

目錄

目錄.....	I
圖目錄.....	III
表目錄.....	IV
Abstract	V
1. Introduction.....	6
1.1 Research Motivation	6
1.2 Research Problem	6
1.3 Research Approach	7
2. Literature Review.....	9
2.1 XML Query Capability	9
2.2 XML Benchmarks	10
2.2.1 XMark	10
2.2.2 XMach.....	10
2.2.3 XOO7.....	10
2.3 XML Benchmarks Comparison	11
2.4 Ontology	13
2.4.1 Ontology and Intelligent Information Integration.....	14
2.4.2 Ontology and Benchmark	14
3. Research Method	15
3.1 Research Approach	15
3.1.1 XML Data Model.....	17
3.1.2 XML Query Model	19
3.1.3 Exact Match	20
3.1.4 Joins	21
3.1.5 Regular Path Expressions	21
3.1.6 Document Construction	22
3.1.7 Ordered Access	23
3.1.8 Sorting.....	23
3.1.9 Missing Elements.....	24
3.1.10 Text Search.....	24
3.1.11 Data-type Cast.....	24
3.1.12 Function Application.....	25
3.2 Ontology Data Model	26
3.3 Ontology Query Model.....	27
3.4 Test Database Generation.....	29
3.5 Experimental Design.....	32

3.6 Prototype Functions	32
3.6.1 Data Loader.....	32
3.6.2 Query Generator.....	33
3.6.3 Scheduler.....	35
3.6.4 Result Collector	36
4. Research Implications and Concluding Remarks	37
4.1 Research Implications	37
4.2 Concluding Remarks.....	39
5. References.....	41
6. 計畫成果自評.....	43
附錄.....	44

圖目錄

Figure 1: Research Model.....	16
Figure 2: Graphic Representation of XML Data Model.....	18
Figure 3: XML Query Model.....	20
Figure 4: Ontology Query Model	28
Figure 5: Data Loader	33
Figure 6: Query Generator	34
Figure 7: Query Generator - Input	34
Figure 8: Query Generator – Select Standard Query Type	35
Figure 9: Query Generator – Set Standard Query Complexity.....	35
Figure 10: Result Collector.....	36

表目錄

Table 1: Comparison of Benchmarks over Workload Characteristics	12
Table 2: Comparison of Benchmarks over Query Functionalities.....	13
Table 3: Complexity Factors Listing.....	25

Abstract

Benchmarks are vital tools in the performance measurement and evaluation of computer hardware and software systems. Standard benchmarks such as the TREC, TPC, SPEC, SAP, Oracle, Microsoft, IBM, Wisconsin, AS³AP, OO1, OO7, XOO7 benchmarks have been used to assess the system performance. These benchmarks are domain-specific in that they model typical applications and tie to a problem domain. Test results from these benchmarks are estimates of possible system performance for certain pre-determined problem types. When the user domain differs from the standard problem domain or when the application workload is divergent from the standard workload, they do not provide an accurate way to measure the system performance of the user problem domain. System performance of the actual problem domain in terms of data and transactions may vary significantly from the standard benchmarks. In this research, we address the issue of domain boundness and workload boundness which results in the ir-representative and ir-reproducible performance reading. We tackle the issue by proposing a domain-independent and workload-independent benchmark method which is developed from the perspective of the user requirements. We present a user-driven workload model to develop a benchmark in a process of workload requirements representation, transformation, and generation. We aim to create a more generalized and precise

evaluation method which derives test suites from the actual user domain and application. The benchmark method comprises three main components. They are a high-level workload specification scheme, a translator of the scheme, and a set of generators to generate the test database and the test suite. The specification scheme is used to formalize the workload requirements. The translator is used to transform the specification. The generator is used to produce the test database and the test workload. In web search, the generic constructs are main common carriers we adopt to capture and compose the workload requirements. We determine the requirements via the analysis of literature study. In this study, we have conducted ten baseline experiments to validate the feasibility and validity of the benchmark method. An experimental prototype is built to execute these experiments. Experimental results demonstrate that the method is capable of modeling the standard benchmarks as well as more general benchmark requirements.

