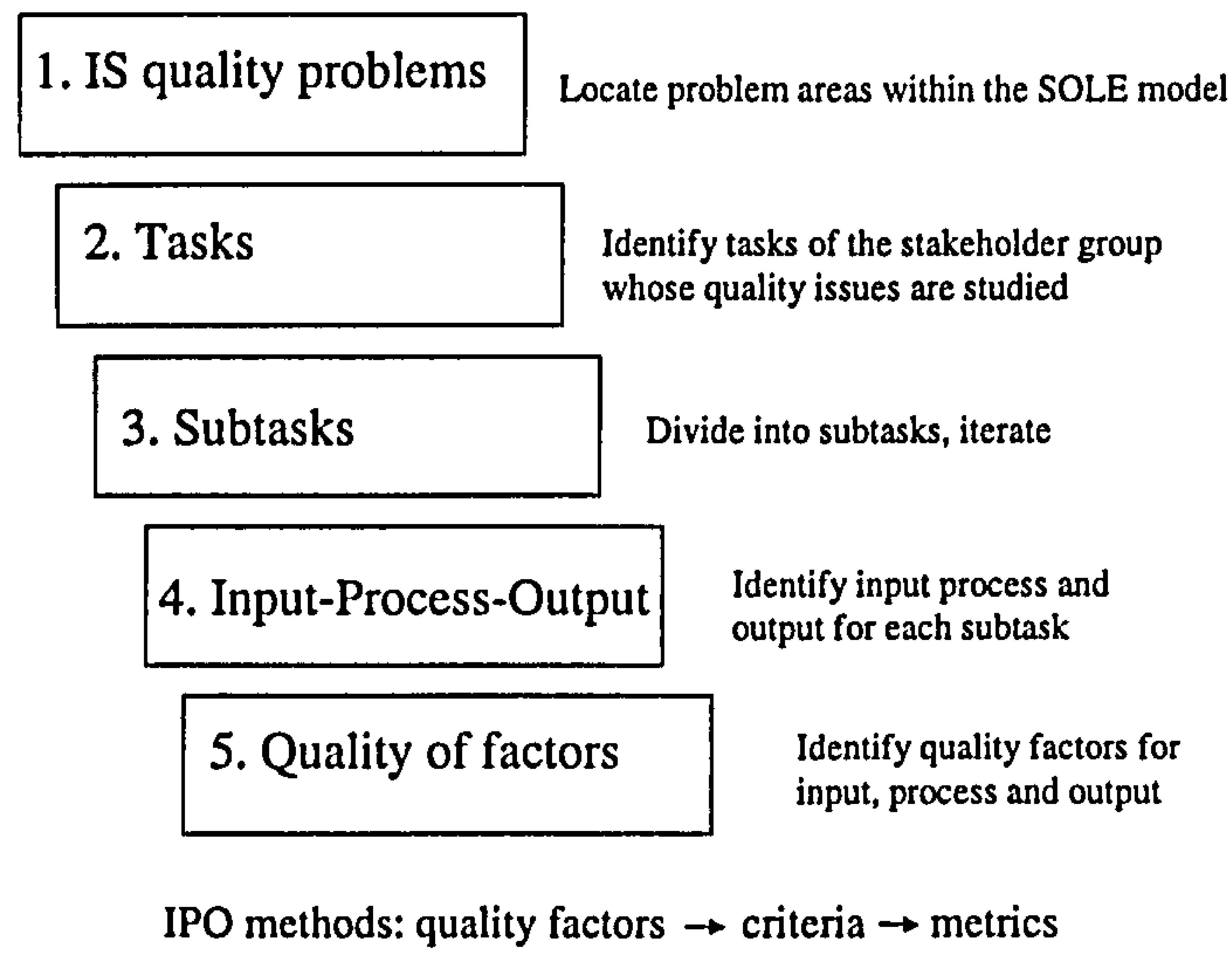

Figure 2.7 IPO Method

With regard to the SOLE model, Salmela (1997) argues that business quality has a wider perspective than IS quality. Business quality is defined as the net value of an information system for the user's organisation. It is affected by two factors: (a) the cost of planning, design, implementing, maintaining and operating the system, and (b) the benefits gained through using the system. He developed three models to describe this issue: model 1 represents the business quality model, illustrated in Figure 2.8; model 2 identifies the costs that are affected by IS quality in Figure 2.9; and model 3 represents the benefits that are affected by the IS quality in Figure 2.10.

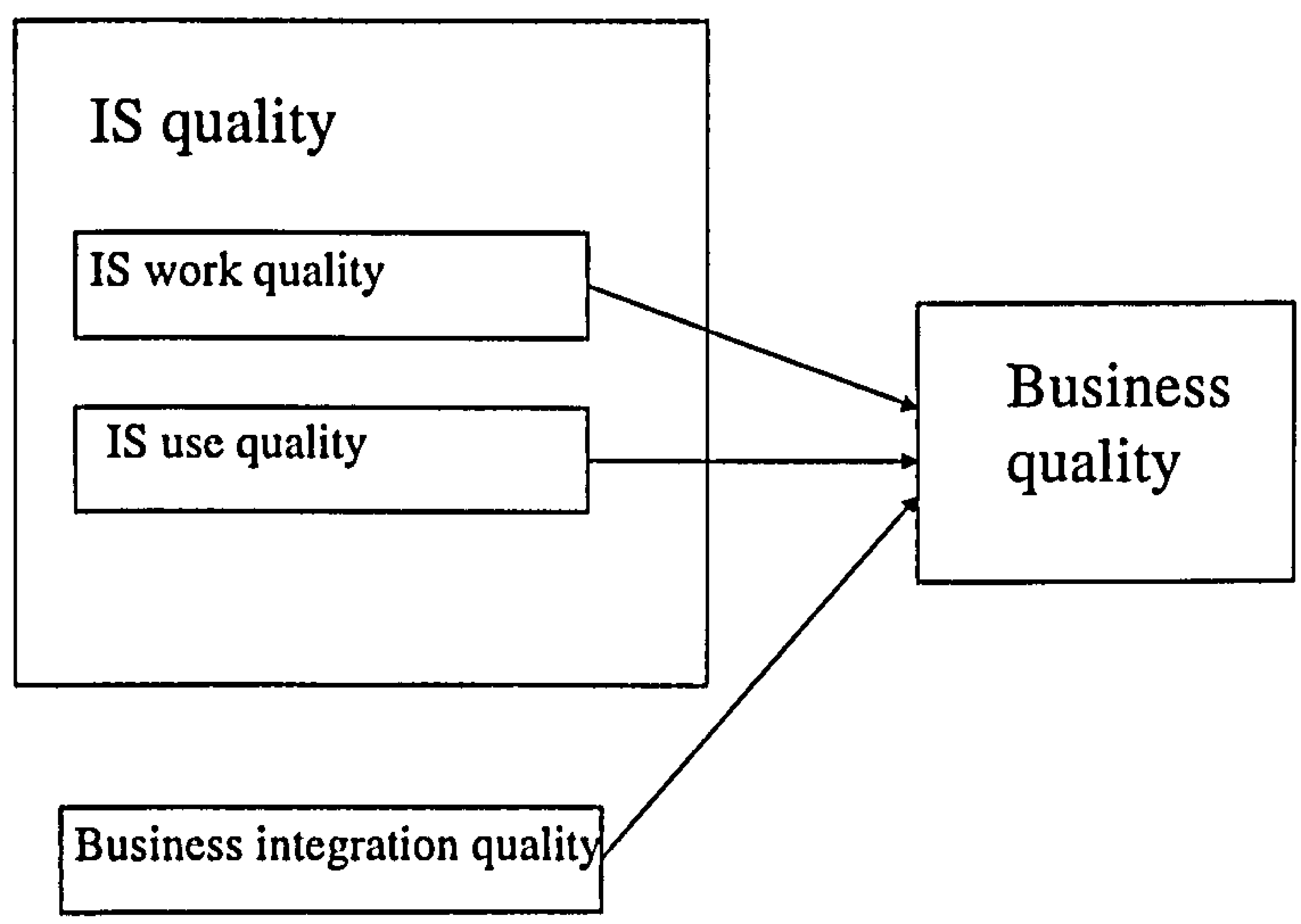

Figure 2.8 Determinants of business quality

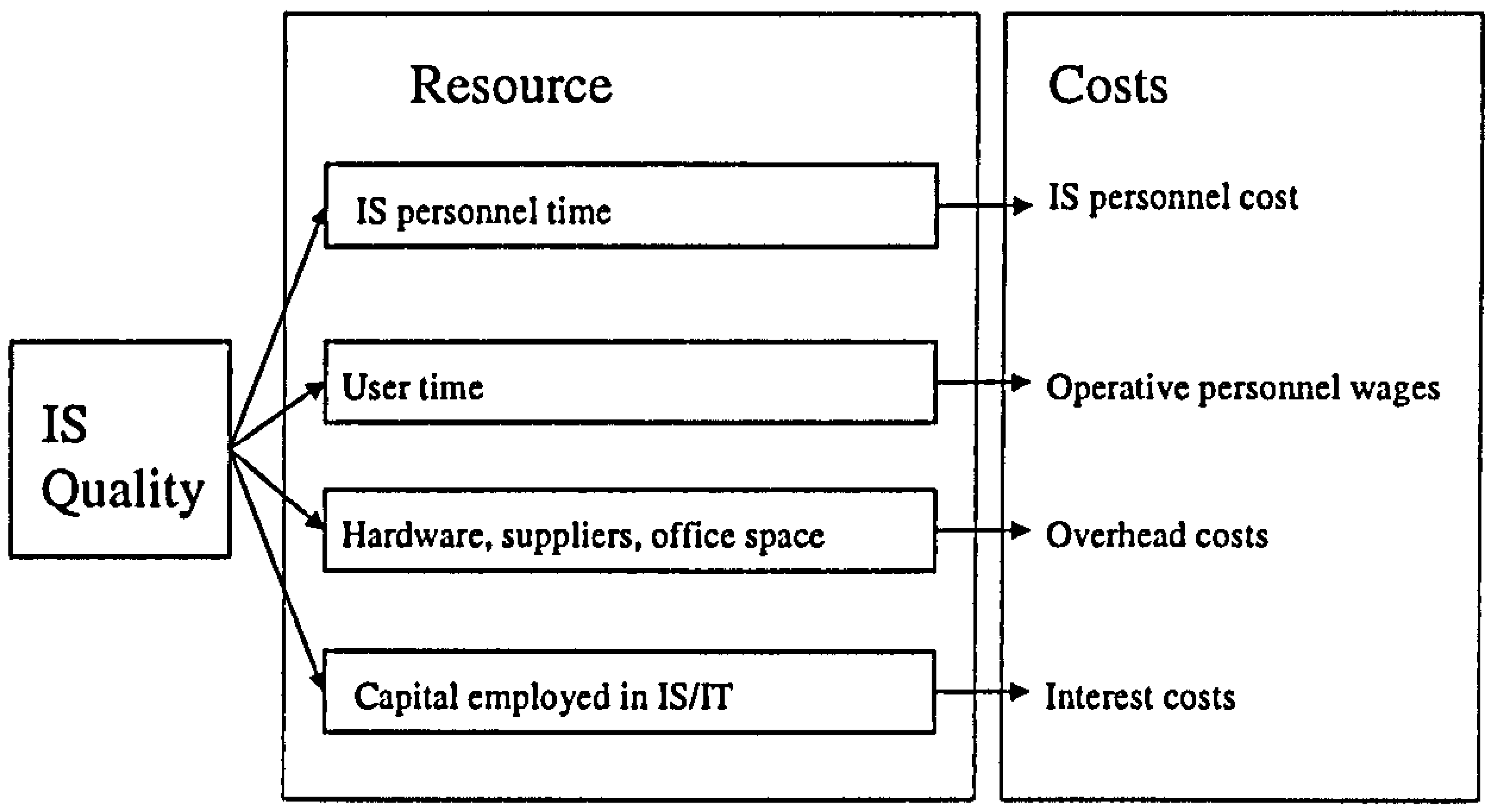

Figure 2.9 IS quality vs. Costs

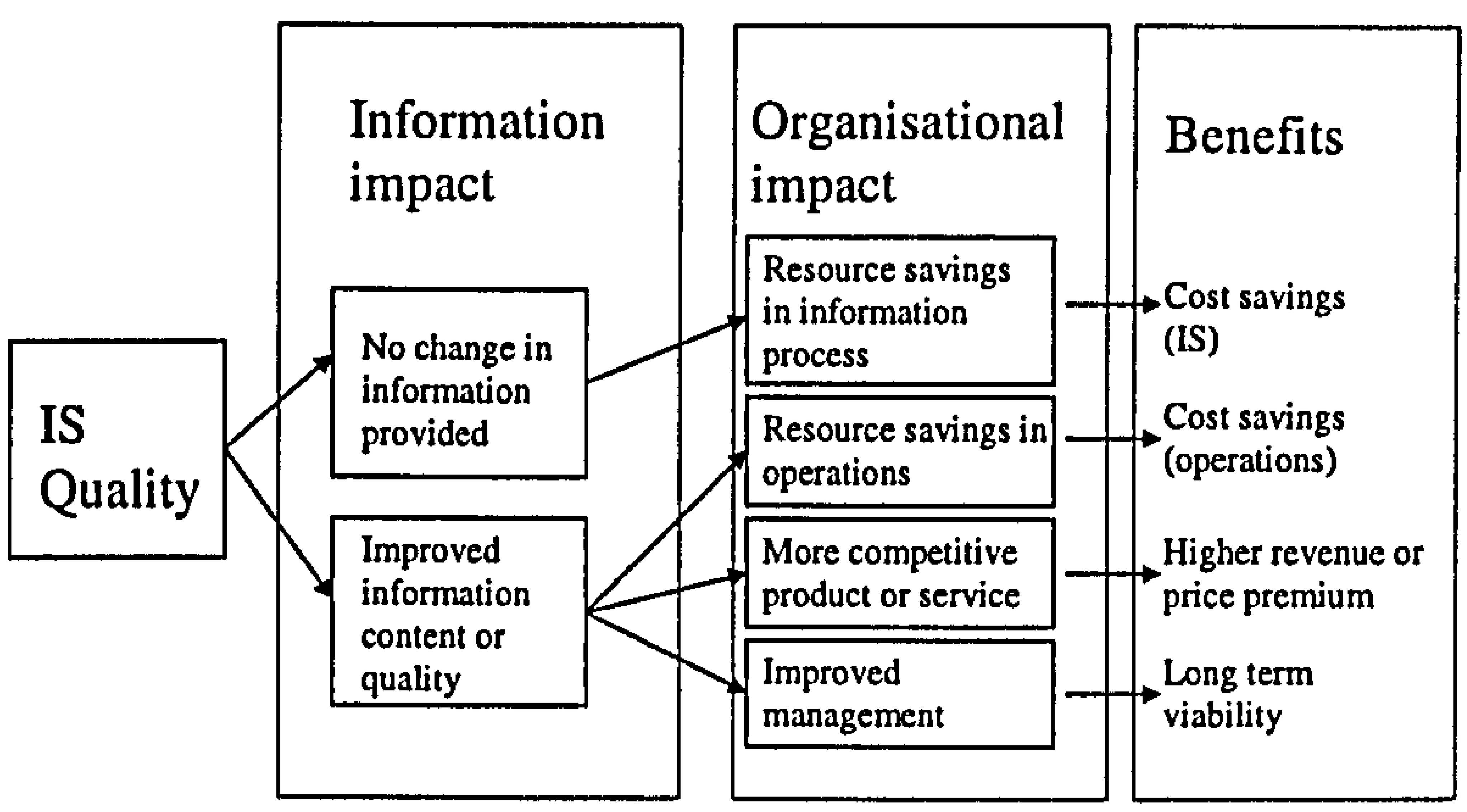

Figure 2.10 IS quality vs. Benefits

In his study, Salmela paid great attention on the approach from IS quality to business quality. He adopts a different perspective on the values of IS quality. It is not enough to increase the IS quality alone. Much attention should be paid on the processes for integrating IS with business. In other words, efforts to improve IS quality should be supported to improve the quality of the processes which aim at integrating IS with business.

Swanson (1997) paid his attention on maintaining IS quality. Swanson reviewed the four definitions of quality by different parties (management, users and IS personnel): quality is excellence; quality is value; quality is conformation to specification; quality is meeting and/or exceeding user expectation (Reeves and Bednar, 1994). He analysed the strengths and weaknesses of these definitions. Then he analysed the IS quality maintenance according the four quality definitions. He concludes that:

1. Maintaining value is concerned most by IS management.
2. Maintaining conformance to specifications is concerned most by IS staff at a technical view.
3. Maintaining user requirement is concerned by IS users in terms of their interests and their being served.
4. Maintaining excellence has a wider perspective and all the parties concern it.

Auer (1998) discussed the information use quality in detail. He argued that researches had been concentrated mainly on the planning, designing and implementing stages, but post-implementation stage is ignored. Further, post-implementation studies have been concentrated on user satisfaction rather than the ability required to use IS in an organisational context. Auer's research put emphasis on assessing levels of skills in using IS in an organisational context. He summarised the evaluation of IS use quality in terms of IS usage, IS skills, IS knowledge, IS views and organisational skills and knowledge. His conclusion suggests that IS abilities might be at an alarmingly low level though IS usage seems to be active (Auer, 1998).

2.3.3 Delone and McLean's model

Delone and McLean (1992) decomposed the IS quality into two components: information quality and system quality. They also decomposed both the information quality and system quality into quality attributes. The model they developed is shown in Figure 2.11.

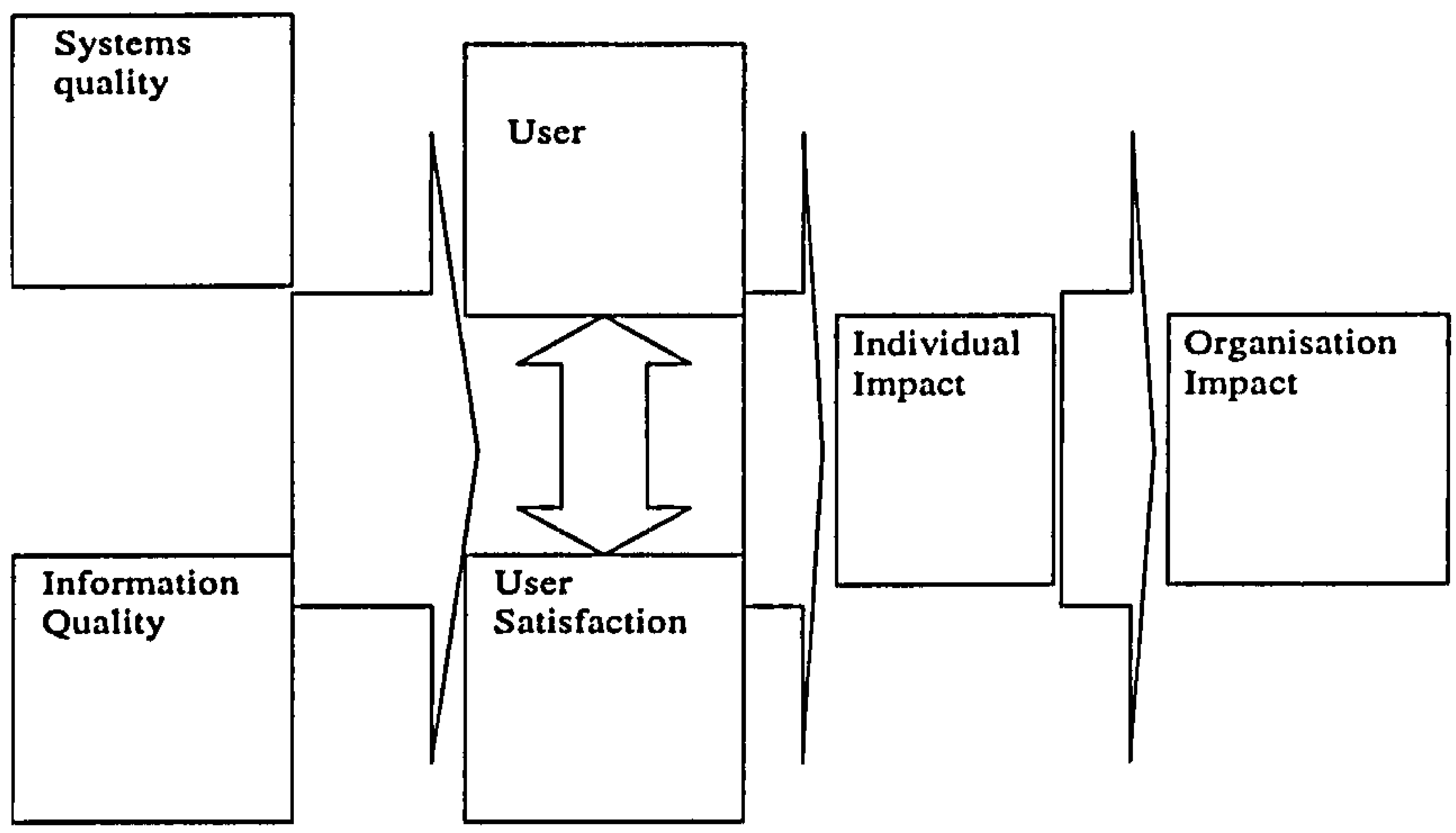

Figure 2.11 Delone and McLean's IS quality model

The information quality focuses on the usable output of the system, and the system quality focuses on the means to the information ends. Both information and system quality affect the users' satisfaction with individual and organisational impacts. (Delone and McLean, 1992).

2.4 WIS quality models

The growth of the Internet and WWW has made an important impact on business, commerce, industry, education, government and service sectors. Web-based information systems impose a great challenge to the current theories and methods of software quality management (Zhu, et al, 2000). Till now, most of the information systems have been developed by handcraft or authoring tools, but not according to an engineering approach. This might cause the 'web-based information systems crisis' like the 'software crisis' forty years ago (Hall, 1999). The main reason is the lack of management methodologies and techniques. There is not a common accepted product model nor process model for web-based information systems, and even fewer efforts have been made on quality model and measuring techniques. Although web-based information systems have been mentioned for nearly ten years, still little is understood about the quality factors affecting product, process and project management.

In accordance with the growth of the Internet and World Wide Web, there has been some research on the quality issues of web-based software systems. In (Lindroos, 1997), the differences between the web-based information systems and the conventional information systems are discussed from the perspective of software quality, and it is concluded that a quality model for web-based information systems is needed. In (Olsina, et al, 1999), a quality model for websites of universities, called Website QEM, was proposed via breaking down the quality of websites into more than a hundred attributes. The Website QEM model focuses on the user's view and therefore cannot satisfy the needs of managers and developers.

2.4.1 Lindroos' work

Lindroos (1997) discussed the quality issue concerning the WWW applications. WWW can be studied both as a media and as an information system. It differs from the traditional IS. The service provider and customer both need a connection to the Internet, but not at the same place. The connection to Internet can be provided by a third party. Hence, Lindroos identified three kinds of stakeholders associated with WWW: WWW user, IS provider and Internet connector.

As the user is free to choose where to surf, it becomes essential to attract the user to stay longer and come back later. It should also consider the issues such as whether the websites are easy to find and quick to access the information.

Many old technical issues like code correctness become a relatively smaller problem than information content, presentation and organisation (Lindroos, 1997). Due to the rapid development of electronic commerce, the security issue becomes more and more important. Consistency is still an important issue in the design of information systems, the designers will have to have a clear understanding of relative standards.

Another issue that should be considered is the so-called screen-reading. Reading from a screen is about 25% slower than reading from a paper (Nielsen, 1996). The standard of the WWW interface and browsers will make retrieval of information and use of service easier for the end-users.

2.4.2 QEM model

In (Olsina, *et al,* 1999), a model, in Figure 2.12, for evaluating the quality of websites of different universities is proposed. The model breaks down hierarchically the quality into more than one hundred quality factors. Most of the factors are specific to the users of websites of universities, such as current and prospective students, academic personnel, and research sponsors.

1. Usability
1.1 Global Site Understandability
1.1.1 Global Organization Scheme
1.1.2 Quality of Labeling System
1.1.3 Student-oriented Guided Tour
1.1.4 Image Map (Campus/Buildings)
1.2 On-line Feedback and Help Features
1.2.1 Quality of Help Features
1.2.2 Web-site Last Update Indicator
1.2.3 Addresses Directory
1.2.4 FAQ Feature
1.2.5 On-line Feedback
1.3 Interface and Aesthetic Features
1.3.1 Cohesiveness by Grouping Main Control Objects
1.3.2 Presentation Permanence and Stability of Main Controls
1.3.3 Style Issues
1.3.4 Aesthetic Preference
1.4 Miscellaneous Features
1.4.1 Foreign Language Support
1.4.2 What s New Feature
1.4.3 Screen Resolution Indicator

2. Functionality
2.1 Searching and Retrieving Issues
2.1.1 Web-site Search Mechanisms
2.1.2 Retrieve Mechanisms
2.2 Navigation and Browsing Issues
2.2.1 Navigability
2.2.2 Navigational Control Objects Contextual (sub-site) Controls
2.2.3 Navigational Prediction
2.3 Student-oriented Domain-related Features
2.3.1 Content Relevancy
2.3.2 On-line Services
3. Site Reliability
3.1 Nondeficiency
3.1.1 Link Errors
3.1.2 Miscellaneous Errors or Drawbacks different browsers
4. Efficiency
4.1 Performance
4.1.1 *Static Page Size*
4.2 Accessibility
4.2.1 Information Accessibility
4.2.2 Window Accessibility

Figure 2.12 QEM model

In their work, Olsina, *et al* outline more than a hundred and twenty quality characters and attributes for the domain of academic sites. The primary goal is to classify and group the elements that might be part of a quantitative evaluation, comparison and ranking process in a requirement tree. After analysing three different audiences regarding academic visitor profiles, the work shows a hierarchical structure for characters, sub-characters and attributes regarding the students' viewpoints.

Olsina *et al* state that different kinds of users should be considered in order to effectively select quality characteristics. They choose three different audiences regarding the visitor standpoint in the academic domain. The visitors are categorised in current and perspective student, researchers and professionals and research sponsors.

Their work outlines the quality attributes for academic site domain regarding the students' view. The work manages to pose quality issues on websites in the earlier research stage. However, some drawbacks exist. One is that the work lacks of the discussion from others' view, managers view and technician's view, as the multi-view methodology has been accepted as an international standard to define and analyse the quality issues. Another drawback is that this model is for specific domain; the quality attributes therefore are domain-oriented. More work should be done to analyse the quality issues for other domains. Also the method of decomposition is arguable.

2.4.3 Miller's model

Miller (2000) defined website quality in the following considerations:

- Time: does website change often and rapidly? How much has the site changed since last upgrade?
- Structural: are all links working? Are there parts of the site that are unconnected?
- Content: does the content match its purpose? Do key phrases exist continually in highly changeable pages?
- Accuracy and consistency: is the data presented accurate enough? Are today's copies of downloaded the same as yesterday's?
- Response time and latency: does the server respond within certain parameters?
- Performance: how does the performance vary by time of day? By load and usage?

Miller also suggested that quality is in the mind of user. This is consistent with the usability engineering approach, which will be discussed in 2.4.5.

2.4.4 Brajnik's model

Brajnik (2001) stated that a quality model of a website is more difficult to develop compared with software applications due to:

- Web involves many different subsystems, languages, databases, and make the definition of quality attributes a challenge task.
- Web has greater information density than other applications. As a consequence, it is very difficult to capture properties like navigability and satisfaction.
- The developers have no control of devices users use.

He suggested that a quality model of website should be a set of criteria that are used to determine if a website reaches a certain level of quality. He categorised quality into three divisions:

1. task-related factors, such as presentation quality, content and function adequacy and navigability
2. performance-related factors, such as response time, reliability and robustness.
3. development-related factors, such as code complexity, code readability, portability, and modifiability.

Brajnik's quality can be regarded as a web-version of McCall's model. He followed McCall's quality modelling method and divided the quality into three divisions.

Brajnik emphasises that the major usage of a quality model is to determine usability problems and to determine a baseline for comparison. His first two divisions of quality model are based on users' points of view, and the third division is for developers.

2.4.5 Web design heuristic evaluation

Heuristic evaluation is based on the idea of testing conformance of interface to a set of predefined guidelines. Nielsen and Mack (1994) described it as "the most informal method." Nielsen's ten heuristics, shown below, have been regarded as '10 Golden Rules' for Web design (Nielsen and Molich, 1990; Molich and Nielsen, 1990):

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Over the years, different Web design heuristics or guidelines have been emerged, (Navarro and Khan, 1996; Siegel, 1997; Shneiderman, 1998; Fleming, 1998; Berners-Lee, 1999; Dalgleish, 2000; Powell, 2002). There is some overlap in the heuristics with some authors citing the same heuristic and using the same terminology. Sometimes different authors will combine heuristics in different ways but they still cover the similar ground. Different authors from different concern developed the differences of the heuristics. For example, a security consideration is a major issue in e-commerce systems design, but it is not included in Nielsen's heuristics, which is regarded as design guidelines for universal websites.

2.4.6 Discussion

Several quality models of websites were developed. Some of them followed a conventional software quality modelling method; some others outlined the quality attributes from the system features of the site. Heuristic evaluation has been seen as an easy and popular way to assess quality from the user's point of view. These approaches managed to pose quality issues and usability problems on websites in the earlier development stage. However, some drawbacks exist. One is that the quality modelling methods are subjective, mainly from the model developers' craft skills. Another drawback is that the models were developed by developers from specific domains. For example, the QEM model is a quality model developed for an academic site. Thirdly, although abundant web design guidelines can be found in the literature, they were developed for universal websites, and system-specific usability issues might be ignored. Furthermore, no commonly accepted design guidelines exist.

2.5 Quality measurement

2.5.1 Measurement basics

2.5.1.1 Definition of measurement

Over the last thirty years, there has been a growing awareness of the relationships between and engineering and measurement. Quality Assurance (QA) is becoming an important technique of software engineering and information systems engineering. Central to this technique is measurement.

Measurement can be defined as the process of assigning symbols, usually numbers, to represent an attribute of the entity of interest, by rule. (Shepperd, 1995).

There are a number of points to note from this definition. First, the entity that to be measured can either be a product, like software and documentation, or a process. Second, there must exist some distinctive attribute, like cost or length, to measure. Third, assigned symbols or numbers should be used to represent the attribute. Last, but not least, the assignment of symbols and numbers should obey certain rules which provide a basis for objectivity in the measurement process.

2.5.1.2 Importance of measurement

Lord Kelvin's following statements clearly explained the topic:

> When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced on the state of science.

The aim of measurement is to answer some questions. Measurement helps overcome subjectivity, to provide a precise analysis, and to predict the future ongoing of the systems.

In software engineering, for instance, managers use measurement to assess the productivity of the software engineers. Another application concerns the assessment of system reliability over time and maintainability. Also the measurement can help the assessment of the process of project. It can monitor the process in every step to decide whether the project goes off course.

2.5.1.3 Representation theory of measurement

In any measurement activity, there are certain rules to be followed. Measurement theory tells us the rules, laying the groundwork for developing and reasoning about all kinds of measurement.

The representational theory of measurement seeks to formalise the intuition of the way the world works. The data obtained as measures should represent attributes, and manipulation of the data should preserve relationships observed among the entities. In the literature, the representation approach has been used in software measurement (Baker, et al, 1990; Melton, et al, 1990; Bieman, et al, 1992; Zuse, 1992; Fenton and Pfleeger, 1997). The basic concepts of measurement theory come from the foundations of measurement in (Krantz, et al, 1971; Finkelstein, 1984; Suppes, et al, 1989; Luce, et al, 1990).

Formally, measurement is defined as a mapping from the empirical world to the formal, relational world. Consequently, a measure is the number or symbols assigned to an entity by this mapping to characterise an attribute.

The real world is the domain of the mapping, and the mathematical world is the range. When mapping the attribute to a mathematical system, many choices can be made for the mapping: real numbers, integers, or even symbols. Besides, the behaviour of the measurement in the number system needs to be the same as the corresponding elements in the real world, so that by studying the numbers the real world can be understood. In other words, the mapping needs to preserve the relation. This rule is called the representation condition.

The representation condition asserts that a measurement mapping M must map entities into numbers and empirical relations into numerical relations in such a way that the empirical relations preserve and are preserved by the numerical relations.

The mapping is sometimes called a representation, because the measure represents the attribute in the numerical world. A relational structure is a $(n + 1)$–tuple $(S, R_1, \ldots, R_n)$, where S is a set, and $R_1, \ldots, R_n$ are relations on S. A function f: $S \rightarrow S'$ is called an order preserving mapping from relational structure $(S, R_1, \ldots, R_n)$ into relational structure $(S', R_1', \ldots, R_n')$, formally,

$$R_i(a_1, a_2, \ldots, a_{ri}) = R_i(f(a_1), f(a_2), \ldots, f(a_{ri})) \qquad \textbf{Eq 2.1}$$

where $i \in (1, 2, \ldots, n)$

The representation theorem, which must be true for any proper measure, states that the relations between the empirical observations must hold for the number system relations. This does not necessarily impose many restrictions upon the measurement function, and it is this observation that enables us to formally distinguish between scale types.

2.5.1.4 Measurement scales and scale types

Not all measurement mappings are the same. The differences among the mappings can restrict the kind of analysis that can be performed. The measurement scales and scale types are introduced to understand the differences. Measurement is referred to mapping, *M*, together with the empirical and numerical relation systems, as a measurable scale.

In general, there are many different representations for a given empirical relation system. The most commonly used measurement scale types are:

- Nominal scale: When defining classes or categories, and placing each entity in a particular class or category. This categorisation is the basis for the most primitive form of measurement, or nominal scale.
- Ordinal scale: the ordinal scale is used when the empirical relation system consists of classes or categories that are ordered with respect to the attribute. The ordering leads to analysis not possible with the nominal scale.
- Interval scale: the interval scale preserves the order. Moreover, the scale captures the information about the size of the intervals that separate the classes or categories, so that one can understand the size of the jump from one class to another. An interval scale can be expressed as an affine transformation: $M = aM' + b$, where a and b are scalars.
- Ratio scale: the ratio scale preserves order and interval scale, moreover, there is a zero element to express the non-existence of a certain attribute. The measurement mapping must start at zero and increase at equal intervals. A ratio scale can be expressed as a ratio transformation: $M = aM'$, where a is a positive scalar.
- Absolute scale: the absolute scale is the most restrictive of all. For any two measures, M and M', there is only one admissible transformation: $M = M'$

One interesting topic concerning a measurement scales is the meaningfulness of the measurement. A measurement is meaningful if its truth-value is invariant of transformations of allowable scales. The five measurement scales carry more and more information from nominal scale to absolute scale. The most useful scale of measurement is ratio scale (Fenton and Pfleeger, 1997).