Keywords: XML, Ontology, Intelligent Information Integration, Generic Construct, Benchmark, Workload Model, Performance Measurement and Evaluation

1. Introduction

1.1 Research Motivation

A benchmark is a standard by which something can be measured or judged. A computer system benchmark is a set of executable instructions to be enforced in controlled experiments to compare two or more computer hardware and software systems. Hence, benchmarking is the process of evaluating different hardware systems or reviewing different software systems on the same or different hardware platforms. A web search service benchmark is therefore a standard set of executable instructions which are used to measure and compare the relative and quantitative performance of two or more systems through the execution of controlled experiments. Benchmark data such as throughput, jobs per time unit, response time, time per job unit, price and performance ratio, and other measures serve to predict price and performance and help us to procure systems, plan capacity, uncover bottlenecks, and govern information resources for various user, developer, and management groups (Can et al., 2004) (David et al., 2001) (Anon et al., 1985).

Examples are the TREC, TPC, SPEC, SAP, Oracle, Microsoft, IBM, Wisconsin, AS³AP, OO1, OO7, XOO7 standard benchmarks that have been used to assess the system performance. These benchmarks are domain-specific in that

they model typical applications and tie to a problem domain. Test results from these benchmarks are estimates of possible system performance for certain pre-determined problem types. When the user domain differs from the standard problem domain or when the application workload is divergent from the standard workload, they do not provide an accurate way to measure the system performance of the user problem domain. System performance of the actual problem domain in terms of data and transactions may vary significantly from the standard benchmarks. Performance measurement and evaluation is crucial in the development and advance of web search technology. A more open and generic benchmark method is needed to provide a more representative and reproducible workload model and performance profile (Jansen et al., 2006) (Richard 2006) (Vaughan 2004) (Kraaij et al., 2002).

1.2 Research Problem

Domain boundness and workload boundness are the research problem we try to tackle in this research. As described above, standard benchmarks model certain application types in a pre-determined problem domain. They represent a fixed problem set presented to the proposed system. When the user domain differs from the standard domain or when the user workload deviates from the standard

workload, the test results vary significantly in the real setting and under the actual application context. Users cannot reproduce the test results and predict the performance. The reason is because benchmark results are highly dependent upon the real workload and the actual application. The standard test workload cannot represent the real workload and the test suite cannot accommodate the application requirement. Standard benchmarks cannot measure the effects of the user problem on the target system nor generate the realistic and meaningful test results (Stephen 2002).

In this research, we address the issue by proposing a domain-independent and workload-independent benchmark method which is developed from the perspective of the user requirements. We propose to develop a more generalized and more precise performance evaluation method from the perspective of the common carriers of workload requirements. We create a user-driven approach which models the benchmark development in a process of workload requirements representation, transformation, and generation.

1.3 Research Approach

Benchmarks can be synthetic or empirical. Synthetic benchmarks model the typical applications in a problem domain and create the synthetic workload. Empirical benchmarks utilize the real data

and tests. Though real workloads are ideal tests, the costs of re-implementation of the actual systems usually outweigh the benefits obtained. Synthetic benchmarks are therefore the common approach chosen by developers and managers. Further, benchmark experiments are composed of the experimental factors and the performance metrics. Experimental factors represent the variables which can affect the performance of the systems. Performance metrics are the quantitative measurements to be collected and observed in the benchmark experiments. They represent the set of independent variables and dependent variables to be modeled and formulated in the benchmark.

A workload is the amount of work assigned to or performed by a worker or unit of workers in a given time period. The workload is the amount of work assigned to or performed by a system in a given period of time. The loads are best described by the amount of work, the rate at which the work is created, and the characteristics, distribution, and content of the work. Conventionally, workload modeling and characterization start with the domain survey, observation, and data collection, and continue with a study of the main components and their characteristics. In general, the workload components consist of the data, operations, and control.