Another topic concerns the applications of the statistical methods. To make the results meaningful, it is of great importance to apply the appropriate statistical methods. Siegel and Castellan (1988) summarised the statistical methods relevant to each type of scale, in Table2.1.

Table 2.1 Measurement scale and statistical methods

Scale type	**Defining relations**	**Suitable arithmetic**	**Suitable statistic**
Nominal	Equivalence	None	Mode Frequency
Ordinal	Equivalence / Greater than	None	Median Percentile Spearman Kendall
Interval	Equivalence / Greater than / Known ratio of any intervals	Addition Subtraction	Mean Standard deviation Pearson product-moment correlation
Ratio	Equivalence / Greater than / Known ratio of any intervals / Known ratio of any two scale values	All arithmetic analysis	Geometric mean Coefficient of variation

2.5.2 Reliability, validity and bias

Watts (1987) suggested seven criteria to assess software metrics. The most important three are:

- Reliability: the results should be precise and repeatable. Good measurement must have a high degree of reliability. This means that repeated measurements should obtain consistent results. In obtaining a reliability coefficient, the common techniques are split halves, test-retest, alternative forms, and parallel forms (Pedhazur and Schmelkin, 1991).

- Validity: the metric should measure the correct characteristic. Valid data must support the inferences made from the measurement. The validity of the inference is determined by the strength of the evidence.
- Objectivity: the results should be free from subjective influences. Many different types of bias may be introduced into the research. In questionnaire-based techniques, careless sampling may introduce selection biass. Others may include leading questions, response bias gender/ethics bias. It is important to prevent and eliminate such bias in research design.

Further important features include standardisation, comparability, economy and usefulness: the measure must address a need. Gillies (1997) pointed out that automation is also an important feature which should be involved.

2.6 Software quality metrics

2.6.1 Software metric basics

Quality measurement is usually expressed in terms of metrics. A software metric is a measurable property which is an indicator of one or more of the quality criteria. A metric is a proposed measure, and only when the following conditions are met, can a measure be called a metric (Gillies, 1997):

- A metric must be clearly linked to the quality criteria that it seeks to measure.
- A metric must be sensitive to the different degrees of the criteria.
- A metric must provide an objective determination of the criteria that can be mapped onto a suitable scale.

It should be noted that metrics are not the same as measures. The two terms, although having some overlap, are distinguished in Section 1.1.2.3.

Software metrics include observation, evaluation and prediction. Observation is to show the specified indicators when software is being used. For example, by observing if a piece of latest news appeared in a media-based website, one can estimate the timeliness of such a site. Evaluation is to emulate the software in the targeted environment, and to find if the objectives are met. When predicting, any part of the system or function may not exist. A predictive metric can be used to predict the software later in the lifecycle, such as structureness can be used to predict the maintenance of a software system.

Software metrics can be implemented by using tools, testing and software emulation. The software measurement process consists of formulation of measurement targets, finding the measurable objects, finding suitable measuring method and tools, and evaluation of the measurement results (Wallmuller, 1994).

2.6.2 Software complexity metrics

2.6.2.1 Halstead's software science

Halstead (1977) distinguished software science from computer science. The premise of software science is that any programming task consists of selecting and ranging a finite number of operators and operands. Halstead developed a system of equations expressing the total vocabulary, the overall program length, the actual volume, program level, program difficulty, and other features, such as development efforts and projected number of faults in the software. The primitive measures of Hallstead's software science are:

n_1: the number of distinct operators that appear in a program

n_2: the number of distinct operands that appear in a program

N_1: the total number of operators in a program

N_2: the total number of operators in a program

Halstead's major equations include the following:

Vocabulary $n = n_1 + n_2$ Eq 2.2

Length $N = N_1 + N_2$ Eq 2.3

Volume	$V = N\log_2(n)$	**Eq 2.4**
Level	$L = V^* / V$	**Eq 2.5**
Difficulty	$D = V / V^*$	**Eq 2.6**
Efforts	$E = {}^{V}\!/_{L}$	**Eq 2.7**
Faults	$B = {}^{V}\!/_{S^*}$	**Eq 2.8**

where V^* is the minimum volume represented by a built-in function performing the task of the whole program; S^* is the mean number of mental decisions between errors. (S^* is 3000 according to Halstead).

Halstead's work has had a great impact on software measurement. However, software science has been controversial since its introduction and has been criticised in terms of methodology, derivation of equations, and other factors. Empirical studies provided support to some equations.

2.6.2.2 Cyclomatic metric

McCabe (1976) designed the cyclomatic metric to indicate a program's testability and maintainability. It comes from graph theory's cyclomatic number, which indicates the number of regions in a graph. When applying it to a program, it is the number of linearly independent paths comprising the program. To determine the paths, the program procedure is represented as a strongly connected graph with a unique entry and an exit node. The general formula for cyclomatic number is:

$$V(G) = e - n + 2p \qquad \textbf{Eq 2.9}$$

where *V(G)* is the cyclomatic number; *e* number of edges; *n* number of nodes; and *p* number of unconnected nodes of the graph.

To have good testability and maintainability, McCabe suggested that no program module should exceed a cyclomatic number of 10. Many practitioners in software testing recommend use of the cyclomatic representation to ensure adequate test coverage, and McCabe's complexity measure has been commonly accepted by practitioners (Kan, 1995).

2.6.2.3 Structure metrics

Halstead's software science and McCabe's cyclomatic complexity metrics of module complexity assume that each module is a separate entity. Structure metrics consider the interrelationships between modules. The most commonly used design structure metrics developed by Myers were Fanin and Fanout (Myers, 1978). Fanin is the number of modules that call a given module; fanout is the number of modules that are called by a given module. Myers pointed out that modules with a large Fanin are relatively small and simple; and a large Fanout may indicate poor design.

Henry and Kafura (1981) developed a structure metric as follows:

$$C_p = (Fanin \times Fanout)^2 \qquad \textbf{Eq 2.10}$$

where C_p is the structure complexity.

Henry and Selig (1990) revised the above equation and defined their information flow metric:

$$HC_p = C_{ip}(Fanin \times Fanout)^2 \qquad \textbf{Eq 2.11}$$

where HC_p is the information flow complexity; C_{ip} can be measured by any module complexity such as cyclomatic number.

Based on various approaches, Card and Glass (1990) developed their system complexity model:

$$C_t = S_t + D_t \qquad \textbf{Eq 2.12}$$

$$C = C_t / n \qquad \textbf{Eq 2.13}$$

where C_t is system complexity; S_t structure complexity; D_t data complexity;

C relative system complexity; n number of modules.

They discussed that Fan-in is not an important factor in structure complexity, and structure complexity is further defined as:

$$S_t = \frac{\sum f^2(i)}{n} \qquad \textbf{Eq 2.14}$$

where S_t is structure complexity; $f(i)$ Fan-out of module i; n number of modules.

The data complexity is defined as:

$$D_t = \frac{\sum \frac{V(i)}{f(i)+1}}{n} \qquad \textbf{Eq 2.15}$$

where D_t is data complexity, $V(i)$ I/O variables of module i; $f(i)$ Fanout of module i.

Card and Glass tested their model using eight software projects. They found that the system complexity measures significantly correlated with subjective quality assessment by a senior development manager and with the development error rate.

2.6.2.4 Discussion

A review of the software metrics literature led Curtis (1979) to quip 'there are more complexity metrics than practicing computer scientists.' Although there was a flurry of activity in the field, there was also much criticism about the area of software metrics and much criticism of the methodologies used (Ott, 1995).

The main drawbacks are:

1. Theoretical basis. Halstead and many other researchers tried to integrate computer science and cognitive science (Curtis, 1981). In general computer scientists will do well to purge from their memories the magic number 7 ± 2

and the Stroud number of 18 mental discriminations per second. These numbers describe cognitive processes related to the perception or retention of simple stimuli, rather than complex information processing tasks.

2. Lack of validation. As was pointed out in (Curtis, 1979), while many metrics were developed, research in the field often lacks the direction provided by sufficient definitions of the constructed studies. In particular, there were too many metrics and too little explanatory theory or empirical evaluation.

3. Invalid validation. In (Bieman, et al, 1995), they summarised two common wrong approaches in which validation of software metrics has been performed. One was to validate software metrics against historically collected data. Metrics developers would claim that their metrics were validated because there was a strong correlation with the data. As such data might be unreliable, or collected in specific circumstances, the approach was rather unreliable. Another approach was to validate software metrics against existing metrics. A very simple example of a software metric was the much quoted lines of code (LOC). There is a well-known attribute of the code which is captured by LOC, namely, length. However, LOC has not been shown convincingly to be a valid measure of complexity. Hence, it is not a sound approach to validate complexity metrics against LOC.

Other criticisms concerned the statistical techniques used (Henry, et al, 1981; Hamer and Frewin, 1982).

2.6.3 Framework of software metrics

It can be seen that, from the above discussion, a rigorous approach of metrics development is of great importance. Over the last twenty years, on the basis of criticisms of existing software metrics, methodologies of software metrics development emerged. They are:

1. Axiomatic approach. From 1981, researchers were identifying attributes which they claimed a given class of metrics should satisfy. For example, Kafura and Henry (1981) presented criteria that they argued that all software metrics

should satisfy. In (Prather, 1984) and (Fenton and Whitty, 1986) different axioms were presented, and the authors argued that all complexity metrics should satisfy. Another well-known axiomatic system on software complexity was developed by Weyuker (1988). Weyuker claimed that syntactic complexity metrics should satisfy certain criteria and argued that the axiomatic system could be used to develop new metrics and to validate existing ones.

2. Measurement theory based. It was not until the late 1980s that significant discussion of the use of measurement theory as the basis for research in the software metrics field continued. Much of this work can be found in (Baker et al, 1987; Melton, et al, 1990; Zuse, 1991; Fenton, 1994; Fenton and Pfleeger, 1997).

Other developments can be seen in the applications of new experimental and statistical techniques. 'Measurement and experimentation are complimentary processes' (Curtis, 1981). Such methodology put the focus on empirical results and experimentation. Sound software metrics should be validated by reproducible experiments. MacDonell (1991) also discussed the topics which covered experiment design, subject selection, choice of subjects, appropriateness of subjective assessment, and choice of statistical techniques.

2.7 Website metrics

There is much research on website design usability. Many design guidelines have been developed (Comber, 1995; Nielsen, 1999; Lynch and Horton, 1999). However, designers have suffered from following guidelines (Borges, et al, 1996; Smith, 1986). One mainstream of Web usability research over the last few years has been on the more rigorous approach – metric-based usability specification and evaluation (Botafogo, et al, 1992; Smith, 1996; Ivory, et al, 2001).

2.7.1 Website Navigation

Navigability design is one of the trickiest areas of website development. It is tricky because it is so subjective – everyone seems to have a different opinion of what works. Fleming (1998) showed that usability focused navigability testing is by no means a simple issue. This is because usability itself is a vastly complex concept, while navigability is only an attribute of usability. In addition, the Web has to cater for different types of users each with an individual style of preference. Web navigation is a challenge because of the need to manage billions of information objects (Nielsen, 1999), which makes the measuring of navigability extremely difficult.

The issue of breadth versus depth of website structure in website design has been widely studied. Results from several studies have suggested that a website with many links, a means to reduce the depth, is the optimal condition for user performance (Kim, et al, 2001; Furnas, 1997). Zaphris and Mtei (1997) also found that in a site of 64 links, the design with 8 links per page and two levels resulted in fastest response time and lowest navigation efforts. In web-design circles, a widely quoted metric, or heuristic, in navigation design is the so-called "three-click rule" (Wise, 2000), which states that the user should be able to get from the homepage to any other page on the site within three clicks of the mouse.

McGovern (2001) believes that the strength of navigation is how quickly users can find what they are looking for. A good navigability design includes a variety of navigation attributes. A questionnaire of 14 questions was produced, questions such as "How easily can you identify where you are within the website?", "how similar are the navigational elements to the other websites?", "how correct are your expectations from links?" etc. Website navigability was tested with a rating from one to five for the above questions. McGovern's study was usability focused. McGovern emphasized the importance to have a baseline statistic against which to compare the results obtained from an analysis.

Another approach to measuring Web navigability is to analyse the server log files. In (Buchanan and Lukaszewski, 1997), real data of web usages, such as Visitors-per-Page, Pages-per-Visitor, Average-Duration-of-visitor-session, are gathered from server log files and analyzed to derive the user's feelings and perception of websites. Sullivan (1997) investigated four aspects when measuring navigability using server log files:

- how do people arrive at the site (leaf page or main page?)
- how do they hit various portions of the site? (identifying frequently used navigational devices)
- are there any portions of the site unexplored? (this would suggest a need to improve visibility)
- users' reaction to page load time

The result of Sullivan's study was that most users used Topical Tour to navigate the site. In addition, a surprising result was that once the users encountered the Tour pages, on average they would visit another 14 additional pages. With the result Sullivan made several changes to his site, mainly with the Topical Tour more accessible. Consequently, the number of high-traffic visitors increased by 14%.

Rodriguez, et al (2001) stated that applying classic usability testing to navigability has proven to be slow, expensive and inaccurate. Navigability testing in their view "requires high precision and permanent observation of the users." They developed a tool called the Automatic Navigability Testing System (ANTS). It was designed to study user behaviour without having to observe the user and testing out which navigational bars/types were the most effective. ANTS was able to record the exact position of the user on a navigational map as well as record the duration of how long a user spends on each page. Although the tool could be useful for future information retrieval concerning navigability, it is still at a starting point, as the results showed little distinctive relationship between user behaviour and navigability (Rodrigues, 2002).

Jin, Zhu and Hall (1997) proposed an abstract model of hypertext application systems as a directed graph (see e.g. Figure 2.13), which is equally applicable to websites. In this model, a website can be modeled as a pair $< G, S >$, where $G = (V, E)$ is a directed graph representing the website; V is the set of nodes representing web pages;

E is the set of edges representing links between web pages; and *S* is the start node of the graph, i.e. the home page of the website. The directed graph must also satisfy the condition that all nodes *v* in *V* are reachable, i.e. there is at least one path from the home page to node v. They suggested the use of the Number of Independent Paths (NOIP) as a measure of hypertext navigational complexity. The larger the NOIP, the more complex the website structure is, the easier for a user to get lost in the network, and the poorer navigability.

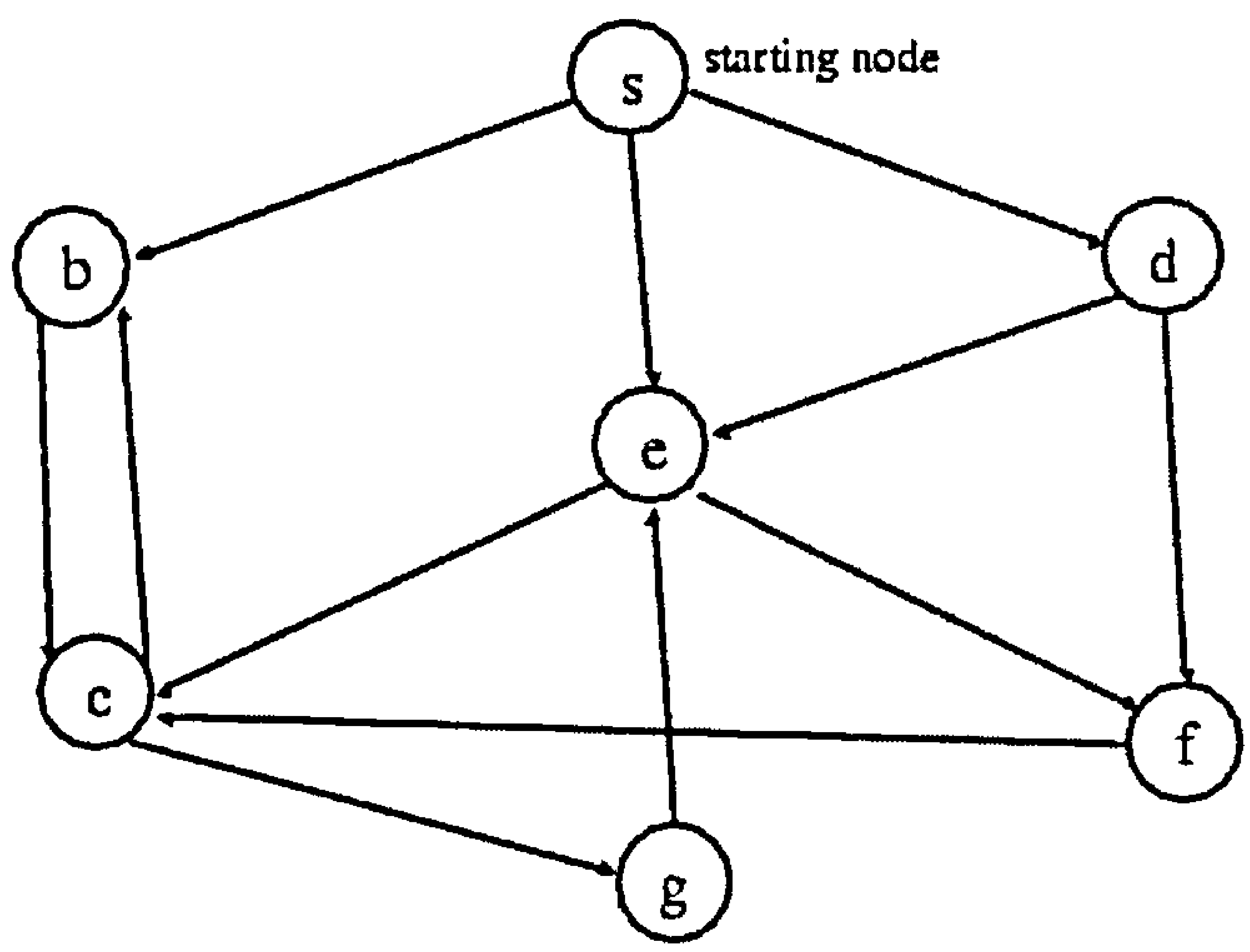

Figure 2.13 An abstract graph model of a website

Other metrics using the Web graph property are investigated in a recent survey by Dhyani et al (2002). As the Web attributes are measured by considering the Web site as a directed graph, the Web graph reflects the Web structural organization, hence determines the ease of navigation (ibid). Some metrics, such as Centrality, Global metrics and Local metrics are presented in (Botafogo et al, 1992; Kumar, et al , 1999; Broder, et al, 2000).

2.7.2 Discussion

Existing Web metrics made the earliest attempts to make measurements about the website. It can be believed that with the development of metric-based usability specification and evaluation, a rigorous engineering approach will be formed. However, some drawbacks exist. Firstly, most existing website metrics lack in effective validation, similar to the drawback discussed in 2.6.3. Some metrics were evaluated using preliminary empirical studies. Secondly, some quality metrics of WIS cannot be found in the literature. For example, timeliness is an important quality attribute for media-based websites. The measurement of timeliness still remains an open problem.

2.8 Summary

2.8.1 Quality models

Existing quality models can be classified into the following three categories:

1. *Software quality models.* These are the quality models where the only concern is software, which include all models covered in section 2 such as the famous McCall's and Boehm's models. It is believed that WIS have much wider contents than software alone. Therefore, although some methods and thoughts developed in the research on software quality are still useful for WIS, only considering the software aspects is not enough.
2. *Information system quality model.* These quality models view quality from various stakeholders' viewpoints, such as the SOLE model, which defines quality from managerial, technical and user-based views, hence divides the quality into cost-effectiveness, work quality and use quality. This work is based on library information systems, many new features of WIS, such as distributed and multimedia features, are not mentioned or ignored.
3. *Separated models.* These models divide information systems quality into information quality and systems quality, such as Delone and McLean's quality model. It is an arguable approach. Information quality and systems quality have big overlapping parts, and cannot be considered separately.

Existing quality models have served software development and information system development fairly well so far. The main quality factors are still correct in today's study. However, this study is concerned with the quality of WIS, in which new features were not fully covered. Furthermore, existing quality models are for general software systems. Specific systems should have different quality concerns.

2.8.2 Quality metrics

Software complexity metrics play an important role in software quality assurance. Since their introduction, positive and negative comments have been merging. Metric derivation and validation lack a rigorous approach. Empirical results of most metrics and models often depend on the product, the development team and development environment (Kan, 1999).

There exist some well-known software and Web metrics. It is problematic to use them on WIS if they are not justified and validated. Most existing Web metrics were developed to assess the general behaviour of the website, not for specified quality attributes.

2.8.3 Open questions to answer

By reviewing the existing research on quality models and metrics, a number of difficult issues must be addressed, such as:

- WIS are distributed, multimedia and intelligent. How to define the quality of such systems?
- WIS can be grouped into different categories, including e-business systems, personal homepage, corporation memory, service provider, etc. Each category has quite different quality requirements from others, and it has the different quality attributes, or different weight of a quality attribute respectively. Is it possible to develop a generic quality model for all kinds of WIS?
- WIS navigability is an important quality attribute for university websites, and timeliness for media-based systems. How to measure such quality attributes?

To address the above questions, the following approaches will be applied:

- Instead of developing a generic WIS quality model, developing a quality modelling method would be a solution.
- Based on existing research, new quality metrics will be developed to measure the quality attributes. A rigorous approach of quality metrics development will be conducted. The metrics should be validated through formal or empirical studies.

Chapter 3 Quality Modelling of Web-based Information Systems

3.1 Introduction

Software quality models are models of software systems for understanding, analysing, evaluating, measuring or predicting quality. They play a significant role in software development. A quality model can provide general guidelines for the elicitation of users' requirements on software quality. It can provide the insight for software designers to seek technical solutions to achieve the required quality. It is the foundation for organising quality assurance activities. For example, it can provide the information about a system's sensitivity to a quality issue so that adequate testing must be targeted directly to the issue. It is the foundation for the development and deployment of software quality evaluation methods and quantitative measurement metrics.

Chapter 2 surveyed the most well known software quality model over the past thirty years, including hierarchical models such as McCall's model (McCall, *et al*, 1997), Boehm's model (Boehm, *et al*, 1978) and relational models such as Perry's model (Perry, 1991) and Gillies's model (Gillies, 1992; 1997). Regarding information systems as a much wider concept than software systems, the SOLE model (Eriksson and Torn, 1991) and its variants provide three different views for different interest groups of stakeholder: users, technical staff and managers. SOLE model's view of quality is consistent with the general theory of quality and quality management proposed by Garvin (1984).

Since WIS are distributed, multimedia, co-operative, and even intelligent, they have imposed great challenges to current theories and methods of software quality management (Zhu, *et al*, 2000). The differences between the WIS and conventional information systems from the perspective of software quality (Lindroos, 1997) require that new quality models for WIS are necessary. Chapter 2 reviewed the recently developed quality models of WIS, such as the QEM model (Olsina, *et al*, 1999), Miller's model (Miller, 2000), and Brajnik's model (Brajnik, 2001). In the literature, a great number of guidelines for website design have been published (Navarro and Khan, 1996; Siegel, 1997; Shneiderman, 1998; Fleming, 1998; Berners-Lee, 1999; Dalgleish, 2000; Powell, 2002). However, such design guidelines and principles are for universal websites, and lack system specific considerations. No commonly accepted standards have been set for defining and assessing websites' quality.

Chapter 2 concludes that existing quality models are intended to be comprehensive and applicable to all software systems. However, as pointed out in (Kitchenham and Pfleeger, 1995) there can be no single and simple measure of software quality acceptable to everybody. It has also recognised that every software system may have its own quality concerns (Kitchenham and Walker, 1989). Special requirements of the software or the information system must be considered in the application of quality models (Kitchenham and Pickard, 1987). In particular, with the ever-growing range of information systems that are developed with different functions using various software and hardware techniques, software engineers are seeking for quality models that can provide useful insight not only for quality management, but also for other development activities. Therefore, it is impractical to find a universal quality model for all kinds of web-based applications and systems. A modelling method will enable the practitioners and researchers to develop their own quality model for the specified systems. Based on these ideas, a quality modelling method is proposed. Using this method, quality models of e-commerce systems were developed (Zhang et al, 2001), (Zhu et al, 2002), (Zhang, et al, 2002).

The rest of chapter is organised as follows. Section 3.2 will discuss existing quality modelling methods. In Section 3.3, a new quality modelling method, which consists of architectural analysis and hazard analysis, is proposed. In Section 3.4, the above method is applied to develop a quality model for e-business systems. The results are also discussed. Section 3.5 will introduce a set of notations for quality modelling. Section 3.6 summarises the chapter.

3.2 Related research on quality modelling

In the survey of the research on software and information systems quality models over the last thirty years, the challenges were recognised to develop a quality model for Web-based information systems. The quality model should be objective, suitable to the new types of Web-based applications, and applicable in the early stage of the system development life cycle.

There exist several quality modelling methods. Dromey (1995) pointed out that the difficulty of quality modelling is due to the perceived scale of the problem, the diversity of quality defects in software, and the difficulty of factoring high-level quality attributes down to tangible properties. A systematic method is necessary to construct testable, assessable, and refineable quality models for different software products and key products of software development. Dromey proposed a generic quality model and a process to systematically develop software quality models for different software products. The generic quality model consists of three principal elements: product properties that influence quality, a set of high-level quality attributes, and a means of linking them. The generic model is instantiated and refined for a particular software product through model construction, which consists of the following five steps.

- Identify a set of high-level quality attributes for the product.
- Identify the product components.
- Identify and classify the most significant, tangible, quality-carrying properties for each component.
- Propose the set of axioms for linking product properties to quality attributes.
- Evaluate the model, identify its weakness, and either refine it or scrap it and start again.

Dromey (1996) applied the methods to various types of software artefacts, such as requirement specification, high-level design and program code. Recently, Bansiya and Davis (2002) applied the method to build a hierarchical quality model of object-oriented design. Although the links between the lower-level and higher-level attributes were established and evaluated based on statistical data, the identification of quality attributes at various levels is still based on empirical knowledge.

Dromey's process seems straightforward, but still relies heavily upon the empirical knowledge of software systems. Although Dromey correctly recognised that quality models must be built through quality-carrying properties of the components of the software product, the applications of the method to requirements definition, design and implementation only produced universal quality models for such software artefacts. The specific features of the application area and system design and implementation were not considered in the quality models.

The card sort method proposed in (Upchurch, et al, 2001) provided a partial solution to the problems. It allows users to find out about people's mental models; the way they think about things. In (Upchurch, *et al*, 2001), it was applied to elicit the quality attributes of web-based applications. The method starts with the selection of a number of pictures of user interfaces, such as the downloaded screens of web pages, as the entity to sort. These pictures were shown to different groups of people, such as farmers, web designers and laypeople. Respondents were required to describe the quality attributes using subjective or objective criteria. The responses from the subjects were collected, and then, processed. There are several ways to process the result. One way is to sort these criteria and count the number of categories within each criterion. The results from the card sort method are promising as the technique is easy to use, and the technique is suitable for the investigation of the visual nature of WIS. A shortcoming of the approach is that it can only be applied after the completion of the development of the website.

It can be concluded that the above quality modelling methods cannot fully address the challenges in developing quality models of WIS.

3.3 Proposed method

In this study, a new quality modelling method is proposed. The method consists of system architectural analysis and hazard analysis.

The method of quality modelling takes architectural models of WIS as input and applies hazard analysis methods to derive the quality sensitive observable phenomena of the behaviour of the components or the system and the causal relationships between the phenomena. The quality carrying property and quality attribute that a phenomenon demonstrates is then identified according to the nature of the phenomenon. These elements are then assembled together and represented in the diagrammatic notation.

3.3.1 System architectural modelling

The quality modelling approach starts by analysing the targeted system architecture. A website can be modelled in terms of how data are collected, processed, stored and delivered to the users. For example, a job recruiting system on a site may be depicted by a number of steps with input and output data. First, there is a job description that is input by human resources and results in output of posted job listings. In response to job listings, job seekers would submit their resumes as input to the system that is then output as applicant data to human resources. This idea can also be viewed as 'workflow' where the activities through a site is defined in terms of the work that the users accomplish, as opposed to data transactions (van Hee, 1994).

System architectural analysis analyses a system by decomposing it according to information flow. The technique is chosen as it has three major advantages. Firstly, quality issues are usually found in different system usage procedures. For example, in e-commerce systems, information relevance is a major issue when users search for a certain product, and security becomes the most important issue in online payment procedures. System architecture analysis enables the system to be decomposed in an objective way and makes it possible to associate quality issues to the parts of the system, and thus makes the quality model more applicable. Secondly, it will distinguish one system from another by its architecture and application domain. It enables the quality model to be tailored for the system. Thirdly, this makes it possible to evaluate the system's quality in the system design stage of the life cycle.

The method relies on the existence of the architectural design of the system. It can be applied at the architectural design stage and any stage after that. This enables quality analysis of a system to start at an early stage of software development. It also makes it possible to build a general quality model for a number of particular information systems of similar functions and of the same architectural structure.

In the method, the architectural modelling of WIS contains two steps. The first step is to identify the information processing activities and the processes. The second step is to decompose the whole system into components or several sub-systems where the information flow between the system and the users as well as between the components is analysed.

Information flow analysis examines the activities of a system from the viewpoint of the information: where the information originates, how it is used and processed, and where it goes, including all the movements within the system. UML sequence diagrams are used to describe and illustrate the movement of information through a system. Such a diagram presents a view of information systems in simple notations that are understandable by the users and business persons, who are part of the development process (Kitchenham and Pickard, 1987). However, the method does not depend on the specific notation that represents the information flow within a system and between the users and specific components or subsystems. Ideally, the notation should support the representation of information flow through several diagrams of different abstraction levels, some providing overviews of major processes and others going into details to show each step of information movements. This enables us to analyse an information system at various levels of abstraction and consider quality models to various levels of detail.

3.3.2 Adaptation of system hazard analysis

Originally, hazard analysis aims to systematically identify, assess and control hazards before a new work process, piece of equipment, or other activity is initiated. Hazard analysis techniques have been widely used in the development and deployment of safety critical systems that involve computer software or not. In such a context, a hazard is a situation in which there is actual or potential danger to people or to the environment. Associated with each hazard is a risk, which is related to the likelihood of the event occurring and its consequences. Once the hazards are identified and analysed, safety as well as other quality requirements can be specified for each component. Risks can be avoided or reduced ultimately through technical design, management and organisational means. Consequently, the quality and reliability of the system are improved (Kletz, 1995; Leveson, 1995; Neumann, 1995; Storey, 1996).

In this context, the word hazard has its widest meaning, which means any situation that may cause harm. The more likely a hazard is to occur and the more serious the consequences of the hazard, the more important the corresponding quality attribute, and vice versa.

There are a number of hazard analysis techniques available in the literature. The FMEA (failure modes and effects analysis) technique is regarded as a strong candidate as it especially suits the needs. FMEA (Storey, 1996) progressively selects the individual components or functions within a system and investigates their possible modes of failure. It then considers possible causes for each failure mode and assesses their likely consequences. In the original FMEA, the effects of the failure are determined for the unit itself and for the complete system. Possible remedial actions are also suggested. A simple example of FMEA is given in Table 3.1.

Table 3.1. FMEA for a microswitch

No	Unit	Failure mode	Possible cause	Local effects	System effects
1	Tool guard switch	Open-circuit contacts	Faulty component Excessive current Extreme temperature	Failure to detect tool guard in place	Prevents use of machine - system fails safe
2		Short-circuit contact	Faulty component Excessive current	System incorrectly senses guard to be closed	Allows machine to be used when guard is absent -dangerous failure
3		Excessive switch-bounce	Ageing effects Prolonged high currents	Slight delay in sensing state of guard	Negligible

FMEA enables to identify a system's potential failure modes, their possible causes and the consequences. Each cause of a failure indicates what quality attribute the system is sensitive to from the developer's point of view. The corresponding consequences of the failure indicate what quality attributes the system is sensitive to from the users' point of view. Both causes and their consequences are observable phenomena of the system. Therefore, the relationships between the quality attributes or quality-carrying properties can be established.

In (Zhang, et al, 2001) (Zhang, et al, 2002) and (Zhu, et al, 2002), hazard analysis is adapted and the concept of hazard is extended to construct quality models of information systems.

3.3.3 User involvement

Users' participation is an important part in the quality modelling method. This is achieved through interviews with the users in applying hazard analysis.

As discussed above, the construction of a quality model is a qualitative research that aims at obtaining detailed information and insight into a system from the users' point of view. The author had a loose-structured discussion with a group of interviewees broadly representing the population of stakeholders in a case study (McDonald, 2002). The major reason to choose interview techniques in quality model construction and evaluation is that it is a task that heavily depends upon the experiences and domain specific knowledge. For different kinds of systems, different user groups will be recruited. For example, to find hazards and related quality attributes for Business-to-Business e-commerce systems, interviewees should be e-commerce web developers and veteran B2B systems users, etc. Conducting interviews makes it possible to obtain insight into quality issues that represent the concerns of all types of stakeholders. Follow-on questions, such as 'why do you think...', enhance the understanding of the study. Therefore, when implemented effectively, interviews can produce substantial data.

In the case study of the quality modelling, ten interviewees were recruited. They were all veteran e-commerce systems designers and users. The number 10 was decided on the basis of Nielsen's study (Nielsen, 2003). He justified that, for testing usability attributes, 5 experienced users will locate 75% of the problems. Recruiting more users only produce replicate results.

3.3.4 Construction of quality model

In the quality modelling method, the construction of a quality model takes the information charted in the FMEA. All the quality attributes derived will be listed and grouped. Following the taxonomy in (Miller, 2000), and decomposing the quality of B2B e-commerce systems into three dimensions: content dimension, time dimension and presentation dimension, one can then put the quality attributes into related groups and form a tree-shaped quality model.

Next step, the number of times each quality attribute appears in the FMEA is counted, and the result is used to decide the importance weight of each attribute.

3.4 Case study: a quality model of e-business systems

This section applies the method proposed above to e-business systems.

B2B e-commerce systems are different from the B2C or C2C systems, as they have a number of special characteristics including large volume of goods traded, high value of goods traded, multiple electronic payment methods, and involvement of business bidding, contracts and agreements (Chan *et al*, 2001). These features make B2B systems more complex to manage than B2C or C2C systems. In this section, a quality model of B2B e-commerce systems is developed using the method described above.

3.4.1 Development of quality model

At the highest level of abstraction, a B2B e-commerce system consists of three parties, the Buy-side, the Virtual market and the Sell-side. Information is exchanged among these parties in certain orders. A cycle of activities must be completed before a deal can be made. The flow of information is depicted in the UML sequence diagram of Figure 3.1, which is revised from (Chan et al, 2001).