In specific, workload analysis

involves the data analysis and the operation analysis. We analyze the size of the data, the number of records, the length of records, the types of attributes, the value distributions and correlations, the keys and indexing, the hit ratios, the selectivity factors. We investigate the complexity of operations, the correlation of operation, the data input into the operation, the attributes and objects used by the operation, the result size, and the output mode. These are further examined with the control analysis of the duration of test, the number of user, the order of test, the number of repetition, the frequency and distribution of test, and the performance metrics.

In the web search context, we develop a benchmark method that comprises a workload requirements specification scheme, a scheme translator, and a set of benchmark generators. We adopt the common carrier of generic constructs. We analyze the key web search algorithms and formulate the generic constructs. The generic constructs describe the page structure and the query structure of web search that is not tied to a per-determined search engine.

Workload Specification Scheme

The workload specification scheme is designed to model the application requirements. It is a high-level generic construct concept to describe

requirements concerning data, operation, and control. A generic construct is the basic unit of operand. An operation is the basic unit of operator. The collection of a generic construct and an operation formulate a workload unit. Each workload unit becomes a building block to compose a larger workload unit.

Scheme Translator

The scheme translator is created with a set of lexical rules and a set of syntactical rules to translate the workload specification. It performs the code generation and produces three output specifications. One is the data specification. The other is the operation specification. Another is the control specification.

Data Generator

The data generator is made up of a set of data generation procedures which are used to create the test database according to the data distribution specification.

Operation Generator

The operation generator is made up of a set of operation generation procedures to generate the search operations. These procedures select operations, determine operation precedence, schedule arrivals, prepare input data, issue tests, handle queues,

gather and report time statistics.

Control Generator

The control generator is made up of a set of control generation procedures to generate the control scripts which are used to drive and supervise the experiment execution.

2. Literature Review

2.1 XML Query Capability

Benchmarking the XML data management systems should consider many factors. Designing a set of comprehensive queries to test the XML databases' performance is an important point. XML query languages should capture the whole characteristics of a XML document, and the functionalities they provide would influence the query performance. The W3C XML Query Language working group (Chamberlin, Fankhauser, Marchiori, & Robie, 2003) list 20 XML query language "must have" functionalities, as Table shows. Some of the expected functionalities may affect the efficiency of the system significantly.

XQuery has met all of the requirements except F12 and F16, and it becomes a standard query language to test the performance of XML data management systems. Generally speaking, queries to benchmark XML

databases would fall into several categories: Match, Join, Navigation, Casting, Reconstruction, and Update. Queries for Match are mainly used to test the database ability to handle simple string lookups with a fully specified path. Join queries can be divided into two parts: Join on References, and Join on Values. References are an important part of XML, because they allow richer relationships than just hierarchical structure. Queries Join on References would test if query optimizer can take advantage of references to be joined. Queries Join on Values, on the other hand, would test the database's ability to handle large intermediate results. Differing from the former, their joins are on the basis of values. Navigation Queries investigate how well the query processor can optimize path expressions, and avoid traversing irrelevant parts of the tree. Strings are the basic data type in XML documents. Casting strings to another data type that carries more semantics is necessary. Queries for Casting challenge the ability of the database to cast different data types. Reconstruction Queries attempt to reconstruct the original document from its fragmentations stored in the databases. Update Queries try to add, delete, and modify elements in the XML document. These queries test the databases' ability to manage XML document. Furthermore, other XML query functionalities such as sort, ordered access, text search, and

aggregation also should be captured in the benchmark query set.

2.2 XML Benchmarks

XMark, XMach-1 and XOO7 are three benchmarks available today that can be used to evaluate certain aspects of XML database systems.