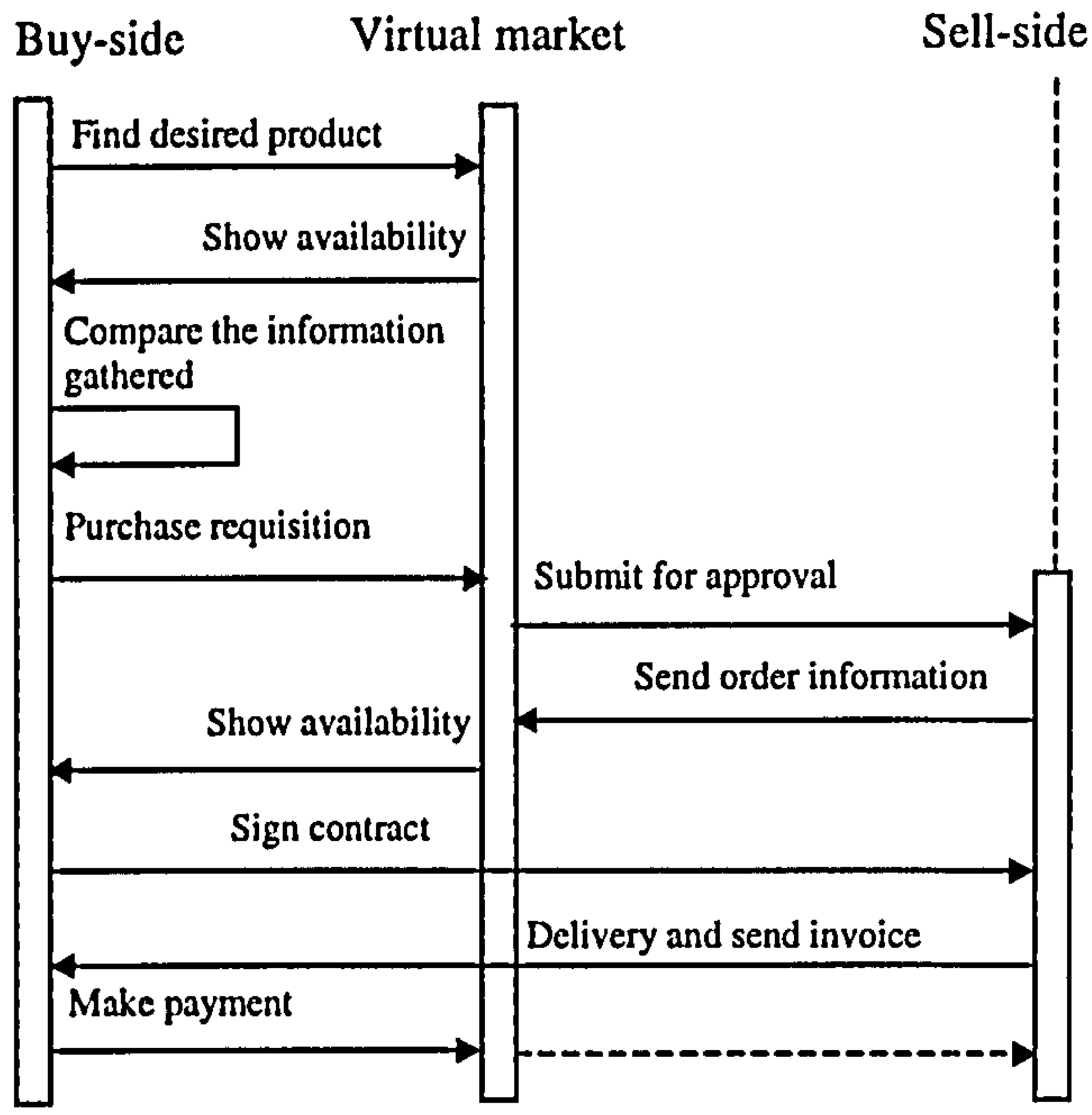

Figure 3.1 Buy-side B2B systems information flow diagram

The information exchange process within the systems can be decomposed into four phases: information searching, purchasing requisition, signing contract, and receiving goods and make payment. The contract-signing activity is usually undertaken between the buy-side and sell-side. Therefore, the basic functions of the virtual market are product-information-searching, purchase-requesting, and online payment.

Phase I: Information Searching

In this phase, a user tries to find the information for a product. The user may use search engines, directories, and other means to find as much useful information as possible. However, problems or failures are common with such methods of information search. By applying hazard analysis, one can identify three types of failures: (1) No information found; (2) The information found is irrelevant, including the situation that too much is presented but only a little is relevant; (3) Information is inaccurate. For each type of failure, there may be one or more possible causes of the failure. For example, the causes of no information found can be the following.

(a) *Broken links*: A broken link may happen for a number of reasons, such as errors in the software, the server is shut down, the web address has been changed but the database of the search engine has not been updated, etc.

(b) *Low response time*: 'Page not found' will appear if the browser fails to open a web page within a certain pre-set value of time-out. The causes of time-out can be heavy traffic on the network, excessive load on the server, large volume of information due to e.g. big graphical image or applet, etc.

(c) *Poor usability*: The end user does not know how to use the system, e.g. does not know the format of keywords, and cannot find online help, etc.

(d) *Incompatible platforms*: The platform or system the user is using does not support the web applications.

For each type of failure, the consequences of the failure can also be identified. Such consequences can be classified into two parts. One is the immediate effect of the failure and the other is the long-term and more profound effect at system level. For example, if the failure of no information found occurs, the immediate effect is unable to present the required information to the user, therefore, the following up activities cannot be performed, i.e. no business can be done. In the long-term, if such failures occur many times, it may drive the user away and cause the loss of customers because of the image of poor quality.

The consequences of a failure highlight the quality issues of an information system from the users' point of view. For example, no information found is a usability problem. The possible causes of a failure provide the insight of the technical issues that contributes to the quality attribute. It provides the relationship between the quality attributes of developers' view to the quality attributes of users' view. The consequences of the failure and the probability of the occurrences of the failure determine the importance of the quality issues involved in the failure mode.

The result of above analysis is then charted in a form shown in Table 3.2. In fact, filling the form drives the analyst to think of the possible failure modes, their causes and consequences. While the completed form itself forms a detailed quality model, hierarchical or relational quality models can be derived from the form.

Table 3.2. FMEA for request of detailed information of a certain product

Ref No.	Failure modes	Possible cause	Quality Attribute	Local effects	System level impacts
1	Cannot find information at all	*Broken links*: the website has been changed or shut down, but the search facilities have not been updated in timely manner.	Navigability – link validity Timeliness	Failure to present informa-tion to end users	Prevent further use of the web site and drive the user away.
		Long response time: 'Page not found' will also appear if the browser fails to open a web page within a certain time. Large volume of information is transmitted, e.g. large image or applet may cause such a problem.	Responsive-ness Information presentation and style		
		Difficult to use: end user does not know how to use the site, e.g. does not know the format of keywords, etc. and cannot find online help	Usability		
		The platform or system that the user is using does not support the web application	Portability		
2	Cannot find relevant information: find too many results but only a few are relevant	The search facilities have remained limited in their capabilities and consequently not adequate to deal with the volume of available resources. The website does not update in a timely manner and neglects useful information.	Completeness Relevance Timeliness	Failure to present useful informa-tion	Users may stop using such sites.
		The search engine returns an enormous number of results to users' queries, and the output of the search lacks of enough details to enable users to assess the relevance of the information.	Relevance Usability		
		Users do not know how to use the search facilities, such as the format of keyword search.	Usability		
3	Information is inaccurate.	The information stored in the system is incorrect when it was first input to the system.	Accuracy Correctness	Incorrect informa-tion is presented to the user	Users move to other web sites if they suspect the quality of this site
		The information is not updated in a timely manner. This makes the information outdated and not correct.	Timeliness		

The most important quality attributes in this phase are as follows.

- *Correctness*: the information is correct and true.
- *Relevance*: the information is what the users expect and need.
- *Navigability*: users find it easy to move between the webpages to find the required information through browsing. Without proper signposts, it is very easy to get lost and disgruntled. Broken links can also drive the customers away.
- *Interactivity:* a quality website should interact with the users. It should not only get feedback from the users, but also provide help, especially when users feel it is difficult to find information on the web.
- *Timeliness*: the information provided is timely and not out-of-date.
- *Responsiveness*: information should be retrieved and presented within an acceptable time delay after the user requested it. This attribute is not only related to the system's structural design, but also the interface design and other environmental factors such as the traffic on the Internet and workload on the server. For example, big pictures or animations usually result in a long delay time.
- *Information organisation*: the information is well organised, such as proper web structure; appropriate graphic, animation used; and easy to read.

Phase II: Purchase Requisition

In this phase, users have found some information of desired goods, and they may have compared several sites for a certain product. They may request a purchase online or by other means. Here, the author only considered the situation in which the users send purchase requisitions online. Table 3.3 lists the FEMA chart for this phase.

Table 3.3 FMEA for purchase requisition

Ref No.	Failure modes	Possible cause	Quality Attribute	Local effects	System impacts
1	Users give up in the middle of transaction	*Long response time*: this may be caused by web design style; too many Java applets or animation would dramatically slow down the response time. Some popular web sites may receive enormous visits at a certain time.	Respons-iveness	Failure to present information to end users	Prevent further use of the web site
2	Goods are out of stock	*Untimely information*: some products have been sold out, but the information was not updated in a timely manner.	Timeliness		

The most concerned quality attributes are:

- *Timeliness*: the information should be updated in a timely manner.
- *Responsiveness*: web site should respond to user within a certain time.

Phase III: Signing Contract and Goods Delivery

These activities are completed between the buy-side and sell-side, without the use of the virtual market. Although they are important parts of the whole of the B2B systems' activities, they are not part of web-based systems.

Phase IV: Online Payment

In this study, the author only considers the case that the user makes payment online. The FMEA for online payment is illustrated in Table 3.4.

Table 3.4. FMEA for online payment

Ref No.	Failure modes	Possible causes	Quality Attribute	Local effects	System effects
1	Stop the payment in the middle of the transaction	*Web security*: this is a prime factor. Users feel that they will lose control if they provide the credit card details online. Some users may stop if they are asked about private information.	Security	Users stop using such sites	Legal actions may be involved
		Long response time: if a user does not receive any response within a certain time after sent the bank details, he / she, in most cases, would stop the transaction rather than risk a second try for the likelihood of purchasing the same products twice.	Respons-iveness		
2	Accept the wrong account information	A user sends wrong account information or errors occur during the encryption/decryption processes or in the transmission of the information.	Security		
3	Information is leaked to other parties	Lack of protection to private information. Hackers invade the net.	Security		

The most important quality attributes at this phase are:

- *Security*: security is becoming an important part of the functionality of an online e-commerce system. Security can go far beyond the website and extend to server security, network security and physical site security.
- *Responsiveness*: information should appear within a certain period of time. A low response time will not only cause systems to fail, but also drive the users away.

Now, all the above quality attributes can be listed from the above tables. Some attributes, such as correctness, accuracy, completeness and security are about the information content. Some attributes, timeliness and responsiveness, can be categorised into the time dimension. Some attributes, like design style, interface, and structure, are on the presentation dimension.

If decomposing the quality of B2B systems into three dimensions: content dimension, time dimension and presentation dimension, a quality model of such systems can be constructed as illustrated in Figure 3.2.

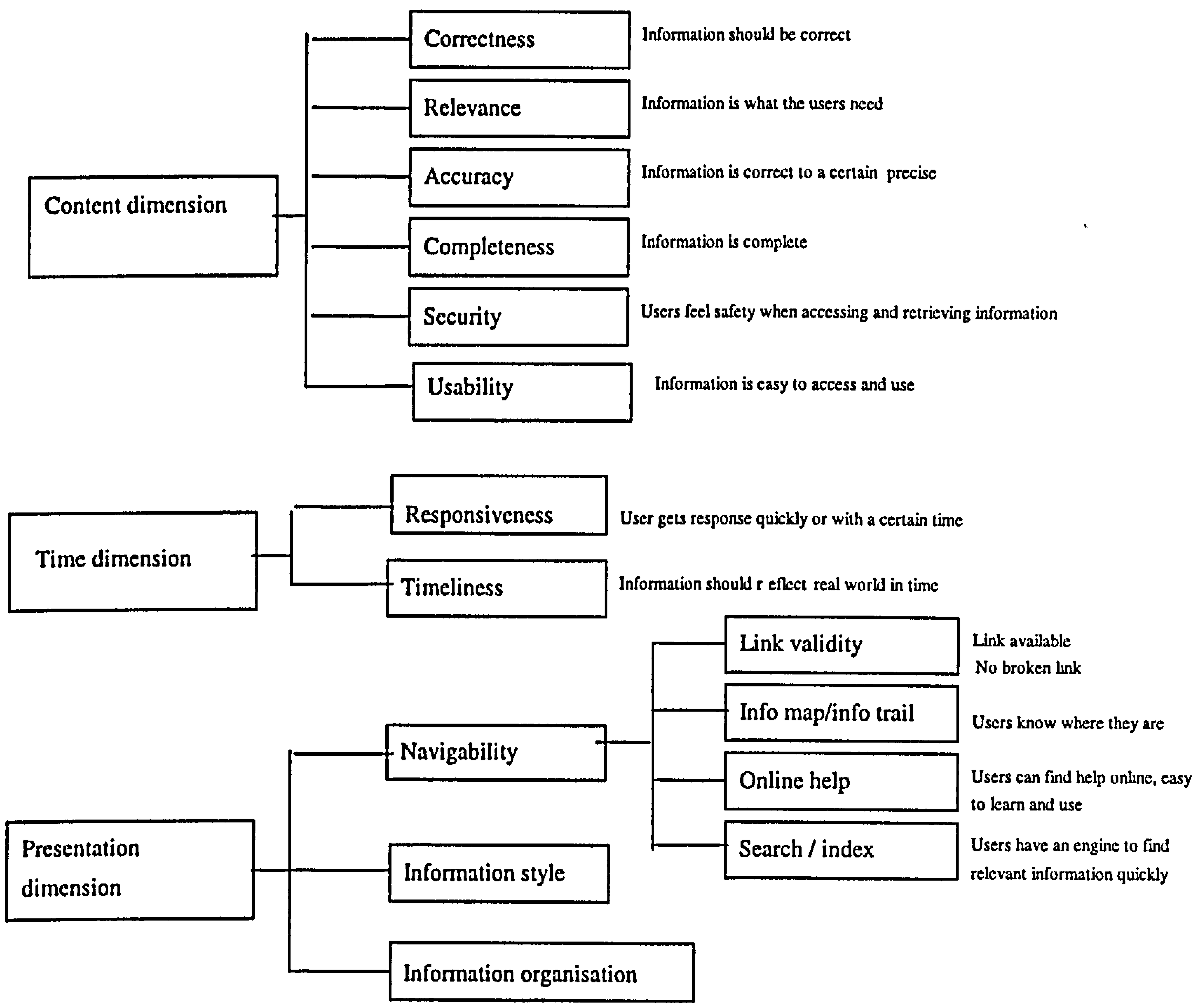

Figure 3.2. A quality model of B2B e-commerce systems

Now it is time to assign a weighting for each quality attribute. A simple way is to calculate the likelihood of a hazard occurring, by counting the times of appearance of each attribute, see Table 3.5.

Table 3.5 Weigh the importance of quality attribute

Quality attribute	Times of Appearance
Correctness	1
Relevance	2
Accuracy	1
Completeness	1
Security	3
Usability	3
Responsiveness	3
Timeliness	4
Navigability	1
Presentation style and organisation	1

From Table 3.5, it can be seen that, for B2B business e-commerce systems, security, responsiveness, timeliness and usability are the four most important quality attributes.

3.4.2 Discussion

From the case study, it can be seen that the quality modelling method is highly applicable and easy to use. It can be seen that the method can be directly used to construct quality models of targeted Web-based information systems. It provides a bridge between the abstract system quality attributes and the tangible quality-carrying properties of components and the observable behaviour of the system.

Compared with the existing Web-based quality models, such as the QEM model, the quality model developed from the proposed method has the following advantages:

1. The QEM model is still applying the traditional software quality modelling method, in which three users were interviewed and the quality attributes were found based on experience and brainstorming. The new method follows a rigorous and logical approach using appropriate system analysis techniques. When developing a quality model of e-commerce websites, veteran e-commerce systems designers and users were interviewed. The approach therefore ultimately avoids subjectivity.
2. The QEM model is a product-based quality model. Quality assessment can only be done after the completion of the website. The new approach enables Web designers to derive quality models at an earlier stage of WIS development. This is particularly important because the awareness of a

sensitive quality attribute at an early stage such as at the design stage enables WIS designers to seek for technical solutions to achieve the required quality standard.

3. The new quality modelling method derives quality models at the architectural level and so enables WIS designers to understand the quality of a type of WIS in a particular application domain and of the same architectural features. The results of such quality modelling method have a wide applicability.

3.5 Notation development

In (Zhu, *et al*, 2002), a set of notations to present quality model was developed. This presentation is developed on the basis of adapted FMEA tables, but brings much more useful information. It not only gives visual presentation, but also illustrates the links between the abstract quality attributes and the software properties.

For example, safety is an important quality attribute of safety critical systems. It is of extreme importance for engineers to understand how faults and failures of the components are related to the safety of the system. Only when such information is available, can design solutions be put forward to eliminate the specific types of faults of the components and to prevent the occurrences of the specific types of failure modes that may contribute to safety. Moreover, testing of the system can then directly target the safety-related components and events to ensure system safety.

A quality attributes/quality-carrying property of a component, such as usability and maintainability, is usually abstract. Consequently, the links between two abstract properties cannot be easily established or validated. However, abstract properties usually demonstrate themselves through various concrete events and observable phenomena, which are tangible and observable. For example, the poor usability of a web page is clearly demonstrated if the user cannot find the required information through the hyperlinks. While relationships between abstract properties are difficult to establish and validate, the relationships between observable phenomena are often self-evident in the context of the system. For example, when the incorrectness of an HTML file of a Web page is demonstrated by the fact that contains a large number of broken links, the usability of the system will be poor because the user would not be able to find the information through the hyperlinks. This example shows that if the observation of one phenomenon implies the occurrence of another phenomenon, the corresponding abstract properties must have an implication relationship. Many authors have used such a rationale in the construction of quality models. Unfortunately, such rationale has never been included in existing quality models. Such information is believed to be of use. For example, in the design and analysis of safety critical systems, it is important for the developers not only to know if the system is safe, but also to know how the system will behave if a certain event happens. This provides the crucial information for software testers to develop test cases to check if the system correctly implements the safety as designed.

The following requirements are identified for the representation of quality models.

Requirements 1: A quality model should explicitly associate quality attributes / quality carrying properties to the components of the system.

Requirements 2: A quality model should associate abstract properties with observable and verifiable phenomena of the components / system.

Requirements 3: A quality model should present the rationale of the relationships between the properties. Such rationales can be system specific and should be able to be verified and validated in the context of the system.

Figure 3.3 illustrates the notations, which consist of two principal elements: nodes and links. Each node contains three basic elements:

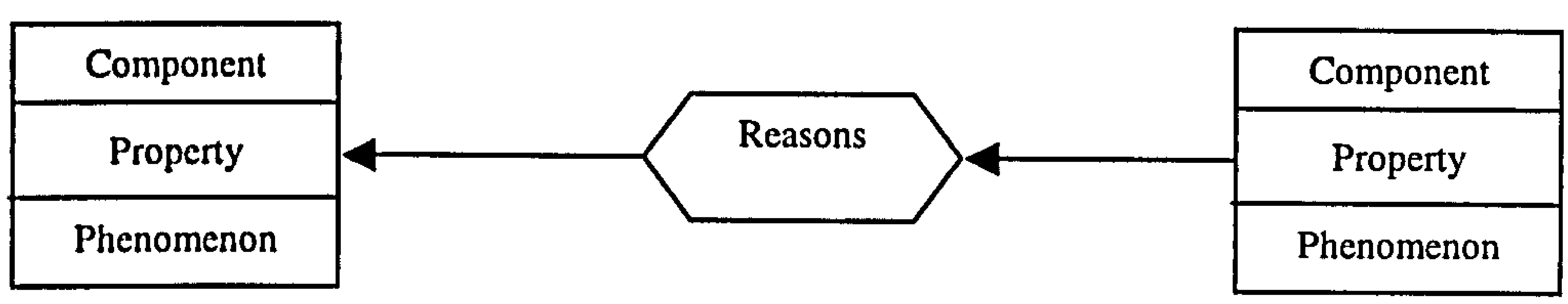

Figure 3.3 Proposed notation for representation of quality models

1. the component of the system;
2. the quality-carrying properties of the component; and

3. the observable phenomena of the property. The links are directed arcs between the nodes. A link from node A to node B means that the observation of the phenomenon on node A implies the occurrence of the phenomenon on node B. Each link can contain an optional annotation for the reasons why the two nodes are related.

3.6 Summary

This chapter presents a new quality modelling method for deriving system-specific quality models from systems' architectural design. The architectural model provides the details of the structure of the system, the information flow between the components of the system and between the users and the system. This enables a systematic analysis of the hazards of the system by considering all possible failures of each component and each connection between the components, the possible causes of the failures and the consequences.

A hazard, which is a concept borrowed from the analysis techniques developed for the safety critical systems, highlights a sensitive quality attribute from the users' point of view. The causes of a hazard give the insight of the technical issues related to the quality attribute and link to the quality attributes. The consequences of a hazard provide the foundation for assigning a weight to the importance of the quality attribute. The failure mode and effect analysis method originally developed for hazard analysis of safety critical systems is adapted for the analysis of WIS. The result of this adapted method can be directly used to construct quality models of WIS.

The chapter developed a quality model using the proposed method to illustrate that the modelling method is applicable and easy to use.

Chapter 4: Measuring Website Navigability

4.1 Introduction

The rapid advancement of the World Wide Web has created new lifestyles, such as searching for information and browsing through various products by using the World Wide Web as a universal tool. However, users have been found to experience severe difficulties in using the websites. One of the biggest difficulties is website navigation (Carmel, et al, 1992).

For a website, Nielsen (1999) claims that navigation design should help users answer three fundamental questions when browsing the site. They are: 'Where am I?', 'Where have I been?' and 'Where can I go?'.

Navigation deign is then the process of determine a path to be travelled through a chosen environment (Darken and Siebert, 1993). By 1997, much of the existing navigation research literature deals with virtual reality (Bachiochi, et al, 1997). Until recently, usability engineering has put Web users at the centre of focus. The Web therefore has become a major concern of navigation research as users become frustrated with poor designs. In fact, navigation is such an important feature that Krug (2000) stated "navigation is not just a part of the websites; it *is* the website".

This study is concerned with the measurement of the quality of website navigation, which is defined as the ease with which the users find the target information by moving through a website. The following steps will be taken to develop quality metrics of WIS:

- *Quality attribute definition*: give an accurate definition to the targeted attribute.
- *Metric definition*: derive and formally define the direct and/or indirect measures

- *Metric implementation*: implement software tools that measure the WIS using the metrics
- *Metric validation*: validate that a candidate metric is indeed a proper measure of the quality attribute of WIS.

The development of WIS navigability metrics is based on the hypothesis of strong correlation between website structural complexity and navigability. Existing empirical research showed a strong negative relationship between website structural complexity and navigability (Spool, et al, 1999; Nielsen, 1997; Navarro and Khan, 1996). It is well known that link numbers and distribution are the main website structural complexity factors (Spool, et al, 1999). Can website structural complexity metrics be used to indirectly measure navigability? This study proved this hypothesis under certain conditions.

To measure website structural complexity, a Web graph model is applied. Dhyani, et al (2002) surveyed the existing research on Web metrics and found that there exists considerable research based on the properties of Web graphs. This study also demonstrated that navigability metrics could be defined based on website structural properties in the graph model.

The evaluation of the navigability metrics consists of the following two steps.

- ♦ The evaluation of the metrics as websites structural complexity metrics
- ♦ The evaluation of the metrics as valid metrics for website navigability

Weyuker's (1988) axiom system of software complexity was used to evaluate the metrics as complexity measures. Since Weyuker's axiom system was proposed for software (program) complexity, it must be adapted for web-based systems before use.

The evaluation of using the metrics in the comparison of website navigability was conducted by empirical studies that involve end-users because website navigation is decided by users' effective and efficient completion of the tasks of browsing/searching the sites. Questionnaire-based surveys were conducted in the empirical studies.

The remainder of the chapter is organised into seven sections. Section 4.2 discusses the concept of Website navigability. A group of six metrics of website structural complexity will be identified and defined in Section 4.3. Section 4.4 assesses the metrics against adapted Weyuker's axiom system of software complexity. Section 4.5 validates the metrics via empirical study based on user-centred questionnaires. Section 4.6 summarises the chapter.

4.2 Concept of website navigability

"Navigation" comes from two Latin words: navis (ship) and agrere (to drive). According to the Merriam-Webster Dictionary, the general meaning of "navigation" is "to steer a course through a medium, to get around, move, to make one's way over or through and to operate or control the course of." The main purposes of navigation therefore are figuring out where you are and moving from one place to another. Here Web navigability is defined as:

> the ability provided by Web-based systems to aid the users to locate themselves and move around the website easily for certain purposes, e.g. finding information, completing transactions, etc.

4.3 Definition of metrics

4.3.1 Website structure model

The structure of the website deals with how the information is broken down into chunks and the relationships between them (Barfield, 2004). The abstract structure is dependent on the nature of the information. Some 'standard' structures of WIS (Lowe and Hall, 1999; Powell, 1998; Fleming, 1998) are linear (Figure 4.1), hierarchical (Figure 4.2) and networks (Figure 4.3). In Figure 4.1-4.3, node S is the Starting node, or homepage. A few have used grid structure (Lowe and Hall, 1999). However, the most common accepted is hybrid structure, which is the combination of different structure types (Powell, 1998). A key point in deciding to use hybrid structures is the fact that large, complex websites are usually made up of information of different natures. In addition, information is distributed geographically at different places. Different parts of the website may use different types of structure, but a well designed website usually adopts one key structure that holds the whole system together. Such a key structure is usually a hierarchy, with the homepage as the starting node, or root.

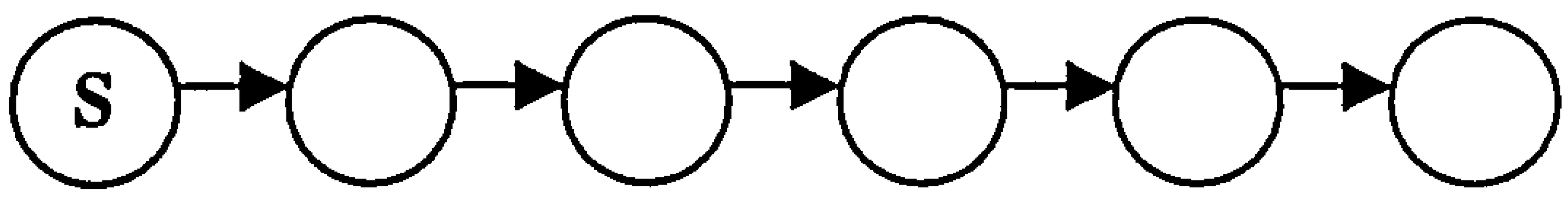

Figure 4.1 Linear structure

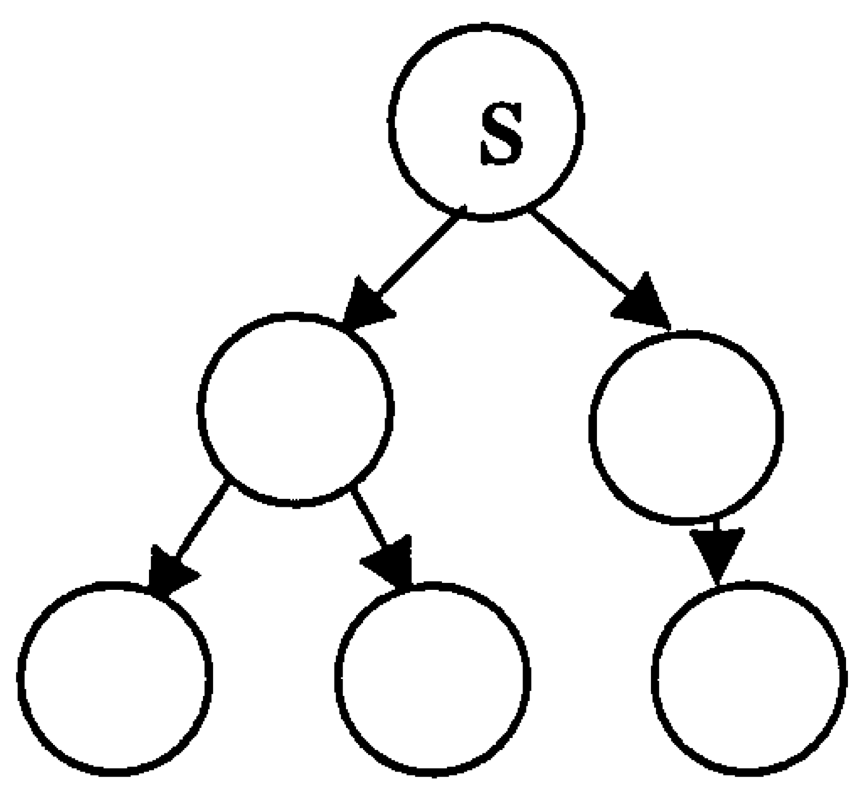

Figure 4.2 Hierarchical structure

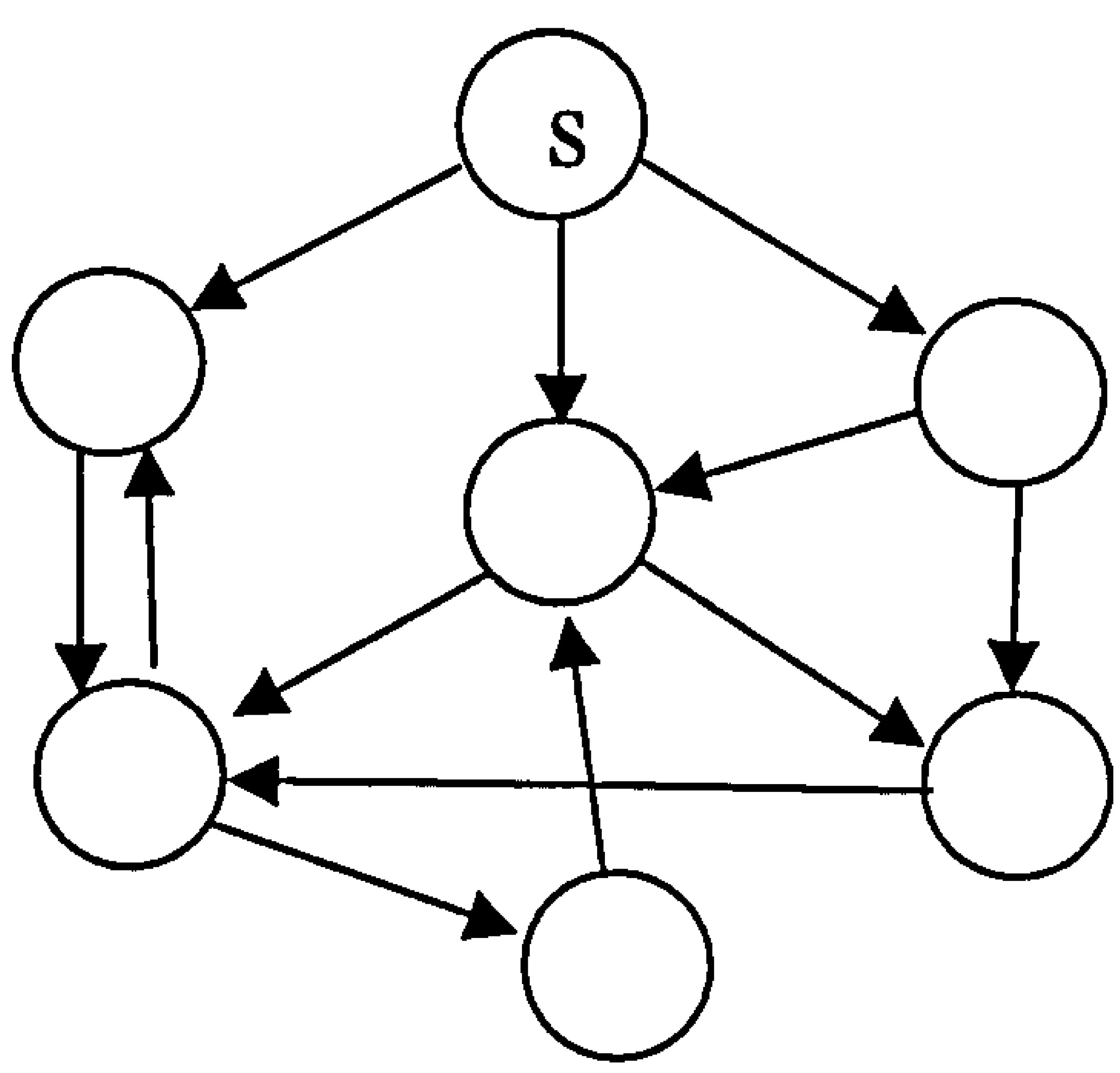

Figure 4.3 Network structure

Figure 4.4 illustrates the part of first three levels of hierarchical structure of Manchester Metropolitan University's site (http://www.mmu.ac.uk), where the nodes are

S: Starting node, or homepage.

1: prospectus

2: Study at MMU

3: Student life

4: Business

5: A-Z index

6: Research

7: Contact

8: About MMU

… …

11: How to apply

12: Browse by subject

13: Advice

14: Order a printed copy

15: Entry requirements

16: Visit MMU

… …

61: News

62: Opportunities

63: RAE results

64: Support

… …

… …

For the sake of simplicity, the figure ignores all links between the homepage and level-3 nodes (each level-3 node has a link back to the homepage). The Figure only illustrates the macro-structure of the hierarchy, and ignores some links embedded in the context.