2.2.1 XMark

XMark (Schmidt, Waas, Kersten, Carey, Manolescu, & Busse, 2002) is a single-user benchmark. The data model of XMark is an Internet auction site. Therefore, its database contains one big XML document with text and non-text data. XMark enriches the references in the data, like the item IDREF in an auction element and the item's ID in an item element. The text data used are the 17000 most frequently occurring words of Shakespeare's plays. The standard data size is 100MB with a scaling factor 1.0 and users can change the data size by 10 times from the standard data (the initial data) each time. However, it has no support for XML Schema. In operation model of XMark, 20 XQuery challenges are designed to cover the essentials of XML query processing, as Table shows. No update operations are specified in XMark.

2.2.2 XMach-1

XMach-1 (Böhme & Rahm, 2001)

is a scalable multi-user benchmark. The main objective of the benchmark is to stress-test XML systems under a multi-user workload. The data model of XMach-1 is designed for B2B applications and considers text documents and catalog data. It assumes that size of the data files exchanged will be small. It provides support for DTD only and does not consider XML Schema for optimization. The operation model of XMach-1 consists of eight queries and three update operations, shown in Table.

Queries specified in XMach-1 cover typical database functionality (join, aggregation, sort) as well as information retrieval and XML-specific features (document assembly, navigation, element access). Update operations cover inserting and deleting of documents as well as changing attribute values. We find that some queries contain several query functionalities. It is hard to analyze the experiment result and ascertain which feature leads to the given performance result. Specially, XMach-1 has defined three update operations that are unique across other XML benchmarks.

2.2.3 XOO7

XOO7 (Li, Bressan, Dobbie, Lacroix, Lee, Nambiar, & Wadhwa, 2001) is an XML version of the OO7 benchmark, which was designed to test

the efficiency of object-oriented DBMS. XOO7 is a single-user based benchmark for XMLMS that focuses on the query processing aspect of XML. The data model of XOO7 comes from the OO7 benchmark by mapping the OO7 schema and data set to XML. No specific application domain is modeled by the data of XOO7. It is based on a generic description of complex objects using component-of relationships. XOO7 also proposes three different databases of varying size: small, medium, and large. It supports DTD only. In operation model, XOO7 provides relational, document and navigational queries that are specific and critical for XML database applications. These queries test the primitive features and each query

covers only a few features. Table displays the queries adopted in XOO7. XOO7 contains large amount of queries, each query covers only a few features. Comparing to the other two benchmarks, XOO7 has certainly the highest ratio which stresses its data-centric focus. However, we can find that some queries are focus on the same functionality. Similar to XMark, no update operation is specified in XOO7.

2.3 XML Benchmarks Comparison

A comparison of key features of these main XML benchmarks against this research is described in Table 1. The key features include application focus, evaluation scope, database and workload characteristics.

Table 1: Comparison of Benchmarks over Workload Characteristics

Feature	XMark	XMach-1	XOO7	This Research
Evaluation Scope	Query Processor	DBMS	Query Processor	Heterogeneous Information Integration
Application Domain	E-Commerce	E-Commerce	Generic	Generic
Data Model				
Documents	Single-document	Multi-documents	Multi-documents	Multi-documents
Scalability of Document Number	1	$10^4 \sim 10^7$	Unlimited	Various
Scalability of Document Size	10MB~10GB	16KB	Unknown	Various
Data Heterogeneity	XML document only	XML document only	XML document only	Heterogeneous data sources
Nodes/KB	18	10	67	Various
Operation Model				
Queries	20	8	23	14
Update Operation	0	3	0	0

Table 2 groups queries of each benchmark by query functionality. Compared to other XML benchmarks, XMark provides a concise and comprehensive set of queries. However, it does not provide update operations to manipulate XML documents. XMach-1 only defines a small number of XML queries that cover multiple functions and update operations for which system performance is determined. XOO7 maps the original queries of OO7 into XML, and adds some XML specific queries. In general, XMach-1, XMark, and XOO7

cover only a subset of the XML query requirements. In this research, we attempt to propose a generic workload model. In order to cover the whole functionalities of XML query processing, we combine queries of these three XML benchmarks and integrate them into ten types of queries. In particular, the intelligent information integration system is generally used for query data, not provide data manipulation functions. Therefore, the query model in this research does not support update operations.

Table 2: Comparison of Benchmarks over Query Functionalities

Query Functionality		XMark	XMach-1	XOO7	This Research
Exact Match		✓	✓	✓	✓
Joins	Join on Reference	✓	✓	✓	✓
	Join on Value	✓			✓
Regular Path Expressions	Full Sub-path	✓	✓		✓
	Unknown Sub-path	✓	✓		✓
Document Construction	Structure Preserving	✓		✓	✓
	Structure Transforming	✓		✓	✓
Ordered Access		✓	✓	✓	✓
Sorting	By String	✓	✓		✓
	By Non-string			✓	✓
Missing Elements		✓			✓
Text Search		✓	✓	✓	✓
Data-type Cast		✓			✓
Function Application		✓	✓	✓	✓
Update Operation			✓		

2.4 Ontology

Ontologies play an important role for integration as a way of formally defined terms for communication. They aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused, shared, and operationalized across applications and groups.

A good ontology should represent the domain specific knowledge explicitly. The question is how do we

know an ontology is good? The answer is the ontology benchmark. There are plenty of benchmark studies in other fields like database or compilers. However, there are no specific benchmarks studies or tools for evaluating ontology-based applications. In fact, there is still no guideline to evaluate ontologies and related technologies.

In this section, we introduce the role of ontologies in intelligent information integration first. And then we discuss a major inference task which

is the main operation of an ontology benchmark. Finally, the ontology related benchmark works are reviewed.

2.4.1 Ontology and Intelligent Information Integration

Traditional integration approaches use inexpressive models of database schemas or XML trees to integrate heterogeneous data sources. This would cause many semantic heterogeneity problems. Ontologies provide much richer modeling means with classes and properties organized into is-a hierarchy and enriched with axioms and relations processable with inference (Maier, Aguado, Bernaras, Laresgoiti, Pedinaci, Pena, & Smithers, 2003). Almost all ontology-based integration approaches ontologies are used for the explicit description of the information source semantics. With respect to the integration of data sources, they can be used for the identification and association of semantically corresponding information concepts. Some approaches use ontologies not only for content explication, but also either as a global query model or for the verification of the (user-defined or system-generated) integration description (Wache, Vögele, Visser, Stuckenschmidt, Schuster, Neumann, & Hübner, 2001). Ontologies are usually expressed in a logic-based language, so that fine, accurate, consistent, sound, and meaningful distinctions can be made

among the classes, properties, and relations. Therefore, ontologies not only have the expressiveness needed in order to model the data in the sources, but their reasoning ability can help in the selection of the sources that are relevant for a query of interest, as well as to specify the extraction process.

2.4.2 Ontology and Benchmark

To the best of our knowledge, the benchmark presented here is the first one for ontology-based intelligent information integration. The ontology benchmark model in this research differs from database benchmarks, such as Wisconsin benchmark, OO7 benchmark, and BUCKY benchmark. They are all DBMS-oriented and storage benchmarks, and there is no inference ability included. In this research, the ontology workload model is applied to an intelligent information integration system, and we focus on the inference ability of the ontology. Ontology and XML are often found together and are often confused. XML is a standard for marking up - adding additional information, called metadata - to documents. The purpose of XML is to tag textual information with additional structure that enables it to be “understood” and exchanged by programs. However, XML tags still require humans to interpret their meanings. Therefore, XML benchmarks only focus on structural and syntactic evaluation of systems, and they have no

semantics. On the other hand, ontology benchmark is devoted to capture the semantic expressions in the system. Thus, ontology and XML are complementary technologies: ontology provides the meaning for XML standards; XML provides a valuable medium for information exchange between programs that share the same ontology.