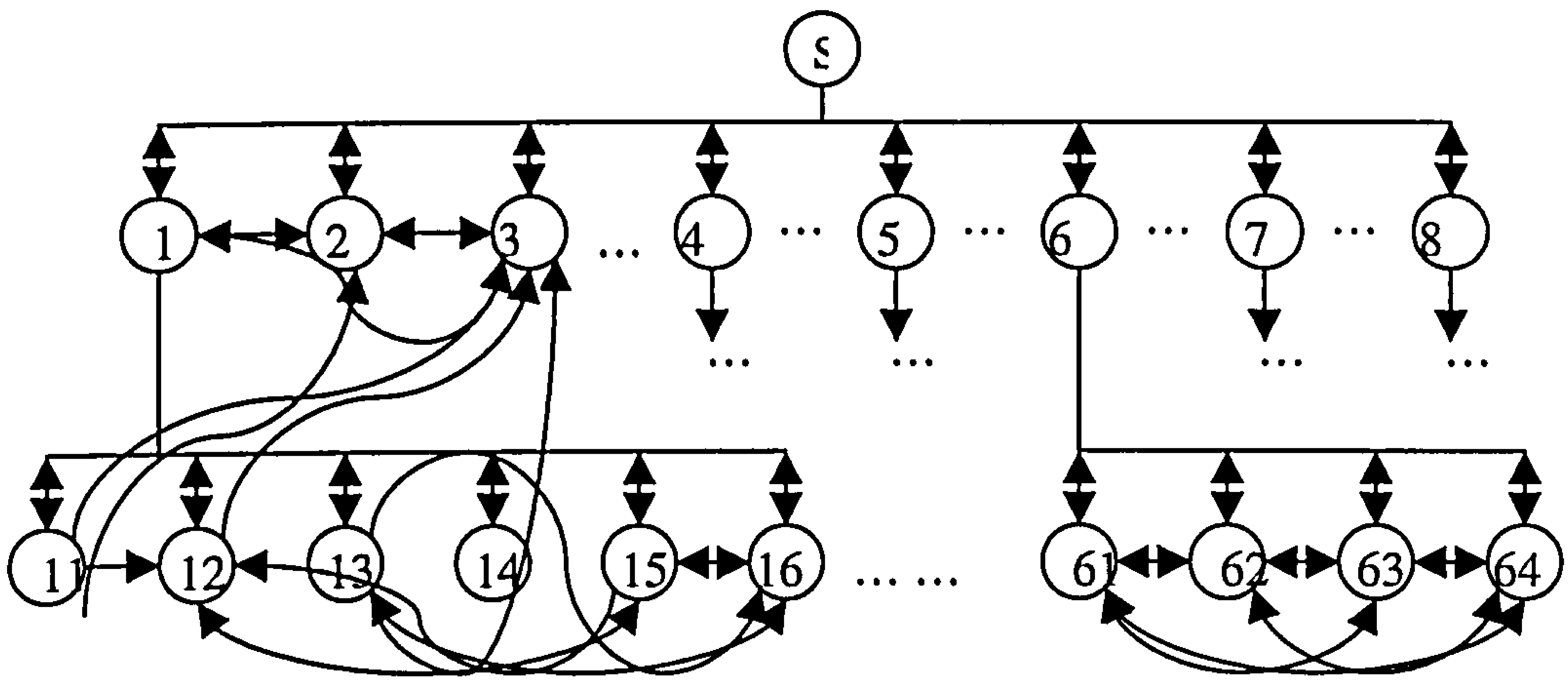

Figure 4.4 Navigational structure of a university site

4.3.2 Website structure and navigation

Website navigability is concerned with users' relationship with information: what they are going to do with it, how they will access it and use it. When studying the Web structure, it is easy to think of 'clickable links' because that is the tangible face of Web structure (Barfield, 2004). It is because clickable links determine how the user moves around in the interactive structure. It plays an important part in information structuring, interaction and thus, navigability. The more complex the structure is, the easier the users may feel confused and get lost, the poorer the navigability is.

4.3.3 Website structural complexity and navigability

A metric is "a quantitative measure of the degree to which a system, component, or process possesses a given attribute" (IEEE, 1992). Although Web navigability, as an attribute of usability, reflects users' subjective feelings while browsing the site (Nielsen, 1997), the use of 'metrics' to measure the navigability can provide us an efficient, effective and reliable means to evaluate a structural design of a website and to compare various designs. However, it is difficult to measure navigability directly.

McCracken and Wolfe (2004) stated that website design can be regarded as two inseparable parts: Website structure design and content design. Structure design can be categorised into three main steps: website organisational schemes, website organisational structure and visual organisation. Organisational scheme is a classification system for content items (placing items into groups), and organisational structure defines the relationships between the groups created using organisational schemes. The third step, visual structural organisation, is the implementation of the first two steps. Further, the major form of visual structural organisation is website navigation design.

McCracken and Wolfe's work indicates that website navigability has a strong relationship with the website structure design. This statement is fully supported by (Nielsen, 1997; Barfield, 2004). Empirical studies also showed the strong relationships between the website structure and navigability (Spool et al, 1999).

This study further defines and investigates website structural complexity metrics as indirect measures for website navigability.

Following many other authors in the literature of web metrics (see chapter 2 for details), the following model of web applications is adopted to represent a website's structure.

Definition 1. (Structural model of website)

A website's structure is a directed graph $G = (V, E, S)$, where $V \neq \emptyset$ is the set of nodes representing the web pages of the website; $E \subseteq V \times V$ is the set of edges representing links between web pages; $(u, v) \in E$ means that there is a hyperlink in the page u to page v; $S \in V$ is the start node of the graph, i.e. the home page of the website. The directed graph must also satisfy the condition that all nodes v in V are reachable, i.e. there is at least one path from the home page S to node v.

4.3.4 Definition of website structural complexity metrics

Structural complexity emerges from the relationships among the pages of the website. The most basic and important relationship is that a page is linked to another through hyperlinks. The hyperlinks between web pages of a website form the navigational paths through which users browse the website to find the required information. The more complex the way in which the web pages are inter-linked, the more likely the users are to get lost in the information ocean, and hence, the more difficult users feel in navigation. In this study, the structural complexity metrics are therefore based on the study of website links.

The structurally simplest system consists of a single page with no links. For more complex systems, structural complexity depends on the structure of the graph model of the website. In general, website structural complexity can be defined as mapping from the structure of websites to non-negative numbers as follows.

Definition 2 (Website structural complexity metrics)

A website complexity metric M is a mapping from the structural models of websites as defined in Definition 1 to non-negative numbers.

For a given web page, this study distinguishes the number of coming in links (in-link) and going out links (out-link) from the page. *In-link* of a page is the count of pages in the website that contains at least one hyperlink to the page, and *out-link* of a page is the count of pages in the website that the page has a hyperlink to. The notion of in-link and out-link can be formally defined as follows.

Definition 3. (In-link and out-link)

Let $G=(V,E,S)$ be the structure of a given website and $v \in V$. in-link and out-link can be formally defined as follows.

$$\textit{In-link}_G(v) = \left\|\{u \mid (u,v) \in E\}\right\|, \text{ and}$$

$$\textit{Out-link}_G(v) = \left\|\{u \mid (v,u) \in E\}\right\|.$$

For example, in Figure 4.5, in-link of page A is 1, and out-link of page A is 3.

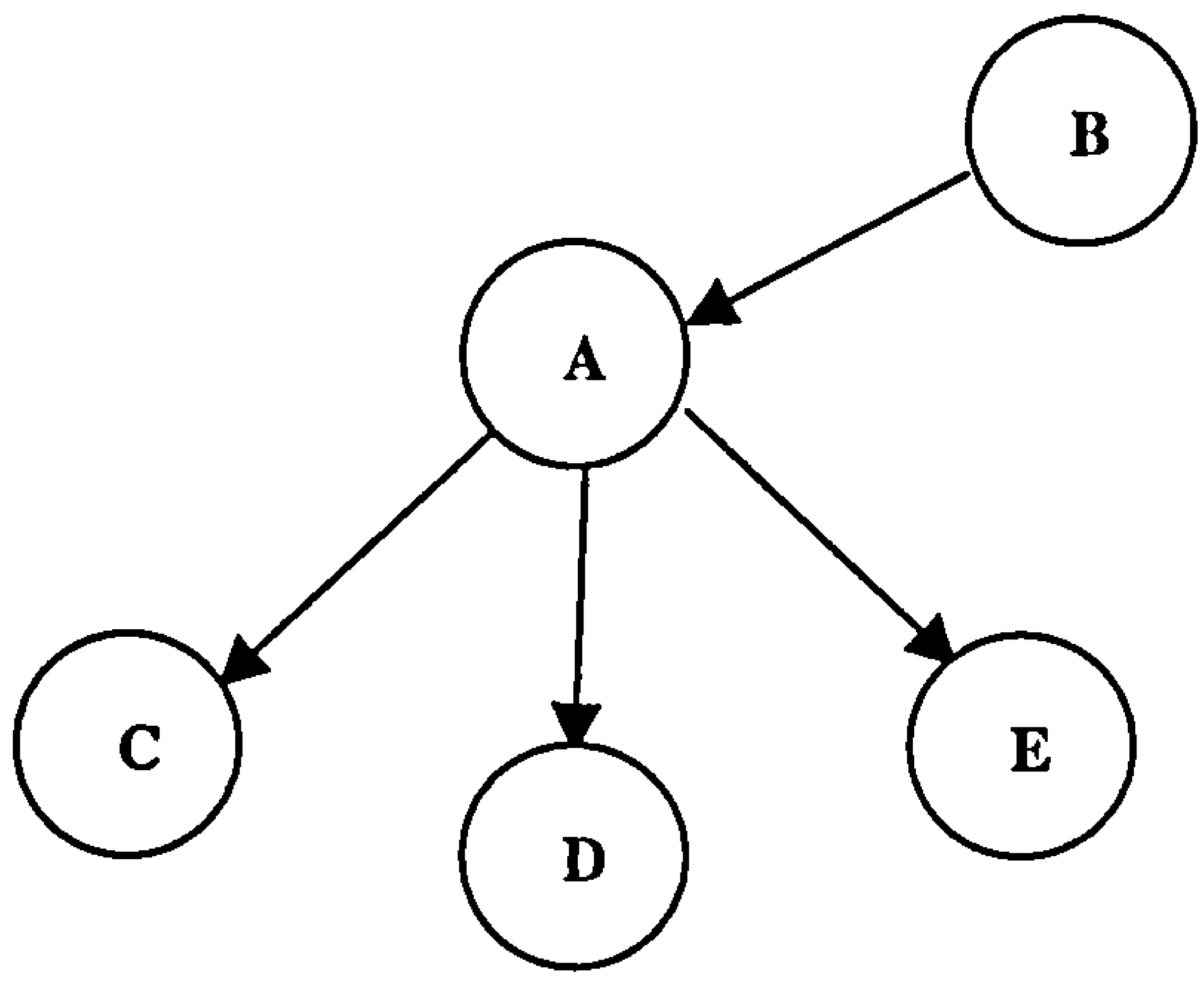

Figure 4.5 In-link and Out-link

Intuitively, in-link, i.e. the number of coming in links, indicates how easy it is to get to a page. A large in-link is generally confined to pages that performed simple functions reused throughout the website. Consequently, a large in-link does not prove to be an important complexity indicator. On the other hand, out-link, i.e. the number of out going links, indicates how easy it is to get lost since each going out link represents a choice for the next step in navigation. For that reason, in-link and out-link are important indicators of navigability. Therefore, the first candidate formula for measuring website structural complexity (WSC) is the following.

$$WSC_1(G) = \sum_{x \in V} Out\text{-}link_G(x) \qquad \textbf{Eq 4.1}$$

where G is the structure model of the website. From graph theory, for all directed graphs, the sum of in-links of all nodes is equal to the sum of out-links, which is equal to the total number of clickable links. Therefore, the first metric can be defined as follows.

$$WSC_1(G) = \sum_{x \in V} In\text{-}link_G(x) = \|E\| \qquad \textbf{Eq 4.2}$$

WSC_1 catches the intuition that a small website with fewer pages and links is less complex than a large web site that has hundreds even thousands of pages and links. However, for comparison purposes, it is desirable to know its relative complexity taking size into consideration. Dividing the overall complexity by the number of pages gives a normalized complexity.

$$WSC_2(G) = \frac{WSC_1(G)}{\|V\|} \qquad \textbf{Eq 4.3}$$

By graph theory,

$$WSC_2(G) = \frac{\sum_{x \in V} Out\text{-}link(x)}{\|V\|} = \frac{\|E\|}{\|V\|}.$$

Informally, WSC_2 defines structural complexity as the average number of links per page.

As suggested in (Jin, Zhu and Hall, 1997), the number of independent paths in a hyperlinked network of web pages can be used as a complexity metric. Let *NOIP*(*G*) denote the number of independent paths in a graph model *G*. The third metric is defined as follows.

$$WSC_3(G) = NOIP(G) \qquad \textbf{Eq 4.4}$$

According to graph theory, the number of independent paths in a directed graph *G* can be calculated by the following formula (Zhu and Hall, 1993; Feghali, et al, 1994).

$$NOIP(G) = e - n + d + 1 \qquad \textbf{Eq 4.5}$$

where $e = \|E\|$ is the total number of links in the graph, $n = \|V\|$ is the number of pages in the graph and d is the number of dead end nodes in the graph. A node is a dead end node if the out-link of the node is 0, $d = \left\|\{u \in V \mid Out\text{-}link(u) = 0\}\right\|$.

$$WSC_3(G) = e - n + d + 1. \qquad \textbf{Eq 4.6}$$

A relative complexity metric can also be defined based on the number of independent paths as follows.

$$WSC_4(G) = {WSC_3(G)}/{n} = (e - n + d + 1)/n \qquad \textbf{Eq 4.7}$$

Because $e = \|E\| = \sum_{x \in V} Out\text{-}link(x) = \sum_{x \in V} In\text{-}link(x)$, and usually $d \ll n$, it is easy to prove the following.

$$WSC_2(G) \geq WSC_4(G) \geq WSC_2(G) - 1 \qquad \textbf{Eq 4.8}$$

Not only does the number of out-links affect structural complexity, but also the distribution of the links within a website. In the discussion of software structural complexity measurement, Belady and Evangelisti (1981) applied the interconnection matrix representation of partition to their study and suggested that complexity increases as the square of connections (fan_{out}), where fan_{out} is number of the calls from a given module. In website designs, all pages are connected by hyperlinks. This leads to the following metrics, WSC_5, for website structural complexity, and WSC_6 for the relative complexity metric.

$$WSC_5(G) = \sum_{x \in V} Out - link_G^2(x) \qquad \textbf{Eq 4.9}$$

$$WSC_6(G) = \frac{\sum_{x \in V} Out - link_G^2(x)}{\|V\|} \qquad \textbf{Eq 4.10}$$

4.4 Axiomatic assessment of the metrics

4.4.1 Axioms of software complexity

In this section, the website complexity metrics defined in the previous section will be assessed against Weyuker's axiom system as proper complexity metrics.

Several axiomatic systems of software complexity have been developed in the literature (Prather, 1984; Fenton and Witty, 1986; Weyuker, 1988). Although the research on axioms of software complexity is still in its infancy and has aroused critics, it does provide a theoretic guideline to preliminarily assess the validity of the metrics. Weyuker's axioms were proposed to characterize ideal complexity metrics of computer software. Compared with other work, Weyuker's axiom system provides more evidence and has been well-accepted by the community.

These axioms are quoted as follows. Let M denote a software complexity metric and P, Q, R denote software systems.

Axiom 1. Not all software systems are of the same complexity. Formally, $\exists P, Q.(M(P) \neq M(Q))$.

Axiom 2. If c is a non-negative number, then there exist only finitely many P such that $M(P) = c$. Formally, $\forall c > 0.\exists n < \infty.\left(\left\|\{P | M(P) = c\}\right\| = n\right)$.

Axiom 3. There are distinct software systems that are of the same complexity. Formally, $\exists P, Q.(P \neq Q \wedge M(P) = M(Q))$.

Axiom 4. There exist functionally equivalent P and Q that are of different complexity. Formally, $\exists P, Q.(P \equiv Q \wedge M(P) \neq M(Q))$ where $P \equiv Q$ means that software P is functionally equivalent to Q.

Axiom 5. The complexity of the composition of components is greater than or equal to the complexities of the components. Formally, $\forall P, Q.(M(P;Q) \geq M(P) \wedge M(P;Q) \geq M(Q))$.

Axiom 6. The replacement of a component with another component of the same complexity may change the overall complexity of the system. Formally, $\exists P, Q, R.(M(P) = M(Q) \wedge M(P;R) \neq M(Q;R))$.

Axiom 7. Permuting the order of the statements in software may change its complexity. Formally, $\exists P, Q.(M(P) \neq M(\tilde{P}))$, where $\tilde{P}$ is obtained from P by permuting P's statements.

Axiom 8. Renaming the identifiers in a software system does not change its complexity. Formally, $\forall P, Q.(M(P) = M(\overline{P}))$ where $\overline{P}$ is obtained from P by systematically renaming some identifiers in P.

Axiom 9. The complexity of the composition of two components may be greater than the sum of the complexities of the components. Formally, $\exists P, Q.(M(P) + M(Q) < M(P;Q))$.

Notice that, Weyuker's axioms were originally proposed to characterize ideal complexity metrics of computer programs. Some formal representations of axioms are specific to high level source code. Thus, they need to be adapted to be suitable for software in the form of web pages linked by hyperlinks in order to assess Web complexity metrics.

Definition 4 (Functionally Equivalence of Structure Models)

Let $G_1 = <V_1, E_1, S_1>$ and $G_2 = <V_2, E_2, S_2>$ be two graph models of websites. They are functionally equivalent, write $G_1 \equiv G_2$, if and only if $V_1 = V_2$ and $S_1 = S_2$.

By Definition 4, two websites are functionally equivalent, if they contain the same information, but may be inter-linked differently. Figure 4.6 shows an example, in which functionally equivalent graph models have different structures.

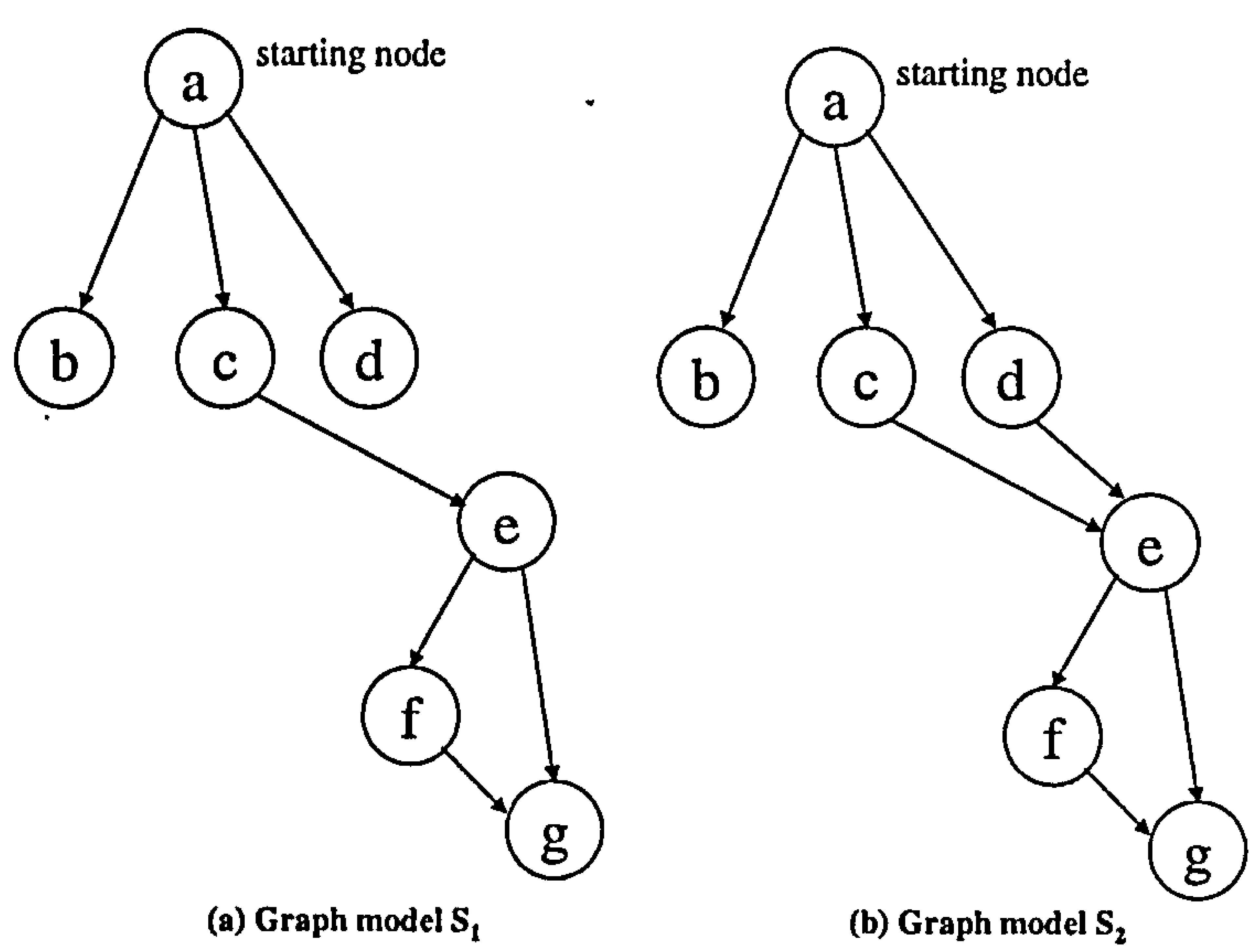

Figure 4.6 Example of functionally equivalent structures

Definition 5 (Composition of websites: $P;Q$)

Let $G_1 = <V_1, E_1, S_1>$ and $G_2 = <V_2, E_2, S_2>$ be two structure models of websites, and $V_1 \cap V_2 = \varnothing$. $G_1; G_2$ is defined as $<V_1 \cup V_2, E_1 \cup E_2 \cup \{(p, S_2)\}, S_1>$, where $p \in V_1$.

Informally, the composition of two websites is to link these two websites by a hyperlink. Of course, in general, a number of links are added between the websites. By Definition 5, the composition of two websites P and Q is to link these two websites by a hyperlink between a node in P and the starting node of Q. An example of $(P;Q)$ is illustrated in Figure 4.6(a), where P's nodes are $\{a, b, c, d\}$ and Q contains nodes $\{e, f, g\}$.

Definition 6 (Permutation of orders)

Let $G_1 = <V_1, E_1, S_1>$ and $G_2 = <V_2, E_2, S_2>$ be two structure models of websites and $V_1 = V_2$. G_2 is obtained by permuting the representation order of G_1 if $\|E_1\| = \|E_2\|$, and $G_2 = \tilde{G}_1$.

By Definition 6, permuting the order of the representation of information means changing the links between the pages without adding or deleting any hyperlinks. Therefore, the number of links between the websites must remain unchanged.

Definition 7 (Renaming)

A website W_1 is obtained from website W_2 by renaming, if W_1 is obtained by changing the texts associated to the hyperlinks and the titles of the pages of W_2.

Obviously, the following property of renaming can be reached.

Proposition 1. Let W_1 be obtained from W_2 by renaming, and G_1 and G_2 be the structure models of W_1 and W_2, respectively. We have that $G_1 = G_2$.

In the following, website structural complexity metrics will be assessed by examining the axioms one by one.

4.4.2 Assessment of Website Structural Complexity (WSC) metrics

Here, the website structure complexity metrics developed in section 4.4 will be assessed against the adapted Weyuker's axioms with the definitions given above.

Theorem 1. WSC_1 satisfies Axioms 1, 2, 3, 4, 5, 8 and 9. It does not satisfy Axioms 6 and 7.

Proof.

(1) By Definition of WSC_1, it satisfies Axiom 1.

(2) WSC_1 satisfies axiom 2 because for all connected graphs, the number of links must be greater than or equal to the number of nodes minus 1. If $WSC_1(P) = c$, P can contain at most $(c+1)$ pages. There can only be a finite number of connected graphs of $(c+1)$ or less nodes and c links.

(3) By Definition of WSC_1, it satisfies Axiom 3.

(4) By Definition 5, WSC_1 satisfies Axiom 4.

(5) By Definition 6, $WSC_1(P;Q) = m+n+1$, when $WSC_1(P) = m$ and $WSC_1(Q) = n$. Hence, WSC_1 satisfies Axiom 5.

(6) WSC_1 does not satisfy Axiom 6, because by Definition 5, for all P, Q and R, if $WSC_1(P) = WSC_1(Q)$ then $WSC_1(P;R) = WSC_1(Q;R)$.

(7) WSC_1 does not satisfy Axiom 7 because by Definition 6, for all P, $WSC_1(P) = \|E\|$ $= \|\tilde{E}\| = WSC_1(\tilde{P})$, where E and $\tilde{E}$ are the sets of links of P and $\tilde{P}$.

(8) By Definition 7 and Proposition 1, WSC_1 satisfies Axiom 8.

(9) By Definition 5, for all P and Q, $WSC_1(P;Q) = WSC_1(P) + WSC_1(Q) + 1$. Hence WSC_1 satisfies Axiom 9.

Theorem 2. WSC_2 satisfies Axioms 1, 3, 4, 6, and 8, but it does not satisfy Axioms 2, 5, 7 and 9.

Proof.

(1) By Definition of WSC_2, it satisfies Axiom 1.

(2) WSC_2 does not satisfy Axiom 2. As a counter example, consider the graphs GC_n in Figure 4.7. All these graph's complexity under WSC_2 equals 1, because there are n nodes and n arcs in GC_n for all $n > 1$. Therefore, there is an infinite number of graphs G such that $WSC_2(G) = 1$.

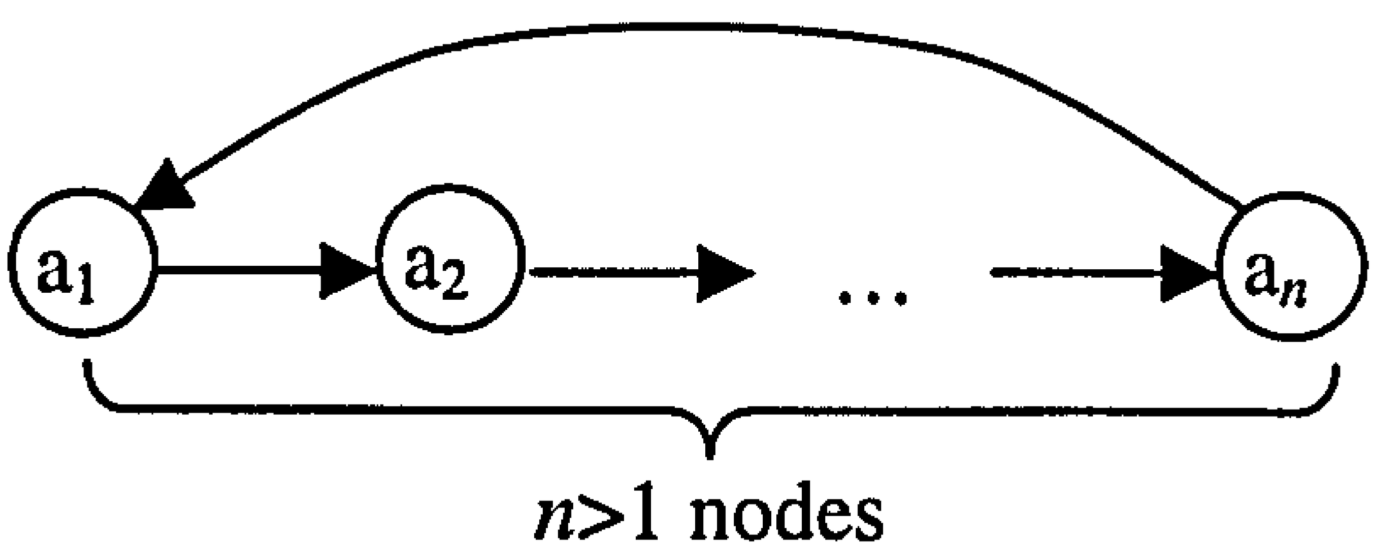

Figure 4.7 Graph GC_n as counterexamples of Axiom 2 for $WSC_{2/3/4/6}$

(3) By Definition of WSC_2, it satisfies Axiom 3.

(4) By Definition 4, WSC_2 satisfies Axiom 4. This can be proved using Figure 4.6.

(5) To prove that WSC_2 does not satisfy Axiom 5, see Figure 4.8, where $WSC_2(P) = 1/2$, $WSC_2(Q) = 4/3$, $WSC_2(P;Q) = 6/5$. $WSC_2(P;Q) < WSC_2(Q)$.

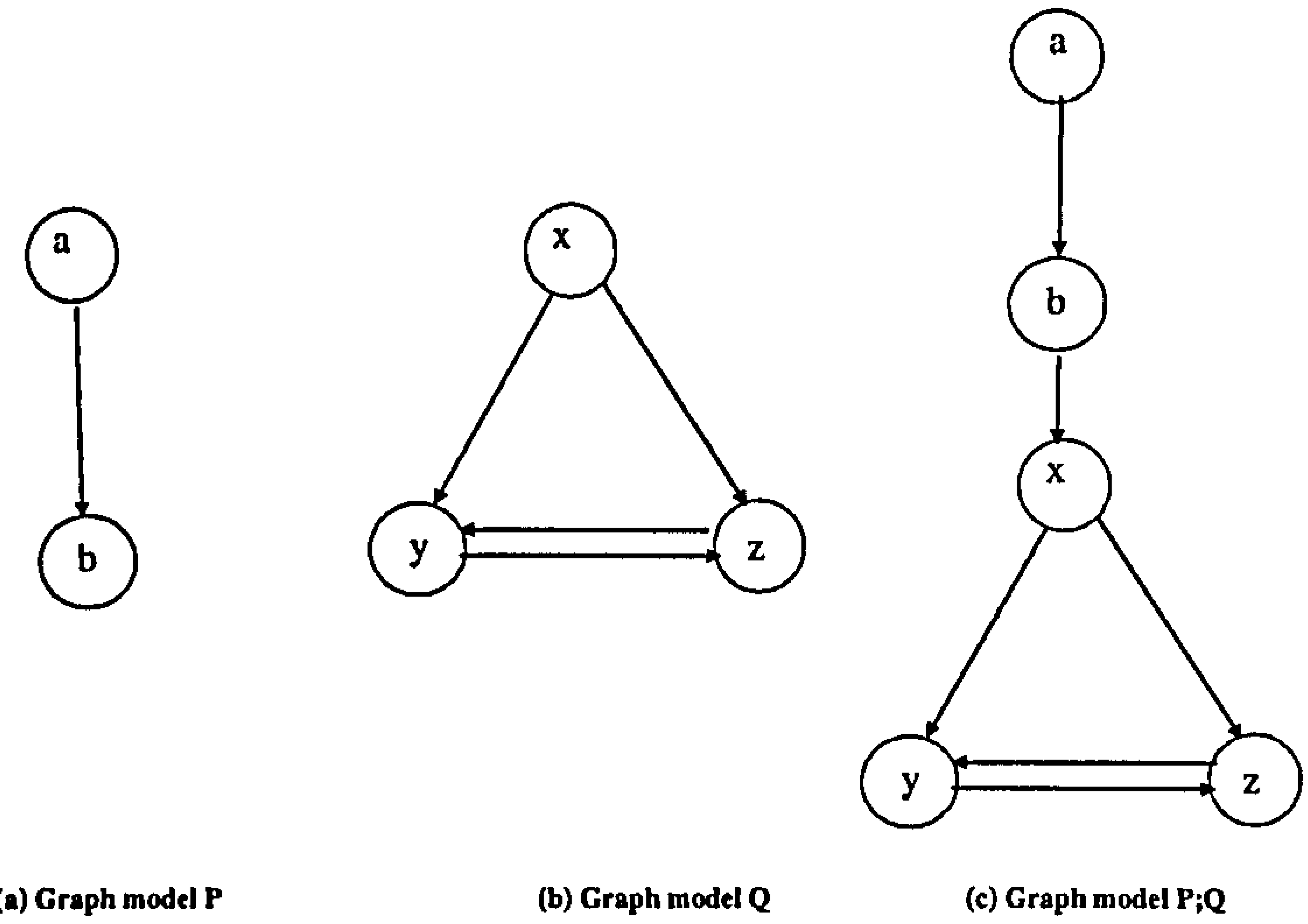

Figure 4.8 Counterexample of Axiom 5 for $WSC_{2/5}$

(6) To prove that WSC_2 satisfies axiom 6, consider the graphs in Figure 4.9. By definition, $WSC_2(P) = WSC_2(Q) = 5/3$, $WSC_2(P;R) = 11/7$ and $WSC_2(Q;R) = 6/4$, i.e. $WSC_2(P;R) \neq WSC_2(Q;R)$.

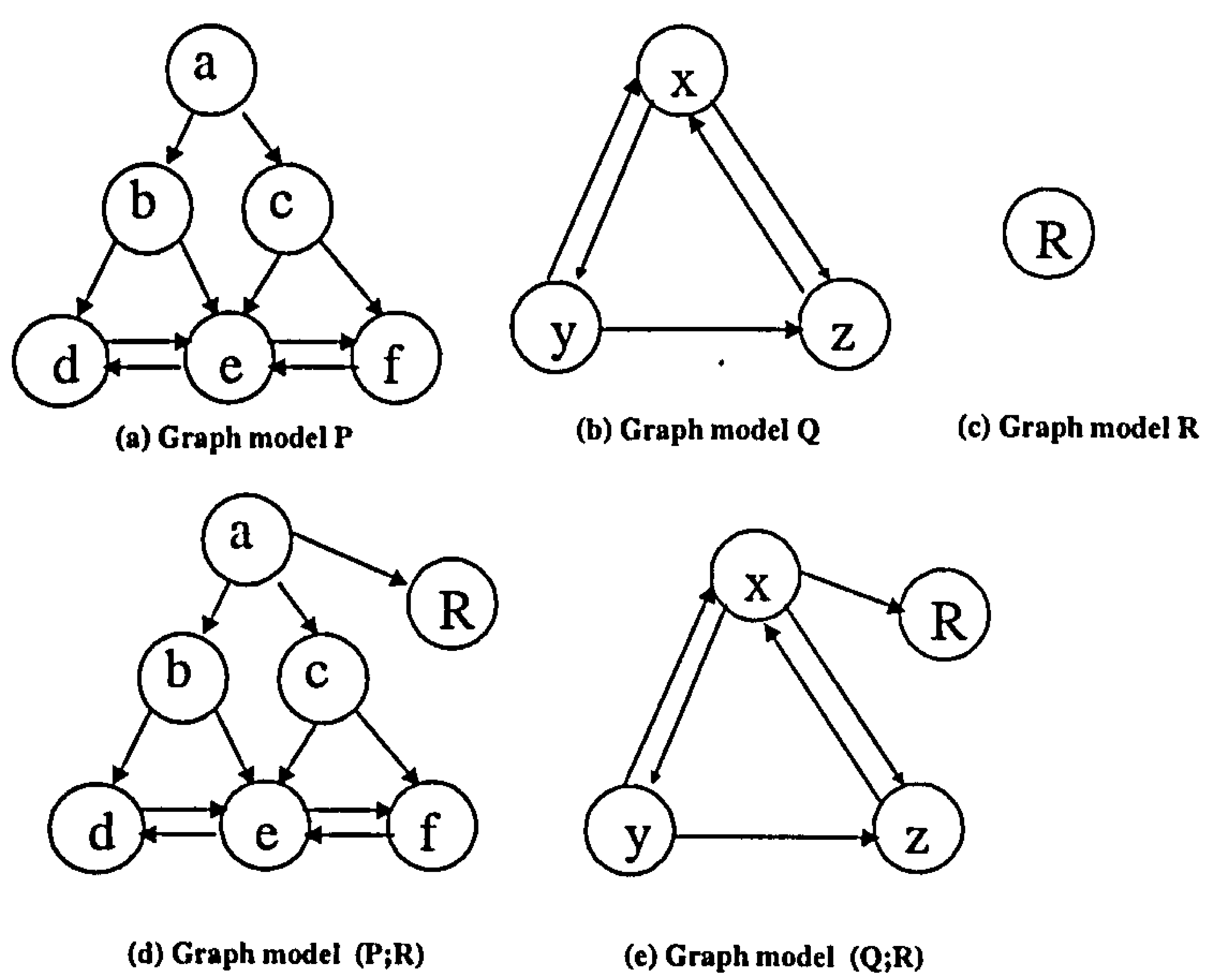

Figure 4.9 Example of Axiom 6 for WSC_2

(7) By Definition 6, it satisfies Axiom 7.

(8) By Definition 7 and Proposition 1, WSC_2 satisfies Axiom 8.

(9) To prove that WSC_2 does not satisfy Axiom 9, let e_1 and n_1 be the number of edges and nodes in a graph P; e_2 and n_2 be the number of edges and nodes in graph Q, respectively. By Definition 5, graph $P;Q$ has $(e_1 + e_2 + 1)$ edges and $(n_1 + n_2)$ nodes. It is easy to prove that for all natural numbers e_1, e_2, n_1, and n_2, $e_1/n_1 + e_2/n_2 \geq (e_1 + e_2 + 1)/(n_1 + n_2)$. That is, for graphs P and Q, $WSC_2(P) + WSC_2(Q) \geq WSC_2(P;Q)$.

Theorem 3 WSC_3 satisfies Axioms 1, 3, 4, 5, 6, 7 and 8; but not satisfies Axioms 2 and 9.

Proof.

(1) By Definition of WSC_3, it satisfies Axiom 1.

(2) Notice that for all graphs G in Figure 4.7, $WSC_3(G) = 1$. Hence, WSC_3 does not satisfy Axiom 2.

(3) By Definition of WSC_3, it satisfies Axiom 3.

(4) By Definition 1, WSC_3 satisfies Axiom 4.

(5) WSC_3 satisfies Axiom 5. Let e_1, n_1 and d_1 be the number of edges, nodes and dead end nodes in a graph P; e_2, n_2 and d_2 be the number of edges, nodes and dead end nodes in graph Q. There are two conditions (a) if the hyperlink, which links the two websites together, is between a dead node in $P(n_1, e_1, d_1)$ and the starting point in $Q(n_2, e_2, d_2)$, the graph $P;Q$ has $(e_1 + e_2 + 1)$ edges, $(n_1 + n_2)$ nodes and $(d_1 + d_2 - 1)$ dead end nodes. Therefore,

$$WSC_3(P;Q) = (e_1 + e_2 + 1) - (n_1 + n_2) + (d_1 + d_2 - 1) + 1 = WSC_3(P) + WSC_3(Q) - 1$$

For a connected graph, $WSC_3(P) < 1$ if and only if P consists of one dead node. Then

$WSC_3(P;Q) = WSC_3(Q)$ and $WSC_3(P;Q) > WSC_3(P)$. (b) if the hyperlink, which links the two websites together, is between a non-dead node in $P(n_1, e_1, d_1)$ and the starting point in $Q(n_2, e_2, d_2)$, the graph $P;Q$ has $(e_1 + e_2 + 1)$ edges, $(n_1 + n_2)$ nodes and $(d_1 + d_2)$ dead end nodes. Therefore

$$WSC_3(P;Q) = (e_1 + e_2 + 1) - (n_1 + n_2) + (d_1 + d_2) + 1 = WSC_3(P) + WSC_3(Q)$$

Thus, $WSC_3(P;Q) \geq WSC_3(P)$ and $WSC_3(P;Q) \geq WSC_3(Q)$.

(6) WSC_3 satisfies Axiom 6. In Figure 4.10, $WSC_3(P) = WSC_3(Q) = 1$, but $WSC_3(P;R) \neq WSC_3(Q;R)$.

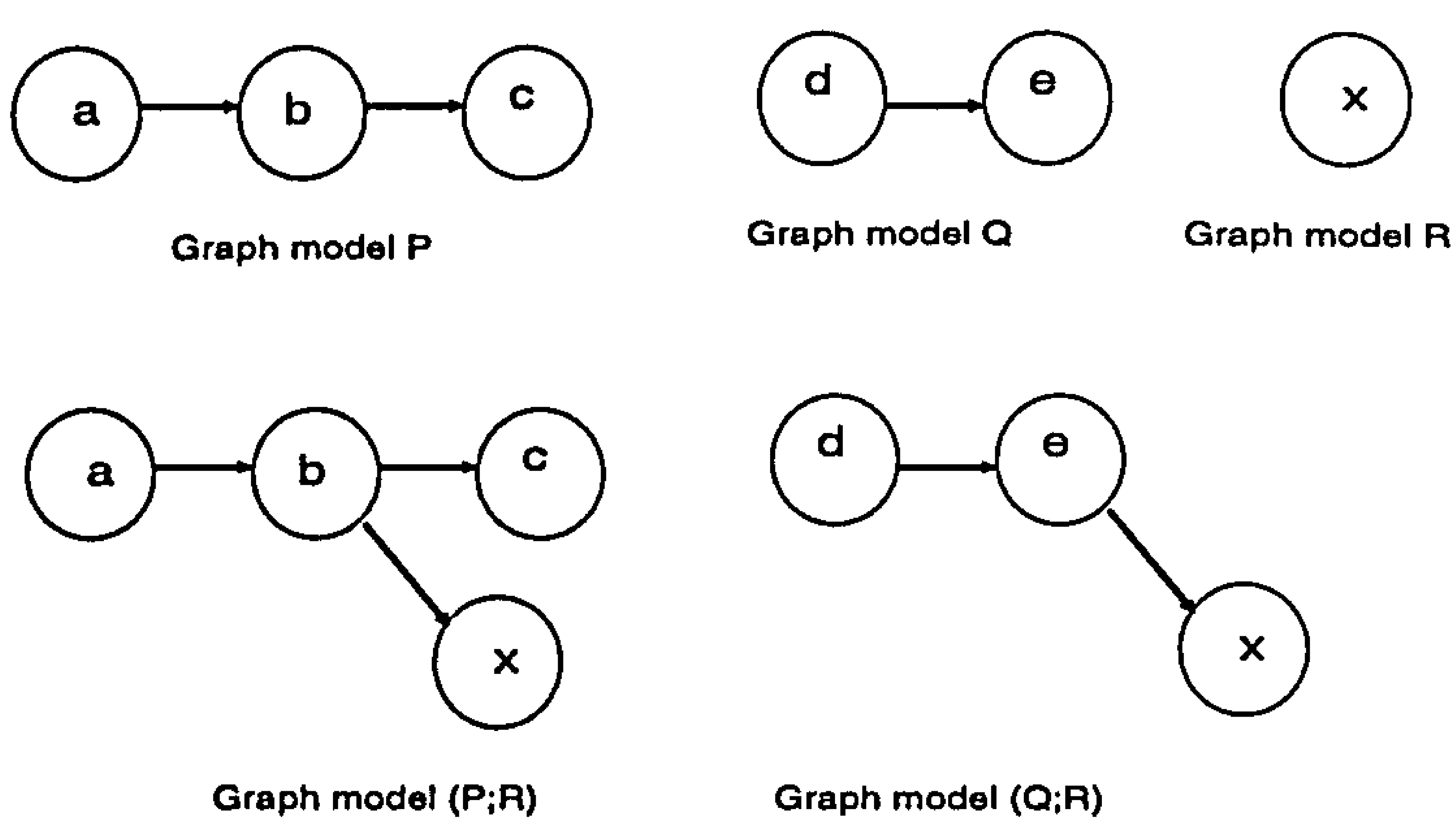

Figure 4.10 Example of Axiom 6 for WSC_3

(7) By Definition 3, WSC_3 satisfies Axiom 7.

(8) By Definition 4, WSC_3 satisfies Axiom 8.

(9) WSC_3 does not satisfy axiom 9. Let e_1, n_1 and d_1 be the number of edges, nodes and dead end nodes in a graph P; e_2, n_2 and d_2 be the number of edges, nodes and dead end nodes in graph Q, respectively. By definition 2, if the hyperlink, which links the two websites together, is between a dead node in $P(e_1, n_1, d_1)$ and the starting point in

$Q(n_2, e_2, d_2)$, the graph $P;Q$ has (e_1+e_2+1) edges, (n_1+n_2) nodes and (d_1+d_2-1) dead end nodes. Therefore

$$WSC_3(P;Q)=(e_1+e_2+1)-(n_1+n_2)+(d_1+d_2-1)+1=WSC_3(P)+WSC_3(Q)-1$$

If the hyperlink, which links the two websites together, is between a non-dead node in $P\ (n_1, e_1, d_1)$ and the starting point in $Q\ (n_2, e_2, d_2)$, the graph $P;Q$ has (e_1+e_2+1) edges, (n_1+n_2) nodes and (d_1+d_2) dead end nodes. Therefore

$$WSC_3(P;Q)=(e_1+e_2+1)-(n_1+n_2)+(d_1+d_2)+1=WSC_3(P)+WSC_3(Q)$$

In both circumstances, $WSC_3(P;Q) \leq WSC_3(P) + WSC_3(Q)$.

Theorem 4. WSC_4 satisfies Axioms 1, 3, 4, 6, 7 and 8; but not satisfies Axioms 2, 5 and 9.

Proof.

(1) By Definition of WSC_4, it satisfies Axiom 1.

(2) Notice that for all graphs G in Figure 4.7, $WSC_4(G) = 1$. Hence, WSC_4 does not satisfy axiom 2.

(3) By Definition of WSC_4, it satisfies Axiom 3.

(4) By Definition 1, WSC_4 satisfies Axiom 4.

(5) WSC_4 does not satisfy Axiom 5. In Figure 4.11, $WSC_4(P;Q) < WSC_4(P)$.

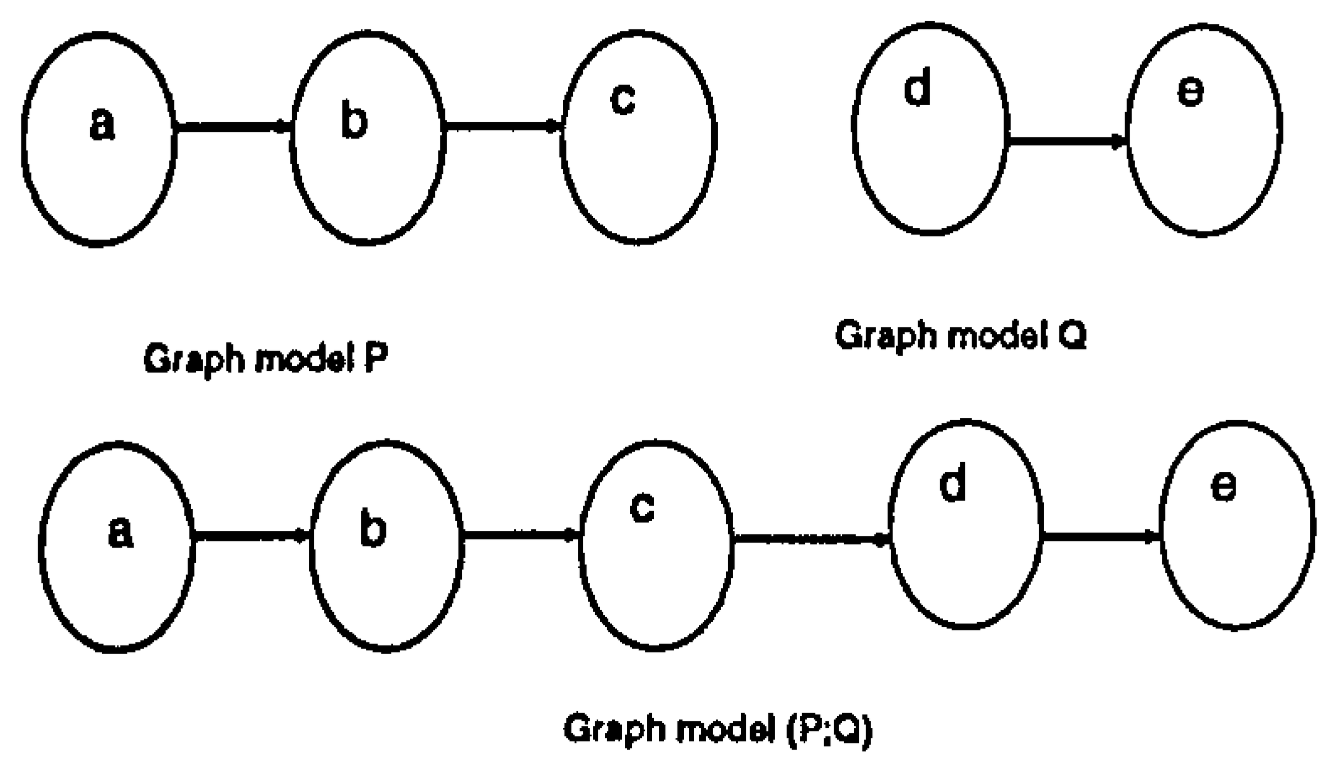

Figure 4.11 Counterexample of Axiom 5 for WSC$_4$

(6) WSC$_4$ satisfies Axiom 6. In Figure 4.12, $WSC_4(P) = WSC_4(Q)$, but it is easy to get $WSC_4(P;R) \neq WSC_4(Q;R)$.

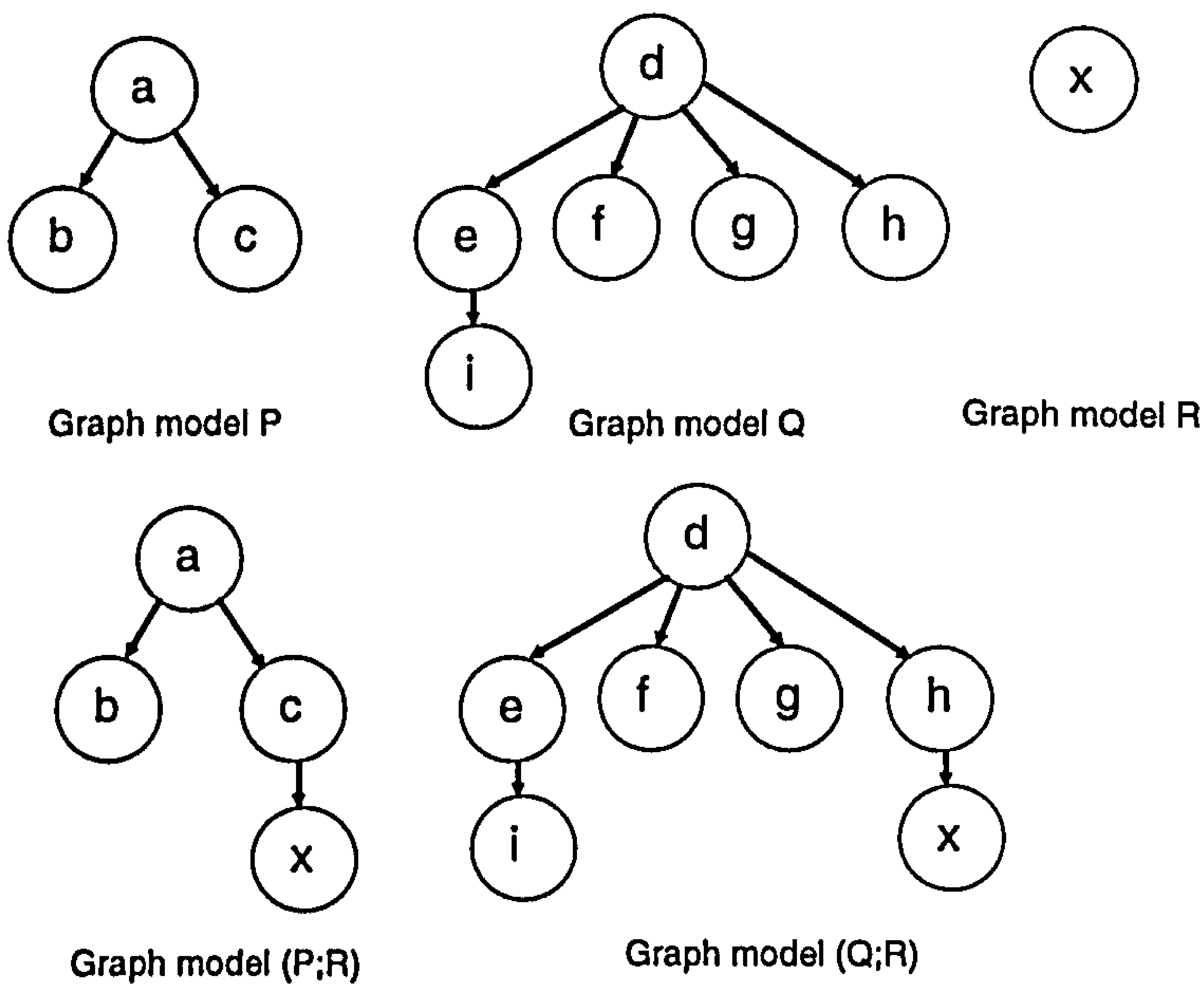

Figure 4.12 Example of Axiom 6 for WSC$_4$

(7) By Definition 3, WSC$_4$ satisfies Axiom 7.

(8) By Definition 4, WSC$_4$ satisfies Axiom 8.

(9) WSC$_4$ does not satisfy axiom 9. Let e_1, n_1 and d_1 be the number of edges, nodes and dead end nodes in a graph P; e_2, n_2 and d_2 be the number of edges, nodes and dead end nodes in graph Q, respectively. By Definition 2, if the hyperlink, which links the

two websites together, is between a dead end node in $P(n_1, e_1, d_1)$ and the starting point in $Q(n_2, e_2, d_2)$, the graph $P;Q$ has $(e_1 + e_2 + 1)$ edges, $(n_1 + n_2)$ nodes and $(d_1 + d_2 - 1)$ dead end nodes. Therefore $WSC_4(P;Q) - WSC_4(P) - WSC_4(Q) \le 0$ [2]; if the hyperlink, which links the two websites together, is between a non-dead node in $P(n_1, e_1, d_1)$ and the starting point in $Q(n_2, e_2, d_2)$, the graph $P;Q$ has $(e_1 + e_2 + 1)$ edges, $(n_1 + n_2)$ nodes and $(d_1 + d_2)$ dead end nodes. It is easy to prove that $WSC_4(P;Q) - WSC_4(P) - WSC_4(Q) \le 0$.

Theorem 5. WSC$_5$ satisfies Axioms 1, 2, 3, 4, 5, 6, 7, 8 and 9.

Proof.

(1) By Definition of WSC$_5$, it satisfies Axiom 1.

(2) WSC$_5$ satisfies Axiom 2, as for connected graphs, the number of links must be greater than or equal to the number of nodes minus 1. Therefore, For $WSC_5(P) = c$, even each node has only one out-link, P can contain a finite number, at most $(c+1)$, of nodes.

(3) By Definition of WSC$_5$, it satisfies Axiom 3.

(4) By Definition 1, WSC$_5$ satisfies Axiom 4.

(5) WSC$_5$ satisfies Axiom 5. By Definition of WSC$_5$, $WSC_5(P;Q) > WSC_5(P)$; $WSC_5(P;Q) > WSC_5(Q)$. For an extreme condition, in which graph Q contains only one dead end node, $WSC_5(P;Q)$ still greater then $WSC_5(P)$ as the number of out-links of one node in P will increase by 1.

[2] Let $X_1=e_1-n_1+d_1$, $X_2=e_2-n_2+d_2$, $WSC_4(P;Q) - WSC_4(P) - WSC_4(Q) =$ $(X_1+X_2+1)/(n_1+n_2) - (X_1+1)/n_1-(X_2+1)/n_2=(X_1+X_2+1)/(n_1+n_2)-(X_1n_2+n_2+X_2n_1+n_1)/n_1n_2=$ $-(X_1n_2^2+X_2n_1^2+n_1n_2+n_2^2+n_1^2)/n_1n_2(n_1+n_2) < 0$

(6) WSC_5 satisfies Axiom 6. Let graph $P = Q$, and both graphs contain m nodes, and nodes i has a_i out-link, $WSC_5(P) = WSC_5(Q) = \sum_{i=1}^{m} a_i^2$; Graph R has n nodes, $WSC_5(R) = \sum_{i=1}^{n} b_i^2$. By Definition 1, when connecting P and R using node a_m in P, and then $WSC_5(P;R) = a_1^2 + a_2^2 + \ldots + a_{m-1}^2 + (a_m+1)^2 + b_1^2 + b_2^2 + \ldots + b_n^2 = WSC_5(P) + WSC_5(Q) + 2a_m + 1$. Then connecting Q and R using node a_{m-1}, and then $WSC_5(Q;R) = WSC_5(P) + WSC_5(Q) + 2a_{m-1}$. If $a_m \neq a_{m-1}$, then $WSC_5(P;R) \neq WSC_5(Q;R)$.

(7) As WSC_5 takes distribution of out-link into consideration, By Definition 3, WSC_5 satisfies Axiom 7.

(8) By Definition 4, WSC5 satisfies Axiom 8.

(9) WSC_5 satisfies Axiom 9. Let graph P contain m nodes, and nodes i has a_i out-link; or $WSC_5(P) = \sum_{i=1}^{m} a_i^2$; graph Q has n nodes, and $WSC_5(Q) = \sum_{i=1}^{n} b_i^2$. By Definition 1, $WSC_5(P;Q) = a_1^2 + a_2^2 + \ldots + a_{m-1}^2 + (a_m+1)^2 + b_1^2 + b_2^2 + \ldots + b_n^2 = WSC_5(P) + WSC_5(Q) + 2a_m + 1 > WSC_5(P) + WSC_5(Q)$.

Theorem 6. WSC_6 satisfies Axioms 1, 3, 4, 6, 7, 8 and 9; but does not satisfy Axioms 2 and 5.

Proof.

(1) By Definition of WSC_6, it satisfies Axiom 1.

(2) WSC_6 does not satisfy Axiom 2 because that for all graphs G in Figure 4.7, $WSC_6(G) = 1$.

(3) By Definition 1, WSC_6 satisfies Axiom 3.

(4) WSC_6 satisfies Axiom 4. To prove this, consider the graphs in Figure 4.13. $WSC_6(P) = {}^{15}/_{7}$; $WSC_6(Q) = 3$, although graphs P and Q are functionally equivalent.

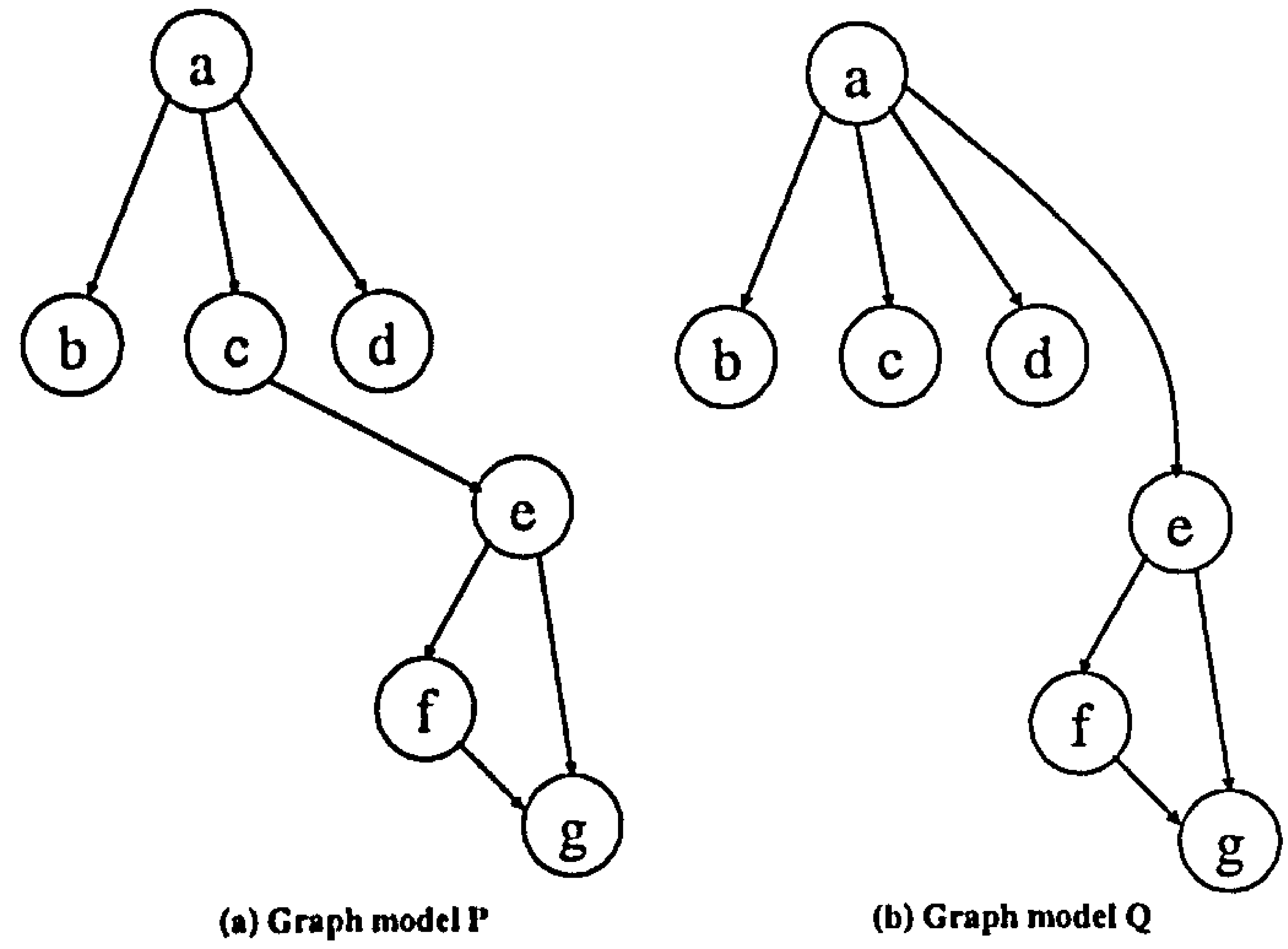

Figure 4.13 Example of Axiom 4 for WSC_6

(5) WSC_6 does not satisfy axiom 5. A counterexample is given in Figure 4.14, where $WSC_6(P) = {}^{4}/_{3}$, $WSC_6(Q) = 0$, $WSC_6(P;Q) = {}^{5}/_{4}$. Hence, $WSC_6(P;Q) < WSC_6(P)$.

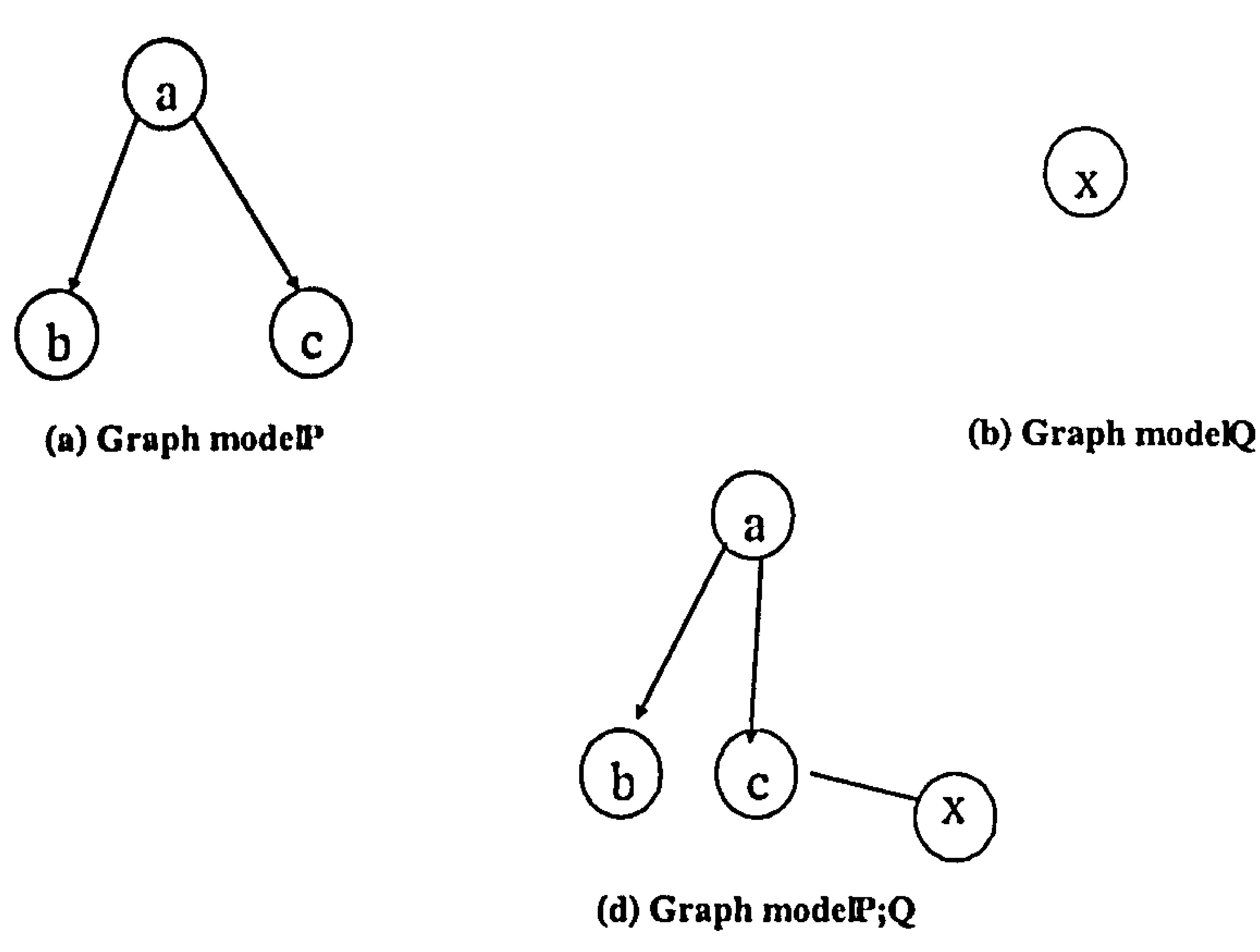

Figure 4.14 Counterexample of Axiom 5 for WSC_6

(6) WSC_6 satisfies Axiom 6. To prove this, consider the graphs given in Figure 4.15. $WSC_6(P) = WSC_6(Q) = 9/4$, $WSC_6(P;R) = 2$, $WSC_6(Q;R) = 14/5$, i.e. $WSC_6(P;R) \neq WSC_6(Q;R)$.

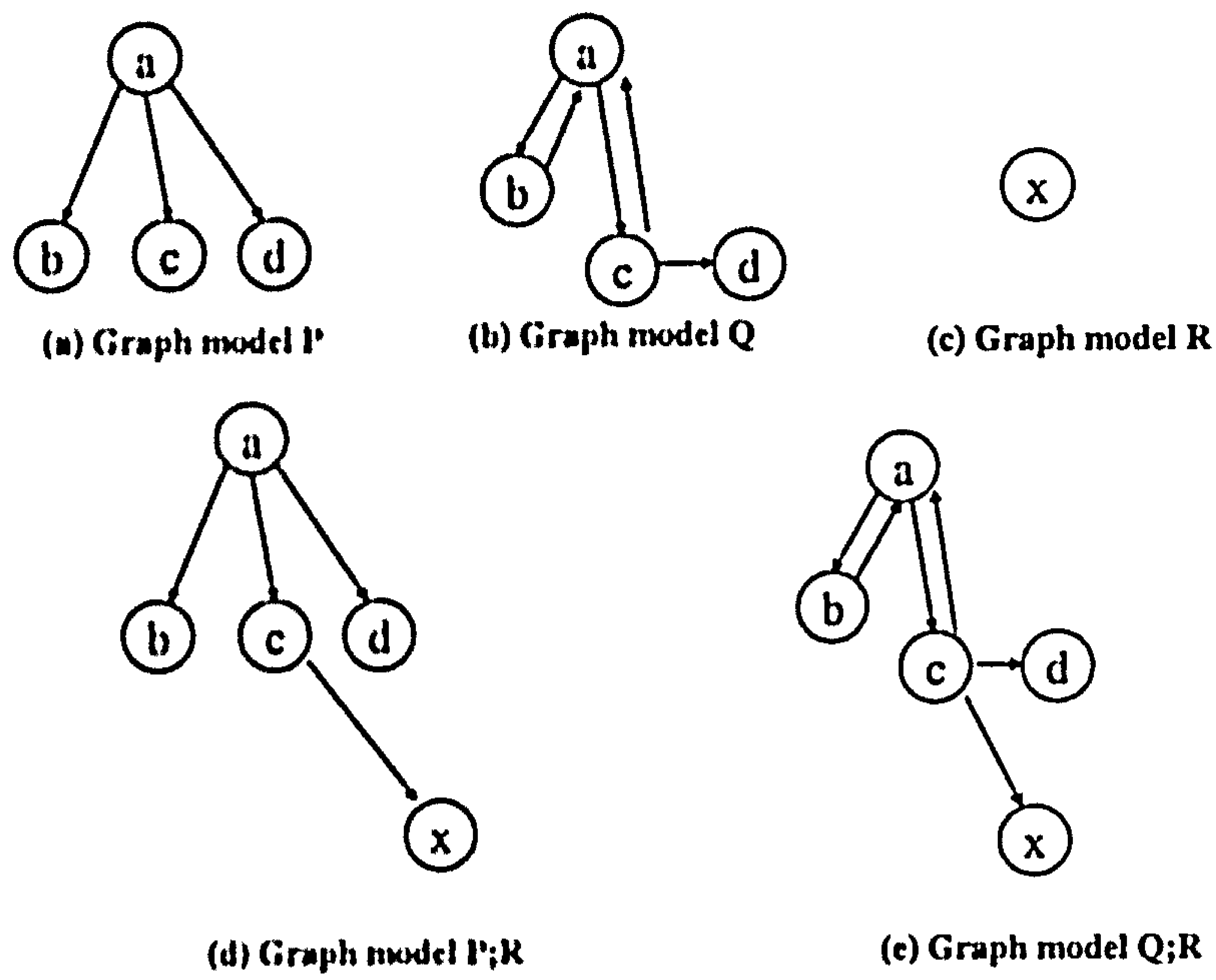

Figure 4.15 Example of Axiom 6 for WSC_6

(7) WSC_6 satisfies Axiom 7. Figure 4.16 is a counterexample, where Q is a permutation of P, but $WSC_6(P) = 2/3$, $WSC_6(Q) = 4/3$, i.e. $WSC_6(P) \neq WSC_6(Q)$.