As mentioned above, there is still no guideline for evaluation of ontology-based application. Horrocks and Patel-Schneider (1998) benchmark description logic systems, or so-called knowledge bases. Description logics (DLs) are a family of knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formally well-understood way. Description logic systems provide their users with various inference capabilities that deduce implicit knowledge from the explicitly represented knowledge. Horrocks and Patel-Schneider try to evaluate the reasoning algorithms in description logics. Terminological part (Tbox) is a set of axioms describing the structure of domain. Assertional part (Abox) is a set of axioms describing concrete situation (Horrocks, 2002). They are related to this research. In an intelligent information integration system, the ontology can be viewed as

the Tbox, and the heterogeneous data can be viewed as Abox. However, the logic described is only a subset of the ontology languages, such as DAML+OIL and OWL. DAML+OIL and OWL can be seen to be equivalent to a very expressive description logic. They provide more constructors and allow more axioms than description logic. Therefore, the inference services of ontology are more complex than traditional description logic systems.

3. Research Method

3.1 Research Approach

The literatures studied would help us to identify the important performance factors for XML and ontology processing. We analyze the XML-specific and ontology-specific requirements in more details to justify the design of the benchmark. The research structure is shown in Figure 1. The benchmark study consists of two benchmark workload models, the XML benchmark workload model and the ontology benchmark workload model. Both of them consist of the data model and query model according to the generic constructs and constraints requirements. Next, the control model is created before the generic workload model to be generated and executed so as to measure and evaluate the systems.