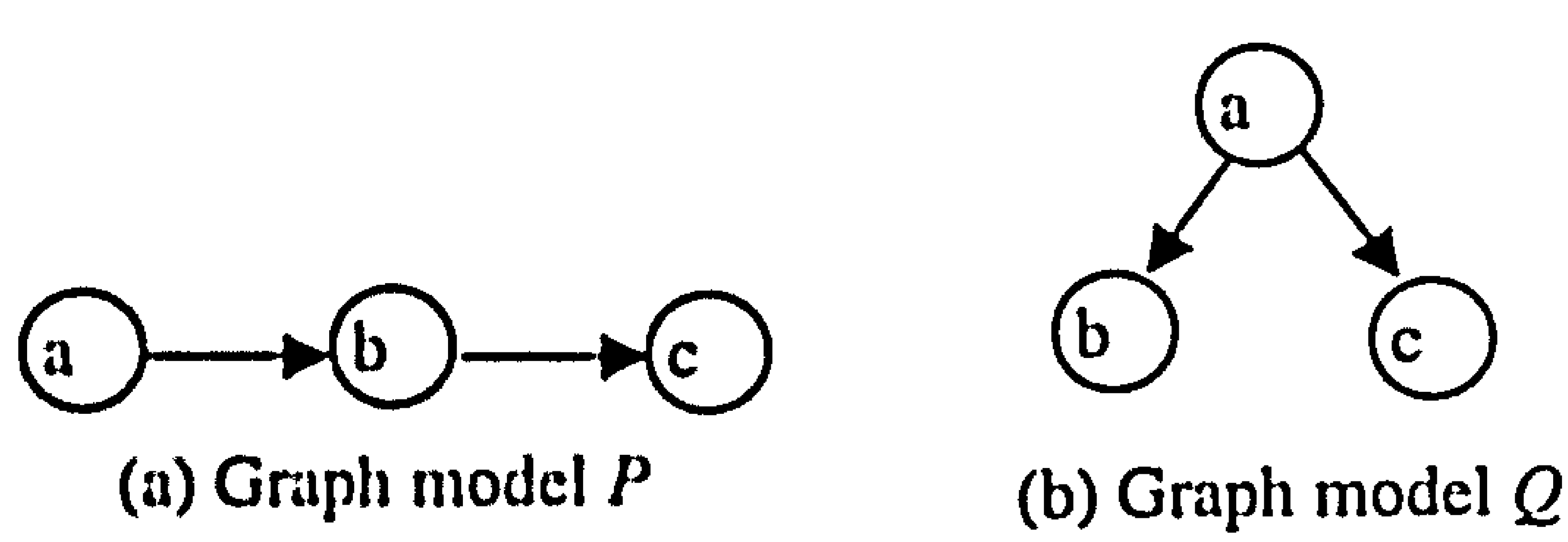

Figure 4.16 Example of Axiom 7 for WSC_6

(8) By Definition 4, WSC_6 satisfies Axiom 8.

(9) WSC_6 satisfies Axiom 9. To prove this, consider the graphs given in Figure 4.17. $WSC_6(P) = \frac{4}{3}$; $WSC_6(Q) = 0$; $WSC_6(P;Q) = \frac{9}{4}$. Therefore, $WSC_6(P;Q) > WSC_6(P) + WSC_6(Q)$.

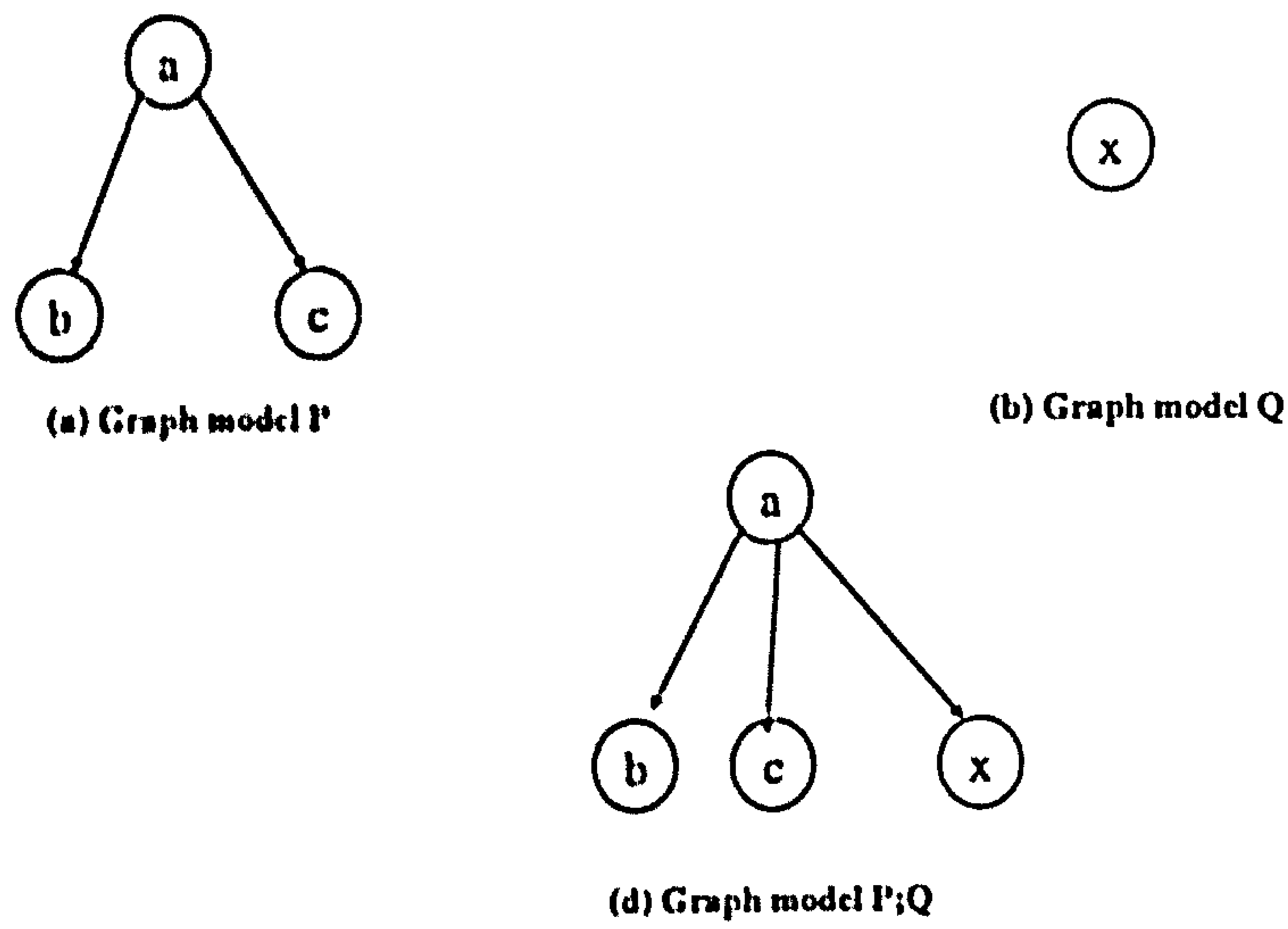

Figure 4.17 Example of Axiom 9 for WSC_6

Table 4.1 summaries the above results.

Table 4.1 Assessment of Metrics against adapted Weyuker's axioms

Metrics / Axioms	WSC_1	WSC_2	WSC_3	WSC_4	WSC_5	WSC_6
Axiom 1	Yes	Yes	Yes	Yes	Yes	Yes
Axiom 2	Yes	No	No	No	Yes	No
Axiom 3	Yes	Yes	Yes	Yes	Yes	Yes
Axiom 4	Yes	Yes	Yes	Yes	Yes	Yes
Axiom 5	Yes	No	Yes	No	Yes	No
Axiom 6	No	Yes	Yes	Yes	Yes	Yes
Axiom 7	No	No	Yes	Yes	Yes	Yes
Axiom 8	Yes	Yes	Yes	Yes	Yes	Yes
Axiom 9	Yes	No	No	No	Yes	Yes

From Table 4.1, it can be seen that WSC_5 complies with the adapted Weyuker's axiom system completely, other metrics comply with most of the axioms, and they can still be successful candidates.

It is important to note that Weyuker's axioms are not sufficient to evaluate the metrics. Thus empirical studies need to be conducted. In this study, a user-centered questionnaire technique is used.

4.5 Empirical evaluations of the metrics

To further evaluate the metrics, four websites of the same nature were chosen as the subject of evaluation. Experiments with human users' access to the websites were designed, and carried out. The experimental results are compared with the data calculated using the metrics. This section reports the empirical evaluation of the metrics.

4.5.1 The subjects of the empirical study

The websites used in the empirical study are all university portals. The universities are geographically located in the same city in England. In the sequel, they are referred to as U1, U2, U3 and U4, respectively.

The empirical study was first carried out in 2003. The same experiment was repeated again in 2004. Three groups of students were selected at random to participate in the empirical study. The numbers of students that participated in the experiments are shown in Table 4.2.

Table 4.2 Number of Students Participated in the Empirical Study

Year	MSc students	BSc students	HND students
2003	0	88	34
2004	22	98	43

All the students participating in the empirical study are from computing fields and fluent in the use of the Web to find information. They all use university websites to obtain their daily study information, etc.

4.5.2 Calculation of metrics

A simple software tool was developed to calculate the navigability of the chosen websites using the metrics defined in Section 4.4. The software tool consists of three parts:

1. **Site Download Agent.** This part downloads the whole site. A simple Perl script from CPAN (http://www.perl.org/CPAN) is used to generate an HTML site map from a given URL. It does this by traversing the site, getting the home page, extracting links from it, getting all the pages linked, and so on. The first level indented list item is the home page; the next level are all the pages linked from the home page. The next level are all the pages linked from each of these pages, and so on. If a page is linked from more than one page, it is shown in the "highest" place in the tree it is linked from, which guarantees that no file will be downloaded twice.

2. **Filter.** This part filters the multimedia files (audio, video, animation...), which are regarded as dead nodes, and only keeps the appropriate text files (html, shtml, htm, php, php3, asp) to process, since only the hypertext files contribute to the site structure.

3. **Metrics Calculation and report.** This part calculates the metrics according to the equations and stores the results in the database.

All the tools are written as Perl scripts, see Appendix C. The results are given in Table 4.3 below.

Table 4.3 Navigability according to the metrics (Tested on 22 March 2003)

Site	Pages	WSC_1	WSC_2	WSC_3	WSC_4	WSC_5	WSC_6
U1	5842	107493	18.4	103403	17.7	6215888	1064
U2	6824	128974	18.9	124197	18.2	8257040	1210
U3	3685	85861	23.3	82913	22.5	4543605	1233
U4	4608	131789	28.6	128563	27.9	8451072	1834

4.5.3 Questionnaire

Based on the literature on website design, especially the guidelines and heuristics, an initial list of attributes of navigability was created. In addition, eight site-frequent users were interviewed. Based on the interview outcomes about the most concerned attributes on navigability from the end-users point of view, a questionnaire was designed. The questionnaire incorporates the list of attributes taken from the usability heuristics regarding navigability.

The questionnaire consists of eight sections. Each section is concerned with one particular aspect of navigability. It contains a number of tests for the participants to perform and to give an objective or subjective rating of the attribute, as well as a comment.

Given the differences in the nature of the tests and ranking contained in the sections, the results of the sections could have different formats. For example, the answer to Section 1 will be the number of clicks, whereas the answer to Section 5 is 'yes' or 'no'. Some answers are subjective preferences. To enable statistic data processing, a Likert scale, rating from 1 for worst to 5 for best, was used to normalise the results. The following summarizes the tests and the meanings of the ratings in each section of the questionnaire.

Section 1 is concerned with the minimal paths to find four pieces of information on the targeted website. The answers are not difficult to find on the websites and that questions designed for each website have the same difficulties, which means that the answers can be found at the same depth level of the site. To test the structure of the site, participants are only allowed to follow hyperlinks to find the answer, rather than use the search facilities. The number of clicks that the participant made to find the information is recorded. The average number of clicks for all participants is calculated. Cochrane's 'three-click' rule is followed, rating 5 for 3 clicks, 4 for 5 clicks, 3 for 7 clicks, 2 for 9 clicks, and 1 for more than 9 clicks or giving-up or finding the wrong information.

Section 2 is concerned with availability of alternative paths to find a piece of information. It contains a test that asks the participant to find a specific piece of information using different paths or navigational facilities. In this section, participants feel free to use any navigational paths and/or navigational facilities, such as a search engine, to find the answer.

Section 3 is concerned with the navigational structure taxonomy. It asks the participant to check layout consistency, which is an important factor for the user to know 'where am I', and to give subjective ratings of the website content organization in terms of the link taxonomy suitability. The results are charted on a scale from 1 for worst to 5 for best.

Section 4 is concerned with link visibility. It asks the participant to check the hyperlinks according to Nielson's heuristic, such as link layout (if embedded links), cursor change, and colour change before and after the link is clicked, and to give a subjective rating on a scale from 1 for worst to 5 for best.

Section 5 is concerned with the search facility. It asks for the participant's subjective rating of the availability and effectiveness of search facilities, such as search engine, menu, site map, and navigational bars, etc. The rating is on a scale from 1 for worst to 5 for best.

Section 6 is concerned with the labels associated with the links. It asks the participants to check the predictability of links in terms of clear, explanatory and suggestive labeling, and give a subjective rating on the scale from 1 for worst to 5 for best.

Section 7 is concerned with navigational errors. The participant is asked to rank the website according to the errors that he/she encountered during the task in Section 1. Such errors could be broken links or other unexpected navigational behaviors. The ratings are on the scale from 1 for worst to 5 for best.

Section 8 is concerned with the availability of supportive mechanisms to aid disabled people, such as large font for weak-sighted people, to navigate the site. The participant is asked to give a subjective rating on a scale from 1 for worst and 5 for best.

The first interesting finding is that different groups gave the same results statistically. Therefore, the groups of participants will not be distinguished. The results of all participants are shown in Table 4.4.

Table 4.4 Questionnaire results of website navigability (March 2003)

Section	U1	U2	U3	U4
1. Minimal path	4.3	3.0	3.0	2.1
2. Alternative path	4.2	3.5	2.9	3.0
3. Link taxonomy suitability	4.2	4.8	4.2	4.1
4. Link visibility	4.4	4.2	4.3	3.2
5. Search facility effectiveness	4.5	3.3	4.2	3.3
6. Navigational predictability	3.2	2.8	3.1	2.6
7. Navigational errors	4.8	4.6	4.2	3.6
8. Supportive mechanism	1.4	1.3	1.3	1.1
Total mark	31.0	27.5	27.2	23.0

4.5.4 Discussion

Table 4.4 shows that, from the participants' subjective point of view on navigability, U1 website gained the highest mark, and U4 the lowest. This is consistent with all metrics defined in Section 4.3.

Comparing Table 4.3 and Table 4.4 using correlations analysis, empirical results showed a strong negative correlation between website structural complexity and Web navigability, see Table 4.5.

Table 4.5 Correlations between data in Table 4.3 and 4.4

Results / Metric vs. Questionnaire	CORREL
WSC_1 vs. Questionnaire Results	-0.452
WSC_2 vs. Questionnaire Results	-0.908
WSC_3 vs. Questionnaire Results	-0.479
WSC_4 vs. Questionnaire Results	-0.909
WSC_5 vs. Questionnaire Results	-0.479
WSC_6 vs. Questionnaire Results	-0.945

The empirical study was repeated after six months and four websites were re-tested using different user groups. The results of metrics calculation are shown in Table 4.6.

Table 4.6 Navigability according to the metrics (Tested on 26 March 2004)

Site	Pages	WSC_1	WSC_2	WSC_3	WSC_4	WSC_5	WSC_6
U1	6742	98965	14.7	94183	14.0	6113481	906.8
U2	7142	160895	22.5	156030	21.8	7850597	1099
U3	4117	100259	24.4	96839	23.5	3938167	956.6
U4	5155	137619	26.7	134257	26.0	6704895	1300.7

Questionnaire results are in Table 4.7, and the correlations analysis results are in Table 4.8.

Table 4.7 Questionnaire results of website navigability (March 2004)

Section	U1	U2	U3	U4
1. Minimal path	4.7	3.5	3.9	3
2. Alternative path	3.8	4.2	3.9	3.1
3. Link taxonomy suitability	4.6	4.5	4.2	4.1
4. Link visibility	4.8	4.8	4.7	4.3
5. Search facility effectiveness	3.8	3.3	3	2.8
6. Navigational predictability	3.7	4	3.3	3.5
7. Navigational errors	4.8	4.7	4.4	4.5
8. Supportive mechanism	1.1	1.5	1.2	1.1
Total mark	31.3	30.5	28.6	26.4

Table 4.8 Correlations between data in Table 4.6 and 4.7

Results / Metric vs. Questionnaire	CORREL
WSC_1 vs. Questionnaire Results	-0.144
WSC_2 vs. Questionnaire Results	-0.858
WSC_3 vs. Questionnaire Results	-0.170
WSC_4 vs. Questionnaire Results	-0.860
WSC_5 vs. Questionnaire Results	0.177
WSC_6 vs. Questionnaire Results	-0.764

Table 4.5 and 4.8 show that relative structural complexity can be used as indirect metrics for navigability. Navigability will be reduced with the increase of website structural complexity.

The study paid great attention to link number and the distribution, which play significant roles in website structure. It ignores some properties of links, such as link taxonomy. In Figure 4.18, for example, the left column showed Dan's e-commerce site with hyperlinks arranged alphabetically, the right column with topics. Although the number of links is the same, the navigability is apparently different. The proposed metrics cannot distinguish such differences. Therefore, the results of the empirical study should be understood under the assumption that the websites studied are well designed in terms of the taxonomy, etc.

Dan's Clothing Store

Checkout
Closeout on pink socks
Email us
July specials
Kid's clothes
Men' clothes
Open an account
Sale on rain wear
Special sizes
Store hours
Store locations
Your account status
Women's clothes

Dan's Clothing Store

Women's clothes
Men' clothes
Kid's clothes
Special sizes

July specials
Sale on rain wear
Closeout on pink socks

Open an account
Your account status

Checkout

Store hours
Store locations
Email us

Figure 4.18 Link taxonomy vs. navigability

4. 6 Summary

This chapter focuses on navigability measurement. Navigability is defined as the ability provided by the sites to aid the users to find information needed by moving around the sites easily. Based on the assumption that a strong correlationship between the website structural complexity and its navigability, this chapter defined and investigated six metrics of website structural complexity (WSC).

The chapter investigated and redefined some properties of Weyuker's axiom system and adapted it to the WIS. The candidate WSC metrics were assessed against the adapted Weyuker's axioms. In addition, user-centred surveys were conducted to evaluate if the WSC metrics were proper measures for navigability. Results of the empirical studies proved the hypothesis that website structural complexity has a strong negative correlation to its navigability. They also showed that the metrics proposed were applicable and some WSC metrics were proper navigability metrics of university websites.

Chapter 5 Measuring Website Timeliness

5.1 Introduction

Timeliness is a quality attribute for websites (Zhu et al, 2002; Zhang, et al, 2002; 2001; Brajnik, 2001). Together with other attributes, such as information completeness, accuracy, correctness, and so forth, timeliness can be used to assess the information quality of a website. Usually, users judge the timeliness of some kinds of sites by matching the sites with the real world. Compared with the printed document, it is easy to make changes to a web page. Users thus always expect websites to provide timely information. If a site has not been updated for a long time, it will not only give a very poor impression of the organisation, but also make the users suspect the quality of information provided in the site.

Nielsen (2003)'s study of Web usability showed that timeliness of information has become a big issue today. The following are among the examples of poor timeliness.

- The latest news in the scrolling headlines was one year old.
- We visited an e-commerce website and filled in a form, then we discovered that the closing date was two months ago.
- One website used blinked text to tell people to be aware of the millennium bug. That page was obviously created last century.

As Nielsen pointed out, such outdated information is obviously useless. Therefore, more and more Web designers have paid attention to this issue (Collins Memorial Library, 2004; CSUS, 2003; Powell, 2002; Lynch and Horton, 1999; Borges et al, 1996). A variety of strategies have been implemented to show the timeliness of websites, which include:

- Put date and/or time indicators on the page.
- Use animation associated with dynamic content to show its timeliness.

- Tie content to current real-world events, such as latest movies, events, political elections, holidays, and so forth.
- Highlight timely content on your site's home page.

However, how to assess website timeliness still remains an open problem. In (CSUS, 2003), three criteria were suggested to evaluate website timeliness:

1) Is there an indication of when the information was published?
2) Is the information regularly revised or updated?
3) Is the information still valid for your topic?

In (Collins Memory Library, 2004), the following five criteria were proposed to assist in evaluating website timeliness.

1) When was the information created or last updated?
2) Is the source appropriate for your needs?
3) How current are the links, statistical data, illustrations, etc.?
4) Does the information appear to be valid and well researched, or is it questionable and unsupported by evidence?
5) What is its relation to other works on the subject?

Such criteria also provide useful design guidelines, or heuristics, for website designers. However, it is rather difficult to validate the correctness and completeness of such heuristics. Furthermore, the application of the heuristics is also a difficult task. The most common way of evaluation of a website's timeliness is by user test, which is, unfortunately, not always practical. As pointed out in (Tes, 2002), it is not an easy task to organise and complete such a test satisfactorily, which is time-consuming and costly especially for repetition tests.

Timeliness measures can be used to evaluate a site's timeliness quantitatively and thus provide insight into the issue. It is therefore important and necessary to develop such measures.

The research on website timeliness measurement is still in its infancy. In this study, the development of timeliness measures will follow a rigorous approach discussed in Chapter 4. Thus, the research work was divided into the following steps.

- Define timeliness of WIS
- Derive and define timelines measures
- Implement software tools to calculate the measures
- Validate the candidate measures using empirical studies.

The rest of the chapter is organised as follows. Section 5.2 defines the concept of website timeliness. Section 5.3 derives and defines three direct and indirect measures according to the definition. Section 5.4 reports the empirical studies that validate the candidate measures and discusses the results. Section 5.5 summarises the chapter.

5.2 Concept of website timeliness

As mentioned earlier, users usually judge the timeliness of a website by comparing the site with the real world. They believe that good quality information should be updated in a timely manner to reflect the changes in the real world. Thus, the WIS timeliness is defined as follows.

> A website's timeliness is its ability to provide and process information in a timely manner, i.e., to create, update and present information within a tolerable time delay in order to keep the information consistent with the real world.

For example, an event occurred at 08:00, site A published a report on the event at 08:30, and site B at 09:30. If users visited both sites around 09:00, they would read the news at site A, but not site B. If the same situation were observed often by the users, site A would be believed to provide timely information and gain users' loyalty due to its timeliness. It seems too early to draw the conclusion that site A has better quality than site B. For example, if the events was partially reported by site A, but completed reported by site B, site B should be awarded for its information relevance and completeness. However, such quality attributes are beyond the scope of this chapter.

5.3 Measures for measuring timeliness

In this study three timeliness measures are proposed. They are formally defined as follows.

Measurement 1: Mean Time Delay To Publish

According to the definition of the concept of timeliness, the direct measure of timeliness is to measure the time delay between the time when the information is published and the time when the event occurs in the real world. Let $A = \{\alpha = |\alpha_n|\}_{n=1,\dots,K}$ be a set of events. The Mean Time Delay To Publish (MTDTP) with regards to the set A of events can be formally defined as follows.

$$MTDTP_A = \sum_{i=1}^{K} (T_{publish}(\alpha_i) - T_{occur}(\alpha_i)) / K \qquad \textbf{Eq 5.1}$$

where $T_{publish}(\alpha_i)$ is the time when an event α_i is published online, and $T_{occur}(\alpha_i)$ records the time it occurs.

This measure provides a direct measure of timeliness, but it has the following limitations.

1. The measure relies on the availability of T_{occur}. In the current practices of web site development, T_{occur} can only be found manually in the websites or from other media types. For big events, the news content usually includes the precise time when the event happened. The accuracy of the measure is dependent on the accuracy of the T_{occur}. If T_{occur} cannot be found, or only a rough idea of the time is available, such as around 5 pm, the measure is not applicable.

2. It is usually time-consuming to find the T_{occur} and $T_{publish}$. As mentioned above, most websites put time indicators on the page. However, without an automatic mechanism, it is obvious that $T_{publish}$ involves much manual reading. Even with the aid of a software tool in this study, it was a rather tedious and time consuming task to complete.

3. It is rather arguable to decide the set of events to test a website's timeliness. Different websites may have different tastes. The same event could be regarded as a 'big' news for one site, but completely ignored by another site. The set of events chosen for testing may significantly affect the test results.

Measurement 2: Site Evolution Speed

Site Evolution Speed (SES) calculates the number of web pages that are changed over a period of time. The empirical study found that the changes of a website were often only reflected by the number of pages added. It is rare that pages were deleted or changed except for the homepage.

For a fast-changing website, it is always ready to publish a piece of news. Usually a headline appears in the homepage, and the detailed information is added to the site as a new page with a link from the headline. Each time a new page is added to the website, a change to the website is made. Let 2 probes of a website w be made at time moments t_0 and t_1, *AddedPages*(w) is the number of new pages added to website w between time t_0 and t_1 based on the state of the website obtained by the probes. The Site Evolution Speed measure can be formally defined as follows.

$$SES = \frac{AddedPages(W)}{t_1 - t_0} \qquad \textbf{Eq 5.2}$$

SES can be automatically calculated. However SES is accurate only if the website does not delete web pages, merge or split web pages.

Measurement 3: Homepage Update Frequency

Website update frequency measures a website's timeliness through the number of times that a website updates its information. The measure can be formally defined as follows.

$$Frq_{Update}(Website) = \frac{Number\ of\ Updates}{Length\ of\ Time\ Period} \qquad \textbf{Eq 5.3}$$

Although update frequency is not a direct measure of time delay, it is closely related to time delay. Two types of websites can be identified according to their strategies for updating information. The first type of websites updates their information periodically, such as websites of daily newspapers. Another type of websites updates their information as soon as the information is available and ready to be published on the web. As shown in the Figure 5.1 below, for websites that update their information periodically, one would expect that the time delay of a website with high update frequency should be less than those that have a lower frequency of update.

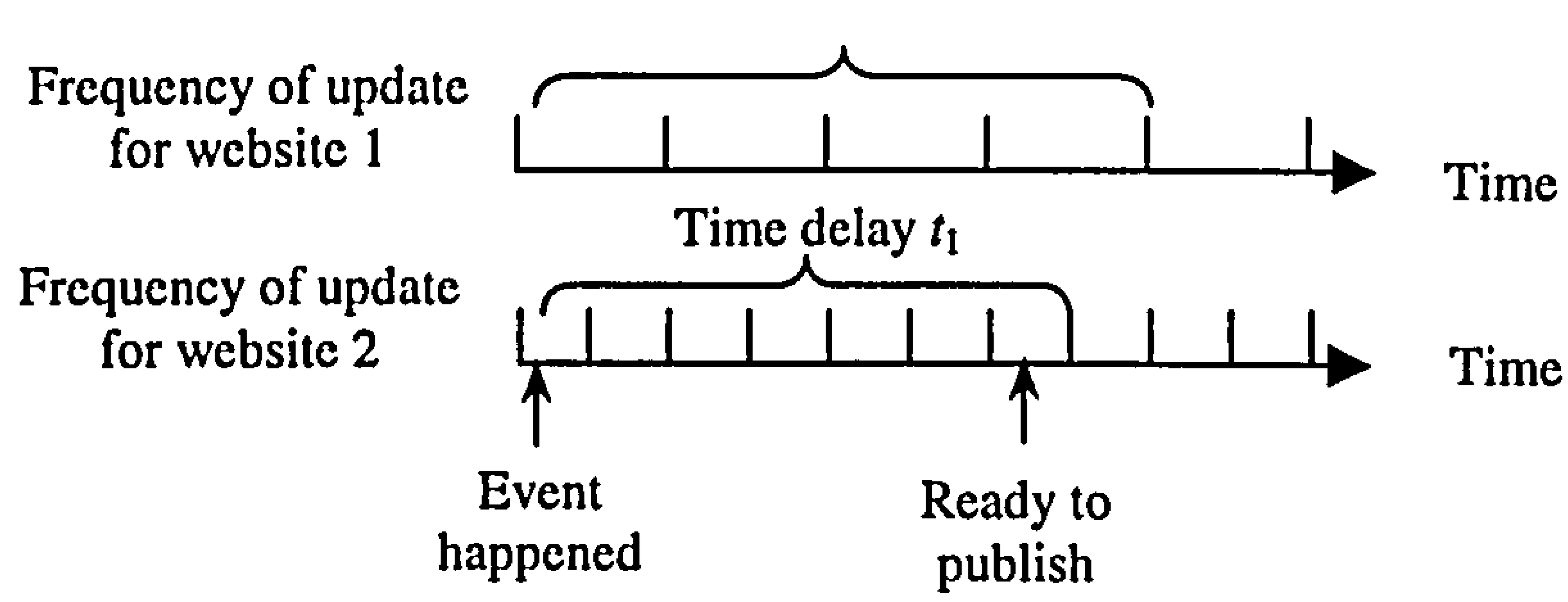

Figure 5.1 Effect of update frequency on time delay

Assume that the same time length was taken to prepare the publication of information on a website, the more frequent one website updates its information, the less one can expect the time delay for that information is to be published.

For websites that do not update their information periodically, but update the information whenever it is available and ready, the average update frequency also indicates the time delay, as shown in Figure 5.2, where the real world events are indicated by capital letters and the publication of the event is indicated by lower case letters. The time delay for an event x is denoted by the symbol δ_x. The figure shows that website 1 has a higher average update frequency than website 2. Its total time delay, i.e. the summation of δ_x for $x = a, b, c$ and d, is smaller than the total delay on website 2.

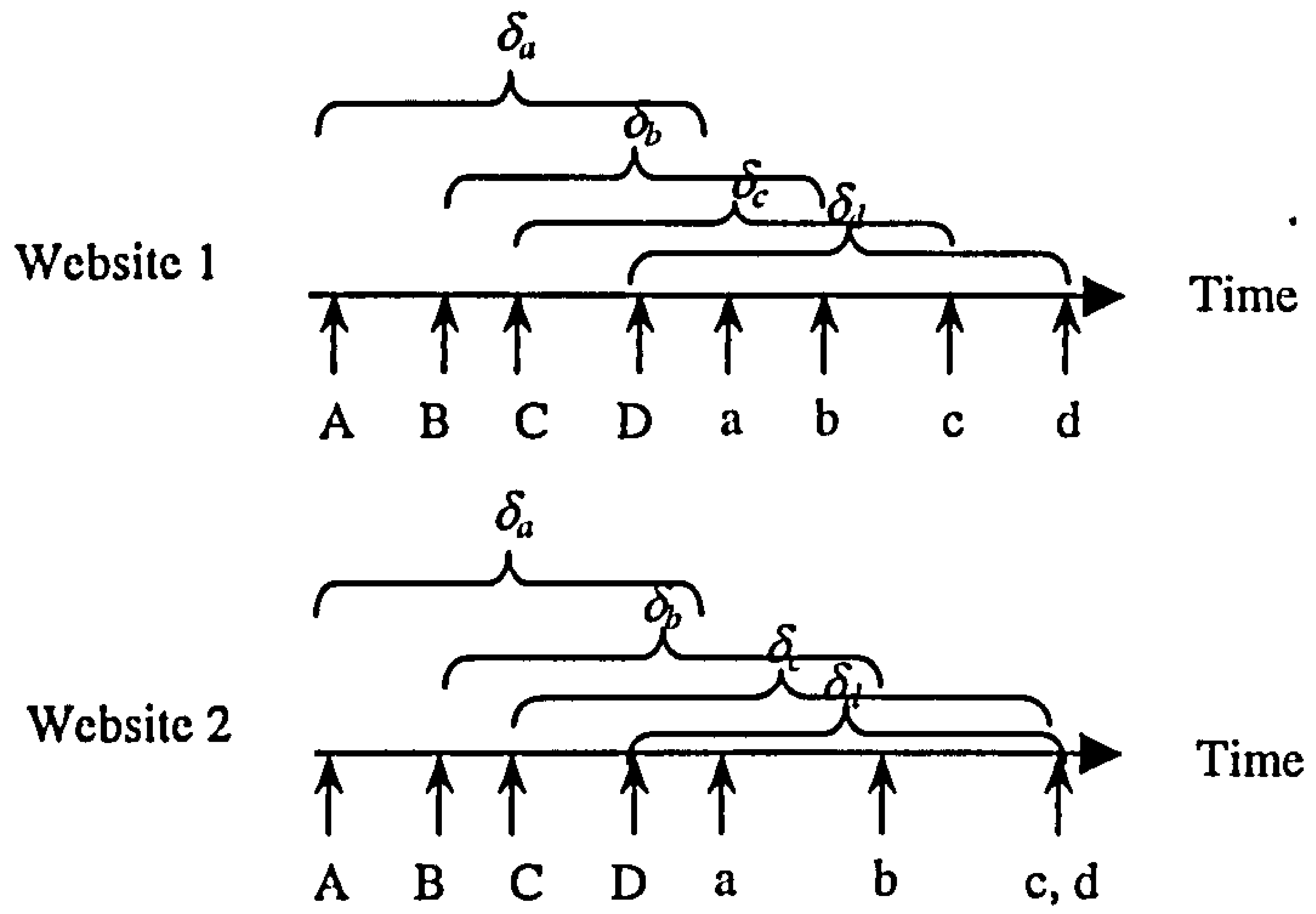

Figure 5.2 Update frequency indicates the time delay

However, the update frequency of a website cannot be measured accurately because web publication is passive in the sense that the information on a website can only be obtained when the user requests the information. The only practical approach to measuring the update frequency is to make a number of requests for the information in certain period of time and to find out how many times the information has changed. Therefore, a measure for update frequency is the probability of change defined as follows.

$$Pr_{Change}(Website) = \frac{Number\ of\ different\ versions}{Total\ number\ of\ requests} \qquad \textbf{Eq 5.4}$$

Assume that the requests for information from a website is made regularly with a fixed gap of time between two consecutive requests, the update frequency can be calculated approximately from the probability of change. It is easy to see that when the sampling frequency is high enough, the detected update frequency should be close to the real average update frequency. That is, if $Frq_{Update}(Website) < Frq_{Sampling}$, formally

$$Frq_{Update}(Website) \approx Pr_{Change}(Website) \times Frq_{Sampling} \qquad \text{Eq 5.5}$$

where $Frq_{Sampling}$ is the sampling frequency, which is defined as the number of samples of the website obtained in a given period of time.

The requirement to request information from a website at a high sampling frequency implies that detecting that the information obtained is different from the previous one must be computed very efficiently. To meet this requirement, MD5 algorithm (Rivest, 1992) is adopted to generate a fingerprint for each web page. Every time an html file is obtained from a website, its fingerprint is generated and compared with the last fingerprint. The MD5 algorithm can create a unique 'fingerprint' for a file. The MD5 algorithm therefore can be used to identify whether the page requested has changed or not. Because the fingerprint of a file is significantly shorter than the original file, and the MD5 algorithm is very efficient, comparing fingerprints is much more efficient than comparing the original files.

Although the use of the MD5 algorithm and fingerprints of files significantly improved the efficiency of detecting changes in a website, detecting changes by comparing all the pages of a website is still not practical. Therefore, only the changes in the home page will be detected because major changes in the contents of a website are almost always reflected in the homepage. For example, the headline news pieces are always listed in the home pages of daily newspaper web sites.

Hence, the Homepage Update Frequency (HUF) measure is defined formally as follows:

$$HUF = \frac{C_{pr}}{N_{pr}} \qquad \text{Eq 5.6}$$

where N_{pr} is the number of probes made to a website in a period, C_{pr} is the number of probes that detected changes to the home page.

In theory, if a website always changes at a frequency of less than once a minute and every time only one page is added to the website and pages are never deleted or merged/split, then, both SES and HUF will detect all changes to a website if they are applied on the same period of time.

$$HUF = SES/SF_r \qquad \textbf{Eq 5.7}$$

where SF_r is the sampling frequency of HUF.

In the practical uses of the HUF, an appropriate frequency of sampling must be carefully set so that the home pages can be downloaded between two probes. Therefore, efficient implementation of the sampling tool is the key issue for the usability of the measures, which has been discussed in (Zhang et al, 2002). The uses of SES do not heavily rely on the efficient implementation of the measure. A question is how well the assumptions made in the uses of SES and HUF measures match reality.

5.4 Results of empirical studies

To evaluate the feasibility of the measurement method and to validate the measures as well, experiments were conducted.

5.4.1 Empirical study 1: Feasibility study of comparing timeliness of UK news websites using HUF

The objective of this empirical study was to evaluate the practical feasibility of using the HUF measure to compare the timeliness of web-based news media. Therefore, this study adopted the regular and fast updating websites that have a high reputation in the empirical study. In addition, different websites have their own interests and objectives. It is hardly meaningful to compare their timeliness. Therefore, in the study, five well-known UK newspaper's websites were chosen to test the measure.

The websites used for testing were:

- The Times (www.times.co.uk)
- The Guardian (www.guardian.co.uk)
- The Daily Telegraphy (www.dailytelegraph.com)
- The Independent (www.independent.co.uk)
- The Sun (www.thesun.co.uk)

A software tool written in Perl was developed to measure websites' HUF continuously for 24 hours. The sampling frequency was once a minute. A total of 1440 requests for each site were made within 24 hours. If a site responded slowly or was locked while updating, several tries had been made until the homepage was obtained successfully. Because some requests could not complete within a minute, the actual number of requests made to a website was a bit less than 1440.

The results of HUF of the websites are given in Table 5.1.

Table 5.1 Timeliness of five news websites

Results / Website	Total requests	Changed times	Unchanged times	Failed	HUF
Times	1411	39	1364	8	2.8
Sun	1436	36	1400	0	2.5
Independent	1438	8	1426	4	0.6
Guardian	1415	79	1311	25	5.6
Daily Telegraph	1420	64	1351	5	4.5

Although the actual total number of requests was a bit less than 1440, which was expected, the average of percentages of successful sampling is very high, as shown in Table 5.2.

Table 5.2. Feasibility of using HUF to measure fast changing websites

Website \ Results	Actual number of requests	Number of Failure	% of Failure	% of missed requests
Times	1411	8	0.57	2.01
Sun	1436	0	0.00	0.28
Independent	1438	4	0.28	0.14
Guardian	1415	25	1.77	1.74
Daily Telegraph	1420	5	0.35	1.39
Average	**1424**	**8.4**	**0.59**	**1.11**

The results showed that using MD5 to measure homepage updates was efficient, and thus the measurement method is feasible and applicable.

The empirical study not only showed that the frequencies of homepage updates vary from time to time, but also there is a common pattern of the distribution of change frequencies for the same type of websites. Figure 5.3 illustrates the pattern of HUF of the sites over a 24 hour period of time.

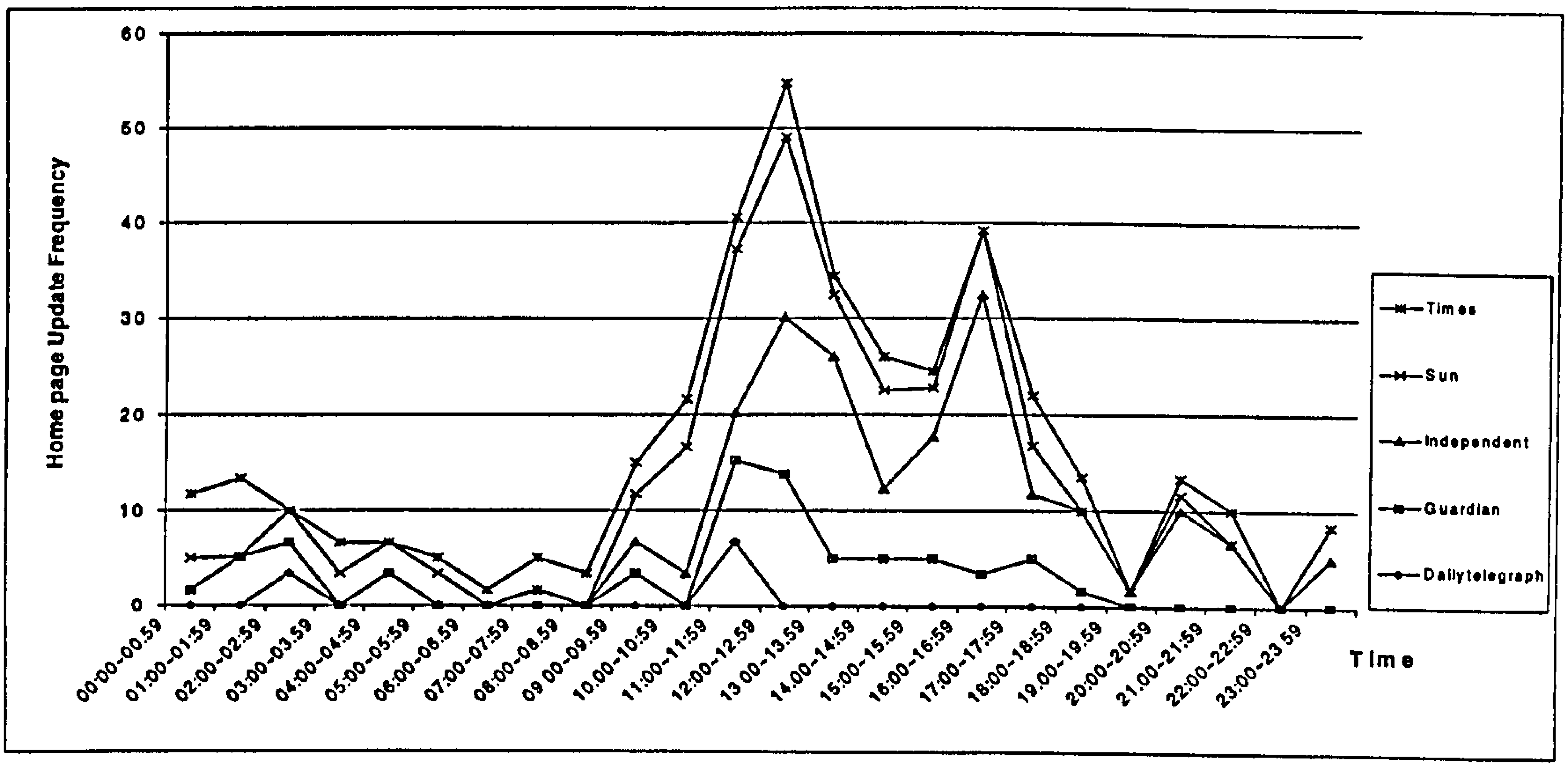

Figure 5.3 Pattern of HUF of news sites

The five news sites showed a quite similar change pattern. Two change peaks around 12:00 and 17:00 were observed in Figure 5.3. At the rest of the time, most homepages kept changing at a lower frequency. The explanation of this phenomenon could be that a website would keep changing when new information was provided. When the information is received and edited, it would be put online as soon as possible. Noon (12:00) is perhaps the time when the preparation work in the morning is finished and ready to publish online. Five o'clock in the afternoon is perhaps the time when the preparation work for the afternoon would finish and be ready to publish on the web.

It evoked an idea to find out whether all different types of websites show a similar change pattern. Hence, a website to show share prices in a stock market was studied. The URL of the site was: http://www.wsrn.com/apps/ISDEX. The site was chosen as it was also a fast-changing site and its timeliness is crucial. Figure 5.4 showed the HUF pattern of the site. It was found that its change frequency distribution shape was near to a rectangle. Between 14:00 and 21:00 GMT, the homepage updated at least once every minute. As this website is in America, that time should be the working hours there. The rest of the time, when the stock market was closed, the homepage seldom changed.

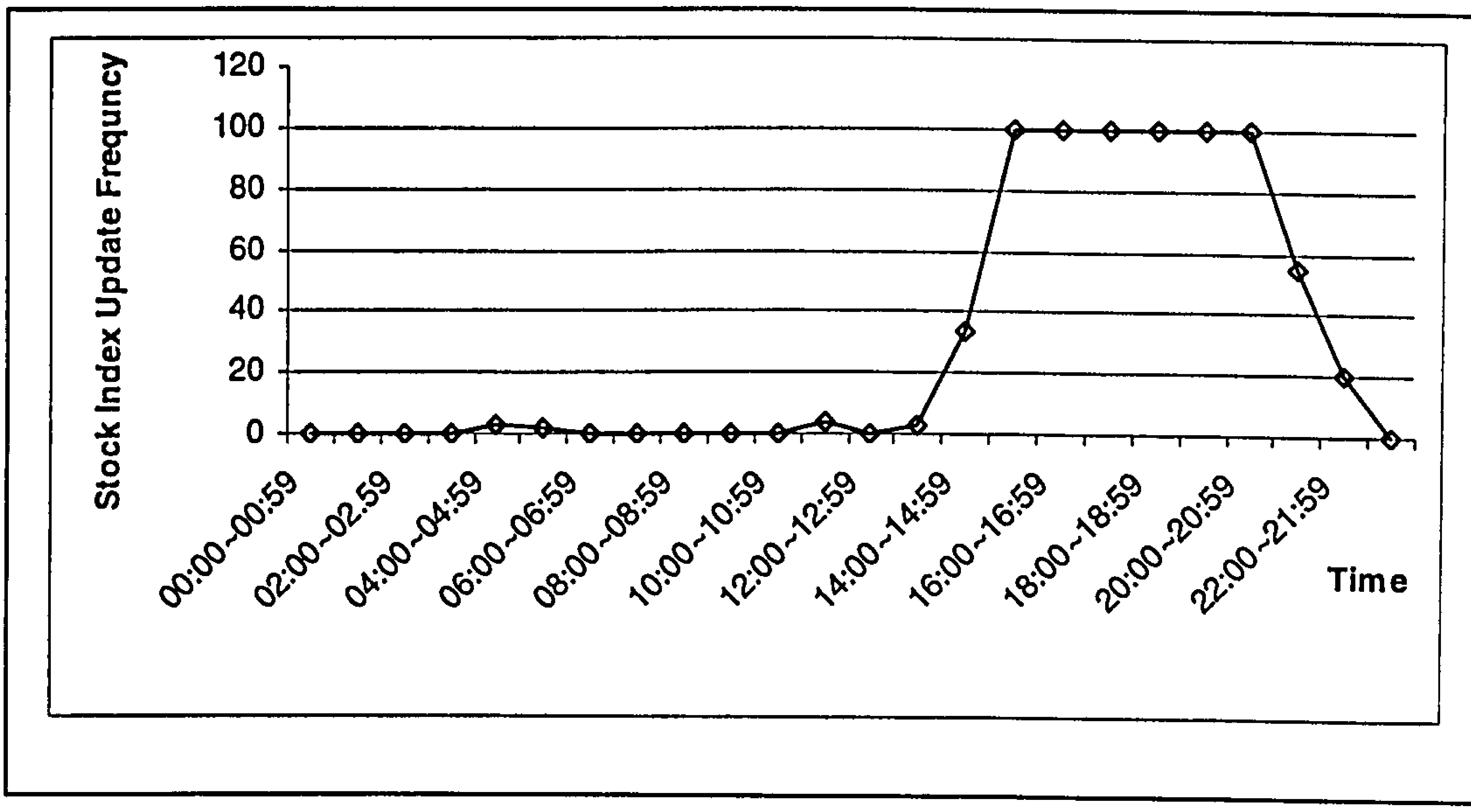

Figure 5.4 Pattern of HUF of a website of share prices

The result implied that a hypothesis for further research is that the pattern of HUF could be used to distinguish different types of websites.

5.4.2 Empirical study 2: Validation of using the HUF measure to compare website timeliness

The objective of the second empirical study is to validate the uses of the HUF measure in comparing website's timeliness. The same group of websites were chosen as in the first empirical study. The experiment not only collected the measurements of the HUF of the websites for 24 hours but also selected 3 'big' news events chosen to compare the times that each website reports it. These events were chosen from the top 3 headlines in reported by BBC. Only the most important news items were chosen such as "Bush and Putin signed Nuclear Treat.", "Saddam Hussein 'Caught Like a Rat'", and "Israel Bomb Blast, 16 killed". This is because these events are reported by all of the websites selected in the empirical study,

The experimental process consists of four steps. A software tool does two parts and the other parts involve manual data collection through reading the related web pages.

1. Detect change then download. A software robot tests once per minute whether the homepage of a website has changed, and downloads it if so. The downloaded files with the time stamp are then stored into a database.
2. Selection of news events. For each experiment, 3 events were manually selected that the websites should report. As discussed above, these events were chosen from BBC news report headlines.
3. Search for website's reports in the database. A keyword search software robot was programd to search the database with a given keyword then reports the location of the target files. For example, a keyword of 'Hughes' can be used for the news of 'Hughes resigns over visa scam row'.
4. Collection of data of reporting times. The related pages returned by the search robot were read manually to confirm if the page really reports the news. The earliest report from each website is thus identified. The time stamp of the file was used as the time that the website reports the event.

For example, the data collected in an experiment on one day are shown in Table 5.3, where Times' site was used as the reference. In Table 5.3, "+" means the website published later than the Times website and "-" published earlier.

Table 5.3 Result of validation experiment (minute)

Site / Items	Times	Sun	Independent	Guardian	Daily Telegraph
1	0	-3	+11	-5	-6
2	0	+1	-1	-40	-33
3	0	+2	+1	+2	-1
Total	0	0	+11	-43	-40

Table 5.4 shows the average latency over 3 events and the HUF measurements on each website. Using Pearson's test to compare the average latencies and HUF measurements, it can be found that there is a strong correlation, r = -0.95 between them. This means that HUF is a valid measure for timeliness.

Table 5.4 Comparison of average latencies and HUF measurements

Results / Website	Average latency	HUF
Times	0.00	2.8
Sun	0.00	2.5
Independent	3.67	0.6
Guardian	-14.33	5.6
Daily Telegraph	-13.33	4.5

The experiment was repeated for 3 more times on 31st March 2004 (r = -0.86); 5th (r = -0.94), 6th (r = -0.88) April 2004. The results are shown in Table 5.5.

Table 5.5 Correlation coefficients between average latencies and HUF

Day 1	Day 2	Day 3	Day 4
-0.95	-0.86	-0.94	-0.88

The results of the repeated experiments demonstrated that HUF is a valid timeliness measure for web-based news media.

5.4.3 Empirical study 3: Feasibility study and validation of MTDTP, SES and HUF measures

The objectives of the third experiment are:

- to evaluate the feasibility of SES and to validate SES as a timeliness measure for news media
- to further evaluate the feasibility of HUF measure and to validate HUF as a timeliness measure for news media in a different context and network conditions
- to investigate the relationship between the SES, HUF and MTDTP measure.

In particular, the experiment aims at evaluating and validating the measures by testing their feasibilities under conditions of less ideal Internet connections, such as measuring the timeliness of the websites that are not located within the UK, and for validating the measures using the actual time delay in reporting news events with regard to the actual time that the event happened in the real world. To achieve these purposes, the following two aspects were considered in the design of the empirical study: the selection of a set of events and the selection of a set of websites. They are discussed as follows.

(1) *The set of events.* In recent Athens Olympic Games, Chinese athletics performed very well. They obtained 32 Gold Medals. Since the Olympic Games are reported by TV lively, the knowledge of when the events happened is available. In this experiment, the events chosen were that the Chinese athletics won a gold medal. The selection of these events is mainly because these events have a fairly equal importance for Chinese media.

(2) *The set of websites.* There are a great number of websites that report the Olympic Games. Chinese web-based media were chosen because of their equal views to all gold medals won by Chinese athletics to minimise the human factors that may affect the outcome of the empirical study. The following websites were amongst the top four when searching the keyword 'Olympics' in Google (Simplified Chinese):

- Sina (http://2004.sina.com.cn)
- TOM (http://2004.sports.tom.com)
- Sohu (http://2004.sports.sohu.com/)
- Yahoo (http://cn.sports.yahoo.com/olympic/)

The accurate time of Chinese obtaining each Gold Medal was provided by the Xinhua Net XinhuaNet (http://www.xinhuanet.com/olympic), listed in Table 5.6.

Table 5.6 Time of Chinese Team Obtaining Gold Medals (Beijing Time)

No.	Name	Date	Time	No.	Name	Date	Time
1	Du Li	14/08/04	16:25	17	Zhang Jiewen/Yang Wei	21/08/04	20:30
2	Wang Yifu	14/08/04	20:12	18	Tang Gonghong	21/08/04	23:12
3	Guo Jingjing/Wu Minxia	15/08/04	02:47	19	Jia Zhanbo	22/08/04	18:55
4	Tian Liang/Yang Jinhui	15/08/04	04:10	20	Zhang Yining	22/08/04	19:27
5	Xian Dongmei	15/08/04	22:43	21	Li Ting/Sun Tiantian	22/08/04	23:43
6	Zhu Qinan	16/08/04	20:22	22	Teng Haibin	23/08/04	02:35
7	Chen Yanqing	16/08/04	22:59	23	Wang Xu	24/08/04	01:09
8	Luo Xuejuan	17/08/04	01:20	24	Peng Bo	25/08/04	03:57
9	Shi Zhiyong	17/08/04	02:22	25	Guo Jingjing	26/08/04	03:00
10	Li Ting/Lao Lishi	17/08/04	02:41	26	Liu Xiang	28/08/04	02:33
11	Zhang Guozheng	19/08/04	02:18	27	Xing Huina	28/08/04	03:19
12	Zhang Ning	19/08/04	20:10	28	Meng Guanliang/Yang Wenjun	28/08/04	15:11
13	Liu Chunhong	19/08/04	23:13	29	Luo Wei	28/08/04	00:28
14	Zhang Jun/Gao Ling	19/08/04	23:35	30	Hu Jia	28/08/04	03:13
15	Wang Nan/Zhang Yining	20/08/04	19:38	31	Women's Volleyball Team	28/08/04	03:29
16	Ma Lin/Chen Qi	21/08/04	19:46	32	Chen Zhong	29/08/04	23:28

A software tool written in Perl was developed to measure the SES, HUF and MTDTP.

(1) For measuring MTDTP, a software tool monitored the homepage of each website, and downloaded it whenever it changed. It was found that all important news items, such as Gold Medal points, would appear in the homepage. A time fingerprint would be recorded as well. A human-involved check then would find out the publishing time of the Chinese Team obtaining each Gold Medal. It was found that all the above websites provided a time indicator, accurate to a minute, in the pages. There was a little time difference between the time indicator provided by the sites and the time measured. This was due to the two factors: network delays and the Web writers' mistakes.

(2) For measuring HUF, the same tool describes in Section 5.4.1 was used and the experiments of the first and second empirical studies was repeated.

(3) For measuring SES, a software tool downloaded the targeted sites every hour. It would take a long time to download the site in the first hour, and then would be much quicker as it only downloaded the changed, or added, pages. Only the Olympic sites were downloaded. For example, for Yahoo, the URL http://cn.sports.yahoo.com/olympic/ is regarded as the homepage and only downloaded the pages within this directory. All pages which were linked to other sites or other servers within Yahoo were ignored.

The software tools were running for over two weeks during the Games. Data was collected between 00:00 14th and 23:59 30th August 2004.

The data of three measures of the websites are listed in Table 5.7 - 5.11.

Table 5.7 MTDTP of four news websites (Minute)

Medal #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
TOM	1	1	1	2	1	3	2	1	2	2	1	1	1	1	3	1
Yahoo	19	20	7	8	7	5	4	5	4	5	5	3	6	2	6	3
Sohu	10	3	4	3	13	4	3	2	4	7	7	4	4	3	3	5
Sina	3	3	3	7	2	4	3	2	6	4	4	2	4	3	4	2
Medal # / Sites	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
TOM	1	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1
Yahoo	6	4	3	6	5	6	11	3	8	2	4	7	7	9	3	18
Sohu	6	2	3	3	6	6	8	1	1	1	2	3	6	4	1	1
Sina	3	2	2	4	4	2	2	4	2	2	2	2	4	2	2	3

Table 5.8 Number of Homepage Updates

Sites \ Date	14	15	16	17	18	19	20	21
TOM	463	471	433	417	432	489	450	473
Yahoo	183	215	301	304	329	420	308	324
Sohu	248	255	280	267	292	148	386	273
Sina	375	378	444	364	447	375	399	449
Sites \ Date	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
TOM	452	419	349	350	374	363	300	347
Yahoo	303	291	236	338	269	351	226	198
Sohu	266	247	168	356	324	258	246	340
Sina	375	378	444	364	447	375	399	449

Table 5.9 HUF of four websites

Sites \ Date	14	15	16	17	18	19	20	21
TOM	32.15	32.71	30.07	28.96	30.00	33.96	31.25	32.85
Yahoo	12.71	14.95	20.91	21.14	22.87	29.14	21.36	22.50
Sohu	17.22	17.71	19.44	18.54	20.28	10.28	26.81	18.96
Sina	26.04	26.22	30.84	25.30	31.02	26.04	27.70	31.21
Sites \ Date	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
TOM	31.39	29.10	24.24	24.31	25.97	25.21	20.83	24.10
Yahoo	21.06	20.23	16.38	23.48	18.65	24.38	15.70	13.74
Sohu	18.47	17.15	11.67	24.72	22.50	17.92	17.08	23.61
Sina	30.65	24.19	26.04	31.39	17.36	14.03	12.00	7.20

Table 5.10 Number of Pages Added

Sites \ Date	14	15	16	17	18	19	20	21
TOM	733	745	754	739	772	853	760	711
Yahoo	267	246	342	363	439	486	397	367
Sohu	296	270	366	298	386	160	442	310
Sina	433	398	495	398	516	406	446	532
Sites \ Date	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
TOM	759	635	612	508	651	632	560	672
Yahoo	329	361	287	378	306	397	259	278
Sohu	282	297	189	390	411	283	261	432
Sina	473	366	406	538	269	218	177	105

Table 5.11 SES of four websites

Sites \ Date	14	15	16	17	18	19	20	21
TOM	30.54	31.04	31.42	30.79	32.17	35.54	31.67	29.63
Yahoo	11.13	10.25	14.25	15.13	18.29	20.25	16.54	15.29
Sohu	12.33	11.25	15.25	12.42	16.08	6.67	18.42	12.92
Sina	18.04	16.58	20.61	16.58	21.51	16.92	18.60	22.18
Sites \ Date	22	23	24	25	26	27	28	29
TOM	31.63	26.46	25.50	21.17	27.13	26.33	23.33	28.00
Yahoo	13.71	15.04	11.96	15.75	12.75	16.54	10.79	11.58
Sohu	11.75	12.38	7.88	16.25	17.13	11.79	10.88	18.00
Sina	19.72	15.24	16.92	22.41	11.20	9.07	7.39	4.37

Results from the above tables showed that the TOM site performed the best. The other sites were similar despite the fact that Yahoo did not perform very well at the beginning and at the end. As the Gold Medals were not evenly distributed into 16 days, the results of MTDTP would not be the best data to validate the other measures. For example, on 18th August, Chinese Team had no Gold medal, but obtained 6 on 28th. Hence, MTDTP, HUF and SES were used to assess the timeliness of the websites, see Table 5.12.

Table 5.12 Timeliness measures results of four websites

Site \ Measure	$MTDTP_{Average}$	$HUF_{Average}$	$SES_{Average}$
TOM	2.69	0.29	28.90
Yahoo	13.19	0.20	14.33
Sohu	7.06	0.19	13.21
Sina	6.13	0.24	16.08

From Table 5.12, all measures agree that TOM was the best. Results of HUF and SES seem quite consistent. Both agree the ranking order should be: TOM – Sina – Yahoo – Sohu. But MTDTP showed the ranking order: TOM – Sina – Sohu – Yahoo. At this stage, the author would rather believe in the results of HUF and SES, as they showed the workload as well. Also, results of MTDTP came from an incomplete set of data, which did not include all the important events during the Olympics.

Correlation analysis was conducted, and the results are shown in Table 5.13.

Table 5.13 Correlations of three timeliness measures

	$MTDTP_{Average}$	$HUF_{Average}$	$SES_{Average}$
$MTDTP_{Average}$	-	-0.76	-0.72
$HUF_{Average}$	-0.76	-	0.94
$SES_{Average}$	-0.72	0.94	-

It can be seem that there exists strong correlation among the measures. Although the correlation coefficient between MTDTP and HUF or SES is comparatively lower, considering the difficulty of choosing MTDTP events, the results are satisfactory.

It is also interesting to note that HUF and SES have a high correlation. Considering SES results are comparatively rather difficult to obtain, the author argues that HUF is a simple and effective timeliness measure. Even occasionally some websites are updated several times within a minute, HUF can still reflect the whole picture of the timeliness with a satisfactory accuracy.

5.5 Summary

This chapter focuses on the timeliness of web-based media, which is an important quality attribute to such websites. The timeliness of a website is defined as its ability to create, update and present information within the tolerable time delay in order to keep the information consistent with the real world. According to the definition, three measures, MTDTP, HUF and SES, were proposed and defined as direct and indirect measures of timeliness. This chapter also discussed the advantages and disadvantages of each measure.

Empirical studies showed that the proposed measurement method was applicable. The measures were also validated and evaluated using the empirical studies, which were involved using a software tool and manual test. The statistical analysis of the empirical data showed that HUF and SES are efficient and effective measures of website timeliness.

Chapter 6 Conclusion

This study fulfilled the study aim and objectives proposed in Chapter 1. A quality modelling method for derivation of quality models of Web-based information systems from architectural design was developed in Chapter 3. Chapter 4 and Chapter 5 reported the work on the development, implementation and validation of quality metrics of website navigability and quality measures of timeliness respectively. This chapter summaries the findings of the study and gives conclusions drawn from the findings. Finally, some suggestions for future research will be given.

6.1 Conclusions

6.1.1 Quality modelling of Web-based information systems

This study surveyed existing well known quality models of software systems, information systems and Web-based systems. The existing quality models are not suitable to assess the quality of WIS. First, such models were mostly developed for traditional software and information systems. WIS have demonstrated many new features and quality concerns. Second, such models were developed mainly from developers' experiences and knowledge. The development of WIS has only a very short history. The experiences and knowledge about such systems are very limited. Third, such models were attempting to be generic models and applicable to all software systems. However, different types of WIS have posed different quality issues and requirements. Such models did not provide insight into the individual system under development and hence provide limited support to the design and implementation of individual system although they help to understand what the quality concerns are. The Web designers may not know how to improve the quality even if they are aware of the quality attributes. Fourth, most existing quality assessment and measurement techniques developed so far based on such quality models can only be applied in late phases of software life cycle. It would be much too costly to modify the design at that phase.

To address the above issues, this study proposed a quality modelling method based on system architectural analysis and hazard analysis techniques. The new method has the following advantages:

1. For WIS, quality issues are usually associated with various system usage procedures. For example, in e-commerce systems, information relevance is a major issue when users search for certain products; and security becomes the most important issue in online payment procedures. System architecture analysis enables designers to decompose the system in an objective way and makes it possible to identify which quality issues are associated to which part of the system, and represent such information in a product specific quality model. Thus, it makes the quality model more applicable by linking users' quality concerns with the systematic design.
2. It will distinguish one system from another by its architecture and its application domain. It enables designers to form a quality model for each system or a group of systems in the same architectural design rather than a generic and universal one.
3. It makes quality evaluation possible to be done as early as at system architectural design stage of the life cycle because the technique is applied to the architectural design.
4. Hazard analysis techniques enable the origins of failures to be systematically identified by classifying various types of faults, to recognise various failure modes by analysing how faults can be combined to trigger various kinds of failure processes, to understand the propagation of failures between components, to establish the cause effect chains and therefore the consequences of failures. Application of hazard analysis techniques in the quality modelling approach provides an objective view to quality attributes and their relationships. The quality attributes and their relationships are no longer obtained solely in subjective ways, such as questionnaire and interview, but are mainly derived from the systematic analysis.

As a case study, a quality model of e-commerce systems was developed using such a method. The result showed that the proposed method is applicable and promising. Web designers can apply the modelling technique to make their own quality models suitable for their own specific systems. Veteran e-commerce systems designers and experienced users were recruited during the quality modelling process. It can not only avoid subjectivity, but also validate the quality model and further, the quality modelling method.

Compared to the existing quality modelling approach, the findings in this study indicated that the new quality modelling method enables the practitioners to develop tailored quality models, which are:

- specific to the individual systems
- applicable in the early stages of system life cycle
- usable to provide insight into quality assessment and improvement.

6.1.2 Quality measurement of Web-based information systems

This study investigated the related software quality metrics and Web metrics. The existing quality metrics have one or more of the following drawbacks.

- lack of rigorous derivation and justification.
- inadequate validation
- irrelevant to WIS quality. Most Web metrics are measuring for general properties of the Web.

In this study, software quality engineering principles were applied to the development of quality metrics for WIS. The development of metrics followed the following steps:

1. clear and accurate definition of the quality attribute to be measured
2. rigorous derivation and justification of metrics
3. empirical and/or formal validation of the metrics.

Quality measurement of two quality attributes, navigability and timeliness, was investigated.

6.1.2.1 Navigability metrics

Website navigation design deals with how the users interact with the site and is related to the feel of the website. Thus, the measurement of navigability is a subjective and difficult task.

Existing research indicates a strong relationship between a website's structural complexity and its navigability. Such hypothesis formed the idea that website structural complexity metrics can be used to indirectly measure website navigability.

A directed graph model of a website was adopted to investigate and define six website structural complexity (WSC) metrics. These metrics can be grouped into three categories: number of links, number of linear independent paths of a site, and distributions of links. The metrics were assessed against the well-known Weyuker's axioms of software complexity. Some properties of the axioms were adapted or redefined to suit WIS.

Empirical studies were conducted to evaluate if the WSC metrics could be used as navigability metrics. A questionnaire was designed to cover all the major navigation design concerns according to usability standard. The websites used in the empirical study are four university portals. 286 subjects from different universities were selected at random to participate in the empirical study.

A software tool was developed to calculate the complexities of these universities' websites according to the metrics, at the same day of the questionnaire survey was conducted. Statistical analysis of measurement results in comparison with the subjective ratings of navigability from the questionnaires showed strong correlation between them.

Results of the empirical study proved that there is clear evidence of validity of using website structural complexity metrics to measure navigability. Based on the findings, the empirical results proved the hypothesis that structural complexity contributes significantly to the navigability of the system. It showed that increasing of website structure complexity would lead to the deceasing in navigability.

6.1.2.2 Timeliness measures

Timeliness is an important attribute for media-based websites. According to the definition given in the study, three timeliness measures were defined and justified as direct or indirect measures, which include Mean Time Delay To Publish (MTDTP), Site Evolution Speed (SES) and Homepage Update Frequency (HUF).

Empirical studies were conducted to validate the measures. MTDTP is a measure directly derived from the definition of timeliness. The applicability of MTDTP relies on the availability of the knowledge of the accurate time when events happen in the real world and the time when the events are reported by a website. The latter can often be obtained from the meta-data published by the website or using other technologies, hence it can be obtained automatically. However, the former is more or less depending on manual collection of data.

Empirical studies found that HUF was a simple but very effective timeliness measure. Results showed that at the sampling rate of 1 per minute, the calculation is efficient. Statistical analysis of the results of validation experiments showed a strong negative correlation between HUF and timeliness. This means that the higher HUF implies less time delay in reporting real world events. Therefore, HUF is a proper measurement of timeliness.

Empirical study also proved that SES was a proper measure for timeliness. It can be calculated automatically. The SES measure is accurate under the condition that the website does not delete web pages, merge or split web pages. Empirical study showed that this condition was true for the websites that were used in the empirical study. Empirical results also showed a strong positive correlation between HUF and SES.

Another interesting finding was that, although different HUF was found for different news websites, a common pattern of HUF values over 24 hours was found. Whether such a pattern of HUF can be used to distinguish different types of website remains an interesting topic for future study.

6.2 Future research

The quality of WIS is a very complicated issue. There are several areas in which further research may be fruitful.

1. There are many types of Web-based systems, such as e-commerce, e-leaning, e-publishing, e-government, etc. Such systems have different requirements and system architectures, and thus, different design concerns and quality attributes. The same attribute may have different weight in different systems and demonstrate their behaviour in different designs. More empirical studies need to be conducted, where quality models for different types of WIS should be developed to further validate the quality modelling method. In addition, with the increase of the systems complexity, the quality modelling would become a tedious task. A software tool to support the quality modelling work will be useful.

2. This study investigated quality measurement issues of two quality attributes, timeliness and navigability. Besides these attributes, there are some other important quality attributes, such as security for most Web-based systems, information relevance especially for search engines, maintainability, testability, portability, and so forth. These quality attributes also need objective and quantifiable measures to obtain an insight into the overall quality of WIS.

3. The importance of same quality attribute on different websites may be different. For example, Web security is an extremely important quality attribute for e-commerce and e-government websites, but not so important for personal homepages. Therefore, it is an interesting but difficult task to find the

weight ratings of all quality measures for different websites. Can such weightings be determined or derived from the quality model objectively?

4. It is important to develop software tools to support the measurement of WIS. As the WIS are usually dynamic, measurements of dynamic features such as timeliness play a crucial role. Traditional approach to the software metrics tools is mostly for the measurement of static features of software systems. It may not suit the need of dynamic measurement. An agent-oriented approach to quality measurement of such dynamic systems seems to be a possible solution. In (Zhu, et al, 2000) a multi-agent approach towards WIS quality management was investigated. In the approach, software agents cooperate to monitor the websites, record the changes and calculate the metrics while one or more agents are responsible for one specific quality assurance task. The development of such a prototype system is in progress.

References

Andersson, T. and Eriksson, I.V. (1996) Measuring the quality needs of an organisation's software, Proceedings of the 29th Annual International Conference on System Sciences, vol. 4, pp. 139-154.

Andersson, T. and von Hellens, L.A. (1997) Information systems work quality, Information and Software Technology, vol. 39, pp. 837-844.

Auer, T. (1998) Quality of IS use, European Journal of Information Systems, 7, pp. 192-201.

Bachiochi, D., Berstene, M., Chouinard, E., Conlan, N., Danchak, M., Furey, T., Neligon, C. and Way, D. (1997) Usability studies and designing navigational aids for the World Wide Web, Journal of Computer Networks and ISDN Systems, vol. 29, pp. 1489-1496.

Baker, A., Bieman, J.M, Fenton, N., Gustafson, D.A., Melton, A.C. and Whitty, R. (1990) Philosophy for software measurement, Journal of Systems & Software, vol. 12, pp. 277-281.

Bansiya, J. and Davis, C.G. (2002) A Hierarchical model for object-oriented design quality assessment, IEEE Transaction on Software Engineering, Vol. 28, No. 1, pp.4-17.

Barfield, L. (2004) Design for New Media: Interaction Design for Multimedia and the Web, London: Addison-Wesley.

Belady, L.A. and Evangelisti, C.J. (1981) System partitioning and its measures, Journal of Systems and Software, 2, pp. 23-29.

Berners-Lee, T. (1999) Weaving the Web: The original Design and Ultimate Destiny of the World Wide Web by its Inventor, New York: HarperCollins Publishers.

Bieman, J.M., Fenton, N.E., Gustafson, D.A., Melton, A. and Ott, L.M. (1995) Fundamental issues in software measurement, in Software Measurement, (eds., A. Melton) London: International Thomson Publishing Company, pp. 39-52.

Bieman, J.M., Fenton, N.E., Gustafson, D.A., Melton, A. and Whitty, R. (1992) Moving from philosophy to practice in software measurement, in Proc. of FACS Workshop on Software Metrics, London: Springer-Verlag, pp. 38-59.

Boehm, B.W. Brown, J.R. and Lipow, M. (1976) Quantitative evaluation of software quality, Proceedings of the Second International Conference on Software Engineering, pp. 592-605.

Boehm, B.W. Brown, J. Kaspar, H. Lipow, M. MacLeod, G and Merrit, M. (1978) Characteristics of Software Quality, Vol. 1 of TRW Serious of Software Technology, North-Holland, New York.

Boehm, B.W. (1981) Software Engineering Economics, Prentice Hall, Englewood Cliffs, NJ.

Borges, J.A., Morales, I. And Rodrigues, J. (1996) Guidelines for designing usable World Wide Web pages, Proceedings of ACM CHI '96 Conference on Human Factors in Computing Systems, vol. 2, pp. 277-178.

Botafago, R., Rivlin, E. and Shneiderman, B. (1992) Structural analysis of hypertext: Identifying hierarchies and useful metrics, ACM Trans. on Information Systems, 10(2), April, pp. 142-180.

Brajnik, G. (2001) Towards valid quality models of websites, Proceedings of the 7th Conference on Human Factors and the Web, Madison, Wisconsin, June, 2001.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. and Wiener, J. (2000) Graph Structure of the Web, Proceedings of the 9th International World Wide Web Conference, Amsterdam, Netherlands, May 15 - 19, 2000.

Buchanan, R. Jr., and Lukaszewski, C. (1997) Measuring the Impact of Your Web Site, New York: John Wiley and Sons Inc.

Carmel, E., Crawford, S. and Chen, H. (1992) Browsing in hypertext: A cognitive study, IEEE Transaction on Systems, Man and Cybernetics, pp.111-117.

Card, D.N. and Glass, R.L. (1990) Measuring Software Design Quality, Englewood Cliffs, N.J: Prentice-Hall.

Chan, H., Lee, R. Dillion, T. and Chang, E. (2001) E-Commerce, Chichester: John Wiley & Sons.

Cherniavsky, J.C. and Smith, C.H (1991) On Weyuker axioms for software complexity measures, IEEE Transactions on Software Engineering, **17**, pp. 636-638.

Collins Memorial Library (2004) Criteria for Evaluating Information on the Web: Accuracy and Currency, http://library.ups.edu/research/handouts/time.htm, accessed on 25 June 2004

Comber, T. (1995) Building usable Web pages: An HCI perspective, Proceedings of the First Australian World Wide Web Conference, Norsearch, Ballina, pp. 119-124.

CSUS (2003) Evaluating Web timeliness, http://library.csus.edu/services/inst/ICCS/infocomp/tutorials/module5/web/timeliness/ , accessed on 25 June 2004.

Curtis, B. (1979) In search of software complexity, Proc. IEEE Workshop on Quantitative Software Models, pp. 95-106.

Cutis, B. (1981) The measurement of software quality and complexity, in Software Metrics (eds., A. Perlis, F. Sayward and M. Shaw) pp. 203-223, MIT Press, Cambridge, MA.

Dahlberg, T and Jarvinen, J. (1997) Challenges to IS quality, Information and Software Technology, vol. 39, pp. 809-818.

Dalgleish, G. (2000) Customer-effective Web Site, Prentice Hall.

Darken, R.P. and Siebert, J.L. (1993) A toolset for navigation in virtual environments, UIST 93 Proceedings, p.157.

Delone, W.H. and McLean, E.R. (1992) Information systems success: the quest for the dependent variable, Information Systems Research, 3(1), pp. 60-95.

De Vaus, D.A. (2001) Surveys in Social Research, 5th Edition, London: Routledge.

Dhyani, D., NG, W.K. and Bhowmick, S.S. (2002) A survey of Web metrics, ACM Computing Surveys, Vol. 34, No. 4, December 2002, pp. 469-503.

Dromey, R. G. (1995) A model for software product quality, IEEE Transactions on Software Engineering, Vol. 21, No. 2, Feb, pp146-162.

Dromey, R. G. (1996) Cornering the Chimera, IEEE Software, Vol. 13, No. 1, January, pp.33-43.

Easterby-Smith, M., Thorpe, R. and Lowe, A. (1991) Management Research: AN Introduction, London: Sage, pp. 119-134.

Eriksson, I. and Torn, A. (1991) A model for IS quality, Software Engineering Journal, July, pp. 152-158.

Eriksson, I. and Torn, A. (1997) Introduction to IST special issue on information systems quality, Information and software technology, vol. 39, pp. 797-799.

Feghali, I., Watson, A., Henderson-Sellers, B. and Tegarden, D. (1994) Clarification concerning modularization and McCabe's cyclomatic, Communications of the ACM, Vol. 37, No. 4, pp. 91-94.

Fenton, N.E. (1994) Software measurement: a necessary scientific basis, IEEE Trans. on Software Engineering, 20(3), pp. 199-206.

Fenton, N.E. and Pfleeger, S.L. (1997) Software Metrics: A Rigorous & Practical Approach, 2nd edn, PWS Publishing Company, Boston, MA.

Fenton, N.E. and Whitty, R.W. (1986) Axiomatic approach to software metrication through program decomposition, Computer Journal, 29(4), pp. 330-339.

Finkelstein, L. (1984) A review of the fundamental concepts of measurement, Measurement, 2(1), pp. 25-34.

Fleming, J. (1998) Web Navigation: Designing the User Experience, O'Reiley.

Furnas, G.W. (1997) Effective view navigation, in Proceedings of CHI 1997, Human Factors in Computing, ACM Press, pp. 22-27

Garvin, D. (1984) What does product quality mean, Slogan Management Review, vol. 4.

Gilb, T. (1988) Principles of Software Engineering Management, Addison-Wesley, Reading MA.

Gillies, A.C. (1992) Modelling software quality in the commercial environment, Software Quality Journal, 1, pp. 175-91.

Gillies, A.C. (1997) Software Quality: Theory and Management, 2nd edn, International Thomson Computer Press.

Greenfield, T. (2002) Research Methods for Postgraduates, 2nd edition, London: Arnold, pp. 151-160.

Hall, W. (1999), All in one: looking forward to communication integration, in 2020 Visions, EPSRC Newsline, (Special Issue), Sept., page 3.

Halstead, M.H. (1977) Elements of Software Science, New York: Elsevier North Holland.

Hamer, P. and Frewin, G. (1982) Halstead's software science: A critical examination. Proc. of 6th International Conference on Software Engineering, pp. 197-206.

Hearst, M. (1998), Trends and controversies, IEEE Intelligent Systems, Sept/ Oct.

van Hee, K.M. (1994) Information System Engineering: A Formal Approach, Cambridge University Press.

von Hellens, L.A. (1997) Information systems quality versus software quality: a discussion form a managerial, an organisational and an engineering viewpoint, Information and Software Technology, vol. 39, pp. 801-808.

Henry, S. and Kafura, D. (1981) Software structure metrics based upon information flow, IEEE Transaction on Software Engineering, SE-7 (5), pp. 510-518.

Henry, S., Kafura, D. and Harris, K. (1981) On the relationships among three software metrics, Proc. of ACM Symposium on Measurement and Evaluation of Software Quality, Associations for Computing Machinery, pp. 81-88.

Henry, S. and Selig, C. (1990) Predicting Source-Code Complexity at the Design Stage, IEEE Software, Volume 7, Issue 2, Pages: 36 – 44.

IEEE (1990) IEEE Standard Glossary of Software Engineering Terminoogy, IEEE Std 610.12.

IEEE (1992) Standard 1061: Software Quality Metrics Methodology, IEEE Computer Society Press.

ISO (1986) Quality Vocabulary, ISO 8042.

ISO (1987) ISO 9001: Quality Systems: Models for Quality Assurance in design, Development, Production, Installation and Servicing, International standards Organisation.

ISO (1991) ISO 9126: Information Technology – Software Product Evaluation – Quality Characteristics and Guide Lines for Their Use, ISO/IEC IS 9126, Geneva, Switzerland.

Ivory, M.Y., Sinha, R.R. and Hearst, M.A. (2001) Empirically validated Web page design metrics, Proc. 01'SIGCHI, 31 March – 4 April 2001 Seattle, WA, USA, pp. 53-60.

Jin, L., Zhu, H. and Hall, P. (1997) Adequate Testing of Hypertext Applications, Journal of Information and Software Technology, Vol. 39, No. 4, pp225-234.

Kan, S.H. (1995) Metrics and Models in Software Quality Engineering, Reading: Addison-Wesley.

Kim, J., Li, Y., Moy, N. and Ni, M. (2001) The menu design and navigational efficiency of the E-Maryland Portal, http://www.otal.umd.edu/SHORE2001/emaryland/index.html, accessed on 20 March 2003.

Kitchenham, B.A. and Pickard, L. (1987) Towards a constructive quality model Part 2: Statistical techniques for modelling software quality in the ESPRIT REQUEST project, Software Engineering Journal, July.

Kitchenham, B.A., Pickard, L. and Pfleeger, S.L. (1995) Case studies for method and tool evaluation, IEEE Software, 12(4), pp. 52-62.

Kitchenham, B.A. and Walker, J.G. (1989) A quantitative approach to monitoring software development, Software Engineering Journal, 4(1), pp. 2-13.

Kletz, T. (1995) Computer control and Human Error, Rugby: Institute of Chemical Engineers.

Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A. (1999) Trawling the Web for emerging cyber-communities, Proceedings of the 8th World Wide Web Conference. Toronto, Canada May 11-14, 1999.

Krantz, D.H., Luce, R.D., Suppes, P. and Tversky, A. (1971) Foundations of Measurement, Volume I, New York: Academic Press.

Krug, S. (2000) Don't Make Me Think: A Common Sense Approach to Web Usability, New Riders.

Leveson, N.G. (1995) Safeware: System Safety and Comuters, Reading, MAL Addison-Wesley.

Lindroos, K. (1997) Use quality and the World Wide Web, Information and Software Technology, vol. 39, pp. 827-836.

Lowe, D. and Hall. W. (1999) Hypermedia and the Web: An Engineering Approach, Chichester: John Wiley and Sons.

Luce, R.D., Krantz, D.H., Suppes, P. and Tversky, A. (1990) Foundations of Measurement, vol. III, San Diego: Academic Press.

Lynch, P.J. and Horton, S. (1999) Web Style Guide: Basic Design Principles for Creating Web Sites, Yale University Press.

MacDonnell, S.G. (1991) Rigor in software complexity measurement experimentation, Journal of Systems and Software, 16, pp. 141-9.

McCabe, T.J. (1976) A complexity measure, IEEE Transactions on Software Engineering, Vol. 2, No. 4, December 1976, pp. 308-320.

McCall, J. Richards, P. and Walters. G. (1977) Factors in software quality, Technical report CDRL A003, US Rome Air Development Centre, Vol. I.

McCracken, D.D. and Wolfe, R.J. (2004) User-Centered Website Development: A Human-Computer Interaction Approach, Prentice Hall, New Jersey.

McGovern, G. (2001) Web Content Style Guide, Part 1: Navigation Design Principles, Prentice Hall.

Melton, A.C., Gustafson, D.A., Bieman, J.M. and Baker, A.L. (1990) A mathematical perspective for software measures research, IEEE Software Engineering Journal, September, pp. 246-254.

Miller, E. (2000) The website quality challenge, Software Research Inc, http://www.soft.com/eValid/Technology/White.Papers/website.quality.html, accessed 01-03-2002.

Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, Communications of the ACM 33, 3 (March), pp.338-348.

Myers, G.J. (1978) Composite Structured Design, Wokingham, UK: Van Nostrand Reinhold.

Navarro, A. and Khan, T. (1996) Effective Web Design: Master the Essential, San Francisco: SYBEX.

Neumann, P.G. (1995) Computer-Related Risks, New York: ACM Press.

Nielsen, J. (1993) Usability Engineering, AP Professional, Cambridge, Massachusetts.

Nielsen, J. (1996) Application and web interface design with Java, JavaDay, 12(2), Inter-Continental Helsinki.

Nielsen, J. (1999) Design Web Usability, Indiana: New Riders, USA.

Nielsen, J. (2003) How to conduct a heuristic evaluation, http://www.useit.com/papers/heuristics , accessed 10-03-2003.

Nielsen, J. and Mack, R. (1994) Usability Inspection Methods, New York: John Wiley.

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), pp. 249-256.

NIST (2004) Metrics and Measures, a newly added sestion in 'SAMATE', http://hissa.nist.gov/-black/SAMATE/ Institute of Standards and Technology.

Olsina, L., Godoy, D., Lafuente, G. and Rossi, G. (1999) Specifying Quality Characteristics and Attributes for Websites, Proceedings of the First ICSE Workshop on web Engineering, 16 -17 May Los Angeles, USA.

Ott, L.M. (1995) The early days of software metrics: looking back after 20 years, in Software Measurement (eds. A. Melton) International Thomson Publishing Company, pp. 7-26.

Pedhazur, E.J. and Schmelkin, L. (1991) Measurement, Design, and Analysis: A Integrated Approach, Lawrence Erlbaum Associates, Inc.

Perry, W. (1987) Effective Methods of EDP Quality Assurance, 2nd edn, Prentice-Hall.

Perry, W. (1991) Quality Assurance for Information systems: Methods, Tools and Techniques, New York: John Wiley & Sons.

Powell, T.A. (1998) Web Site Engineering: Beyond Web Page Design, Prentice Hall, PTR, NJ.

Powell, T.A. (2002) Web Design: the Complete Reference, 2nd Edition, New York: McGraw Hill/Osborne.

Prather, R.E. (1984) An axiomatic theory of software complexity measure, Computer Journal, 27(4), pp. 340-348.

Ragland, B. (1995) Measure, Metric, or Indicator: What's the Difference, http://www.stsc.hill.af.mil/crosstalk/1995/03/Measure.asp

Reeves, C.A. and Bednar, D.A. (1994) Defining quality: alternatives and implications, Academy of Management Review, 19(3), pp. 419-445.

Richards, P. and Walters, G. (1977) Factors in Software Quality, Technical report CDRL A003, US Rome Air Development Centre, Vol. I.

Rivest, R.L. (1992) RFC 1321: The MD5 Message-Digest Algorithm, Internet Activities Board.

Rodriguez, M.G., Emilio, J., Gayo, L., Lovelle, J. (2001) Web Navigability Testing with Remote Agents, New Riders.

Rodriguez, M.G. (2002) Automatic data-gathering agents for remote navigability testing, http://dsonline.computer.org/0212/f/s6rod.htm, accessed on 24 March 2003

Salmela, H. (1997) From information systems quality to sustainable business quality, Information and Software Technology, vol. 39, pp. 819-825.

Shepperd, M. (1995) Foundations of Software Measurement, Prentice Hall.

Shneiderman, B. (1998) Designing the User Interface, 3rd Edition, Reading: Addison Wesley.

Siegel, D.S. (1997) Secrets of Successful Web Sites: Project Management on the World Wide Web, Hayden Books.

Siegel, S. and Castellan, N.J. Jr., (1988) Nonparametrics Statistics for the Behavioural Sciences, 2nd edn, McGraw-Hill, New York.

Smith, P.A (1996) Towards a practical measure of hypertext usability, Interacting with Computer, vol. 8, No. 4, pp. 365-381.

Smith, S.L. (1986) Standard versus guidelines for designing user interface software, Behaviour and Information Technology, 5(1), pp. 47-61.

Sommerville, I. (2001) Software Engineering, 6th edition, Harlow : Addison-Wesley.

Spool, J.M., Scanlon, T., Schroeder, W., Snyder, C. and DeAngelo, T. (1999) Web Site Usability: A Designer's Guide, CA: Morgan Kufmann Publishers, pp. 33-48.

Storey, N. (1996) Safety-Critical Computer Systems, Reading, MA: Addison, pp. 33-58.

Sullivan, T. (1997) Reading reader reaction: a proposal for inferential analysis of Web server log files, http://www.pantos.org/ts/papers/rrr.htm, accessed on 20 March 2003.

Suppes, P., Krantz, D.H., Luce, R.D. and Tversky, A. (1989) Foundations of Measurement, vol. II, New York: Academic Press.

Swanson, E.B (1997) Maintaining IS quality, Information and Software Technology, vol. 39, pp. 845-850.

Tes, (2002) Developing a basic numeric metric for Web usability, http://www.kuro5hin.org, 14 April 2002, accessed on 2nd March 2003.

Upchurch, L., Rugg, G. and Kitchenham, B. (2001) Using card sorts to elicit web page quality attributes, Journal of IEEE Software, July/Aug, pp.84-85.

Virtual Chase (2000) Information timeliness, http://www.virtualchase.com/govdoc/timely.html, accessed on 25 June 2004.

Wallmuller, E. (1994) Software Quality Assurance: A Practical, Hemel Hempstead : Prentice-Hall.

Watts, R. (1987) Measuring Software Quality, NCC Publications, Manchester, UK.

Weyuker, E.J. (1988) Evaluating software complexity measures, IEEE Transaction on Software Engineering, SE-14(9), pp. 1357-65.

Wikipedia (2005) Software metric, http://en.wikipedia.org/wiki/Software_Metric .

Wise, R. (2000) The 3-click Rule, http://www.website-owner.com/articles/design/3clickrule.asp, accessed on 25 June 2003.

Yau, S.S. and Collofello, J.S. (1980) Some stability measures for software maintenance, IEEE Transactions of Software Engineering, Vol. SE-6, pp. 545-552.

Zaphris, P. and Mtei, L. (1997) Depth vs. breadth in the arrangement of Web links, http://www.otal.umd.edu/SHORE97/, accessed on 20 March 2003.

Zhang, Y., Huo, Q., Zhang, L., Zhu, H., Greenwood, S. (2001) Measuring the Timeliness of Websites, Proceedings of the Fourth European Conference on Software Measurement and ICT Control in Co-operation with DASMA, Heidelberg, Germany, 8-11 May 2001, pp. 139-147.

Zhang, Y., Zhang, L., Huo, Q., Zhu, H.and Greenwood, S. (2001) Structure and Page Complexity Metrics for Web Applications, Proceedings of the Fourth WWW10 Workshop on Web Engineering, Hong Kong, 1-5 May 2001, pp 72-81.

Zhang, Y., Zhu, H. and Greenwood, S. (2001) Overview of Quality Models of Information Systems, submitted to the Journal of Informatica.

Zhang, Y., Zhu, H., Greenwood, S. and Huo, Q. (2001) Quality Modelling of Web-based Information Systems, Proceedings of the Eighth IEEE Workshop on Future

Trends of Distributed Computing Systems, Bologna, Italy, 31 October - 2 November, pp. 41-47.

Zhang, Y., Zhu, H., Greenwood, S. and Huo, Q. (2002) A Method to Derive Quality Attributes for Web-based Information Systems, Proceedings of the Second ASCERC Workshop QSSE 2002, Banff, Alberta, Canada, 18 - 21 February.

Zhang, Y., Zhu, H., Huo, Q. and Greenwood, S. (2002) Measurement of Timeliness of Web-based Information Systems, Proceedings of the 6th World Multi-Conference on Systemic, Cybernetics and Informatics (SCI 2002), July 14-18, 2002, Orlando, Florida, USA.

Zhang, Y., Zhu, H., Greenwood, S. (2004). Website Complexity Metrics for Measuring Navigability. Technical Report DOC-TR-2004-03, Department of Computing, Oxford Brookes University, UK, March 2004.

Zhu, H and Hall, P. (1993) Test data adequacy measurement, Software Engineering Journal, January, pp. 21-28.

Zhu, H., Greenwood, G., Huo, Q. and Zhang, Y. (2000) Towards Agent-Oriented Quality Management of Information Systems, Workshop Notes of Second International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2000) at AAAI'2000, Austin, USA, July 30, pp.57-64.

Zhu, H., Zhang, Y., Huo, Q. and Greenwood, S. (2002) Application of Hazard Analysis to Quality Modelling, in Proceedings of the 26th COMPSAC Conference, Oxford, UK, August.

Zuse, H. (1992) Properties of software measures, Software Quality Journal, vol. 1, pp. 225-260.

Appendices

Appendix A. List of Publications

A1. Towards Agent-Oriented Quality Management of Information Systems
Hong Zhu, Sue Greenwood, Qingning Huo and Yanlong Zhang
Workshop Notes of Second International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2000) at AAAI'2000, Austin, USA, July 30 2000, pp.57-64.

Abstract

The emergence of agent-oriented information systems imposes a great challenge to the current theories and methods of software quality management. The difficulties in the development of quality information systems are not only due to the openness of the environment, the diversity of platforms, the vast volume of distributed information in diverse formats, and the complicated combination of data and program, but also the dynamic lifecycles of such systems.

In the paper, we report work in progress on agent-oriented quality management of distributed hypermedia and cooperative information systems. We first analyze the challenges that agent-oriented information systems impose upon established theory and technology of software quality management. We then present our growth lifecycle model of information systems development and outline our approach towards intelligent tool support of quality management activities in the development of information systems.

A2. Structure and Page Complexity Metrics for Web Applications
Yanlong Zhang, Lu Zhang, Qingning Huo, Hong Zhu and Sue Greenwood
Proceedings of the Fourth WWW10 Workshop on Web Engineering, Hong Kong, May 1-5, pp.72-81

Abstract

Research on quality issues of web applications comes up only very recently. In this paper, we exploit the graph model and the hierarchical model to represent respectively the structure of web applications and the hierarchy of web pages, and based on the two models, we present two complexity metrics for web applications, which may help measuring the complexity of web applications. We also apply the well-known Weyuker's nine properties for examining the theoretical feasibility of the metrics.

A3. Measuring the Timeliness of Websites
Yanlong Zhang, Qingning Huo, Lu Zhang, Hong Zhu and Sue Greenwood
Proceedings of the Fourth European Conference on Software Measurement and ICT Control in Co-operation with DASMA, Heidelberg, Germany, May 8-11 2001, pp.139-147

Abstract

Timeliness is an important quality factor for websites. In this paper, we present an automated approach to measuring the timeliness. Four metrics for measuring website timeliness are proposed in this paper, which include the change frequency, the structural change of a website, the editing distance and the vector distance between web pages. A prototype system has been designed and implemented to realise the metrics. Some preliminary results of applying the prototype system are also reported.

A4. Quality Modelling of Web-Based Information Systems
Yanlong Zhang, Hong Zhu, Sue Greenwood and Qingning Huo
Proceedings of the Eighth IEEE Workshop on Future Trends of Distributed Computing Systems, Bologna, Italky, October 31 – November 2 2001, pp.41-47

Abstract

World Wide Web has become an important medium of software applications. As web-based information systems are distributed, multimedia, co-operative and even intelligent, they have imposed great challenges to the quality management theories and methods of conventional software systems. In this paper, we present a framework of modelling the quality of web-based information systems. We firstly use information flow analysis to break down a complex system into sub-systems. Then within each sub-system, we apply hazard analysis techniques to find out the system failure mode, possible causes, and further, we find out the relevant quality attributes. As a case study, we use such method to develop a quality model of Business-to-Business e-business systems. Three users, all having rich experiences in using e-commerce systems have been interviewed in the modelling process.

A5. Deriving Quality Models of Web-Based Information Systems
Yanlong Zhang, Hong Zhu, Sue Greenwood and Qingning Huo
Proceedings of the Second ASCERS Workshop QSSE 2002, Banff, Canada, February 18-21 2002.

Abstract

To support a wide range of software development activities, software quality models should provide the system specific information about quality, such as to what extend the system is sensitive to a quality attribute and how a quality attribute can be addressed in the design, implementation and testing of the system. Such quality modelling is of particular importance for web-based information systems, not only because the variety of applications of web-base systems, but also the distributed, multimedia, co-operative and even intelligent features of such systems. The complexity of such systems due to these features has imposed great challenges to existing quality management theories, methods and techniques developed for conventional software systems over the past three decades.

In this paper, we present a method for deriving quality models of web-based information systems from their architectural models. The method starts with information flow analysis to break down a complex system into sub-systems. Hazard analysis techniques are then applied to investigate the system's failure modes and their possible causes and consequences so that the relevant quality attributes can be identified and their importance can be recognised. This paper reports a case study in the application of the method to Business-to-Business e-commerce systems.

A6. Application of Hazard Analysis to Quality Modelling
Hong Zhu, Yanlong Zhang, Qingning Huo, Sue Greenwood
Proceedings of the 26th COMPSAC Conference, Oxford, UK. August 2002, p.139-144.

Abstract
Quality is a fundamental concept in software and information system development. It is also a complex and elusive concept. A large number of quality models have been developed for understanding, measuring and predicting qualities of software and information systems. It has been recognised that quality models should be constructed in accordance to the specific features of the application domain. This paper proposes a systematic method for constructing quality models of information systems. A diagrammatic notation is devised to represent quality models that enclose application specific features. Techniques of hazard analysis for the development and deployment of safety related systems are adapted for deriving quality models from system architectural designs. The method is illustrated by a part of web-based information systems.

A7. Measurement of Timeliness of Web-based Information Systems
Yanlong Zhang, Hong Zhu, Qingning Huo and Sue Greenwood
Proceedings of the Sixth World Multi-Conference on Systemic, Cybernetics and Informatics (SCI 2002), Vol. 18, July 14-18 2002, Orlando, USA.

Abstract
The Internet was originally conceived to act as an information highway -- a venue for the exchange of information. Now its function has been expanding dramatically, such as e-commerce, e-government, e-banking, etc., and it has penetrated into every corner of our daily life. Timeliness of information is of prime importance for web-based information systems because of the web'S specific feature. The information age has brought with it a tendency to think that only timely information is quality information. For web-based information systems, several quality models have been developed, and timeliness is included into these quality models as an important quality attribute. It is therefore of importance to understand the timeliness and the ways to measure it.
In this paper, we present an automated approach to measuring timeliness. Metrics of measuring timeliness are proposed. We have also implemented experiments to test the metrics. Some results are provided.

A8. Website Complexity Metrics for Measuring Navigability
Yanlong Zhang, Hong Zhu and Sue Greenwood
Proceedings of the Fourth International Conference on Quality Software (QSIC 2004), Braunschweig, Germany, September 8-9 2004, pp.172-170

Abstract
In recent years, navigability has become the pivot of website designs. Existing works fall into two categories. The first is to evaluate and assess a website's navigability against a set of criteria or check list. The second is to analyse usage data of the website, such as the server log files. This paper investigates a metric approach to website navigability measurement. In comparison with existing assessment and analysis methods, navigability metrics have the advantages of objectiveness and the possibility of using automated tools to evaluate large-scale websites. This paper proposes a number of metrics for website navigability measurement based on measuring website structural complexity. We will validate these metrics against Weyuker's software complexity axioms, and report the results of empirical studies of the metrics

A9. Empirical Validation of Website Timeliness Metrics

Yanlong Zhang, Hong Zhu and Sue Greenwood

Accepted by the 29th COMPSAC Conference, July 2005, Edinburgh, UK.

Abstract

Information timeliness is crucial for websites. Although a couple of timeliness design strategies have been developed, few quality metrics can be found in the literature to provide a quantifiable measure, which provides an insight to the issue. Based on our previous research, this paper presents three timeliness metrics and reports the empirically validation of them in a case study conducted recently.

Appendix B. Questionnaire (used in Chapter 4)

Questionnaire

Section 1: About this survey

This survey forms part of a comprehensive study into the critical success factors of Web navigation design. The aim of the study is to develop quality metrics for Web navigability measurement. This survey is looking for end-users' perceptions and opinions of Web navigation design. Your responses will be invaluable to this study and will be treated in total confidence. So please take the time to complete this questionnaire.

Section 2: About yourself

1. How old are you?
 □ 18-20 □ 21-25 □ 26-34 □ over 34
2. Are you? □ female □ male
3. What course are you currently studying?
 □ Foundation degree □ HND/HNC □ BSc/BA □ MSc/MA
4. Averagely, how much time do you spend in surfing the Internet every week?
 □ 0-2 hours □ 3-5 □ 6-10 □ over 11

Section 3: About the Web navigation design

Please open University of Manchester's homepage in the 'Bookmark', and answers the Question 5-16.

5. If you want to do BSc.(Hons.) course in Chemistry, give one course title provided by the University.

____

6. Now, are there any alternative paths for you to find the answer to Question 5?
 □ alternative path easily found □ not easy, but still can find one □ hard to find
7. If an international student wants to do Master's degree in Law, what is the minimum score he or she should obtain for an IELTS test?

____

8. What is the URL of the University's library homepage?

____

9. According to the site, at 2002/2003 pricing, how much is it for a single room per week in a Hall of Residence with meals provided by the University?

____

10. The navigational items are clearly organised and the layout is consistent throughout the site.
 □ strongly agree □ agree □ undecided □ disagree □ strongly disagree
11. Hyperlinks on the website are easy to find, by 'cursor change' or contrast with the texts around.
 □ strongly agree □ agree □ undecided □ disagree □ strongly disagree

12. Colour of hyperlinks changes after the link is visited. This makes me easy to know which links have been clicked.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
13. The labels of hyperlinks on the website are clear and meaningful. I can predict where I am going from reading the labels.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
14. Do you encounter any unexpected navigational behaviour, such as broken links?
□ Yes, more than once □ Yes, once □ No
15. The website provides the following navigational facilities (tick more if necessary)
□ search engine □ menu □ Navigational bars □ site map □ others
____
16. The website provides supportive mechanism for disabled people, such large-font for weak-sighted people.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree

Please open Manchester Metropolitan University's homepage in the 'Bookmark', and please answer the Question 17 - Question 28.
17. How many undergraduate courses on Economics provided by the University?

____

18. Now, are there any alternative paths for you to find the answer to Question 17?
□ alternative path easily found □ not easy, but still can find one □ hard to find
19. If an international student would like to do Foundation Degree in Arts, and obtained an IELTS 5.0, is it enough for language entry requirement?

____

20. What is the URL of the University's library homepage?

____

21. If you want to express your opinions or comments on your experiences of the Student Union at MMU, what is the email address?

____

22. The navigational items are clear and the layout is consistent throughout the site.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
23. Hyperlinks on the website are easy to find by 'cursor change' or contrast with the texts around.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
24. Colour of hyperlinks changes after the link is visited. This makes me easy to know which links have been clicked.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
25. The labels of hyperlinks on the website are clear and meaningful. I can predict where I am going from reading the labels.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
26. Do you encounter any unexpected navigational behaviour, such as broken links?
□ Yes, more than once □ Yes, once □ No
27. The website provides the following navigational facilities (tick more if necessary)
□ search engine □ menu □ Navigational bars □ site map □ others
____
28. The website provides supportive mechanism for disabled people, such large-font for weak-sighted people.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree

Please open UMIST's homepage in the 'Bookmark', and answers the Question 29 - Question 40.
29. If you want to do MSc in Marketing or a related course at UMIST, give one course title offered by the University.

____

30. Now, are there any alternative paths for you to find the answer to Question 29?
□ alternative path easily found □ not easy, but still can find one □ hard to find
31. If an international student wants to do MBA at UMIST, what is the minimum TOEFL score for the language entry requirement?

---

32. What is the URL of the University's library homepage?

---

33. Who is the Student Advisor at Student Advisor Centre?

---

34. The navigational items are clear and the layout is consistent throughout the site.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
35. Hyperlinks on the website are easy to find by 'cursor change' or contrast with the texts around.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
36. Colour of hyperlinks changes after the link is visited. This makes me easy to know which links have been clicked.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
37. The labels of hyperlinks on the website are clear and meaningful. I can predict where I am going from reading the labels.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
38. Do you encounter any unexpected navigational behaviour, such as broken links?
□ Yes, more than once □ Yes, once □ No
39. The website provides the following navigational facilities (tick more if necessary)
□ search engine □ menu □ Navigational bars □ site map □ others

40. The website provides supportive mechanism for disabled people, such large-font for weak-sighted people.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
Please open University of Salford's homepage in the 'Bookmark'. Please find the answers to the Question 41 - Question 52.
41. If you are interested in Space Technology and want to do a related BSc course, what course you may choose? Please give one course title.

---

42. Now, are there any alternative paths for you to find the answer to Question 41?
□ alternative path easily found □ not easy, but still can find one □ hard to find
43. For an international student, who is studying BSc in Computing at the University, how much is the annual tuition fees?

---

44. What is the URL of the University's library homepage?

---

45. Please give the location of the Student Activities Office of Student Union (Name of the House).

---

46. The navigational items are clear and the layout is consistent throughout the site.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
47. Hyperlinks on the website are easy to find, by 'cursor change' or contrast with the texts around.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
48. Colour of hyperlinks changes after the link is visited. This makes me easy to know which links have been clicked.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
49. The labels of hyperlinks on the website are clear and meaningful. I can predict where I am going from reading the labels.
□ strongly agree □ agree □ undecided □ disagree □ strongly disagree
50. Do you encounter any unexpected navigational behaviour, such as broken links?

□ Yes, more than once □ Yes, once □ No

51. The website provides the following navigational facilities (tick more if necessary)

□ search engine □ menu □ Navigational bars □ site map □ others

______

52. The website provides supportive mechanism for disabled people, such large-font for weak-sighted people.

□ strongly agree □ agree □ undecided □ disagree □ strongly disagree

Section 4: Additional comments

Please use the rest of this page (or overleaf) to add comments that you have. Comments on the following would be particularly helpful:

1. Problems encountered when in your testing which are not included above
2. Your overall feeling about the navigability of the above four websites. For example, put 'good', 'average' or 'poor' for each website.
3. Any comments that you may have about this questionnaire