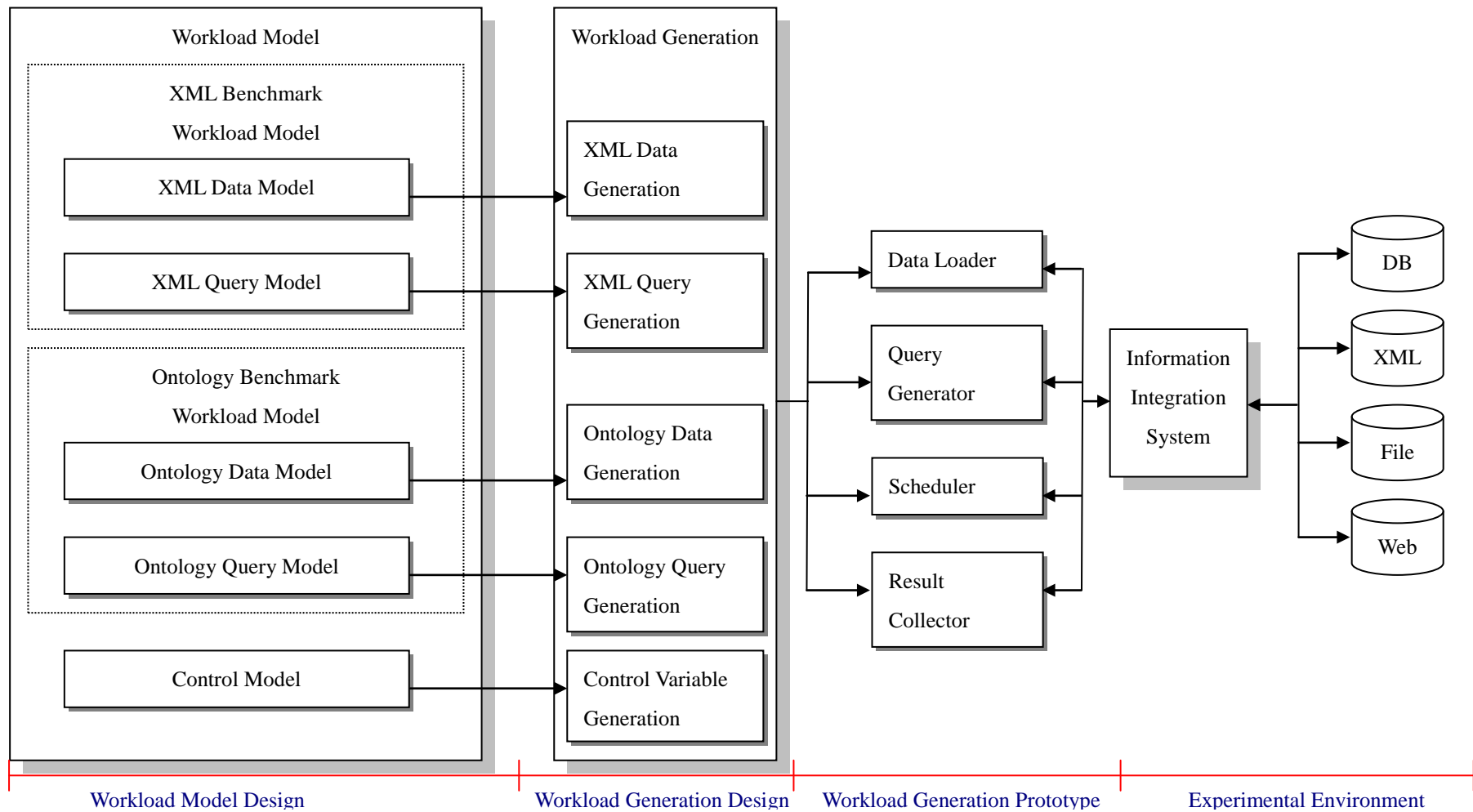


Figure 1: Research Model

In this research, we focus on intelligent information sources integrated in XML and ontology. The benchmark model we propose would capture most features of the released XML-based and ontology-based specifications.

Developing a benchmark requires the definition of the test workload model first. In this research, we provide a benchmark workload model that combines XML and ontology in intelligent information integration. In XML workload model, the data model describes a generic XML data model and the operation model defines a comprehensive set of test queries that covers the major aspects of XML query processing. In ontology workload model, the data model describes the major ontology component, and the operation model defines some important criteria to query the ontology. The control model defines the variables that used to set up the benchmark environment.

3.1.1 XML Data Model

XML is a hierarchical data format for information exchange on the Web. An XML document consists of nested elements that contain data or other elements. The boundaries of these elements are either delimited by start-tags and end-tags, or, for empty elements, by empty-element tags. The text between start-tags and end-tags is

the content of the element. Each element has a type, identified by name, sometimes called its “generic identifier” (GI), and may have a set of attribute specifications. Each attribute specification has a name and a value (Bray, Paoli, Sperberg-McQueen, & Maler, 2000). XML documents may comply with a Document Type Definition (DTD) or a XML Schema. DTD has traditionally been the most common method for describing the structure of XML document. But DTD lacks enough expressive power to properly describe highly structured data. XML Schemas are an XML language for describing and constraining the content of XML documents. It provides a richer and more powerful means for defining the data. Therefore, XML schema becomes the most common method for defining and validating highly structured XML documents rapidly.

In the XQuery and XPath data model, XML documents are modeled as an ordered tree. The tree contains seven distinct kinds of nodes: document, element, attribute, text, namespace, processing instruction, and comment. In this research, for simplicity, we only consider document, element, attribute, and text nodes. The data model is a node-labeled, directed graph, in which each node has a unique identity shown in Figure 2. Document order is defined for all the nodes in the document and corresponds to the order in which the

first character of each node occurs in the XML document.

- **Document nodes:** The document node is a virtual node pointing to the root element of an XML document. The document element in a XML document is a child of the document node.
- **Element nodes:** Every element in the document is an element node. Element nodes have zero or more children that can be element nodes or text nodes.
- **Attribute nodes:** Each element node has an associated set of attribute nodes. Note that the element node that owns this attribute is called its “parent” even though an attribute node is not a “child” of its parent element. An attribute node has an attribute name and an attribute value. Attribute nodes have no child nodes. If more than one attribute of an element node exists, the document order among the attributes is not distinguished. This is because there is no order among XML attributes.
- **Text nodes:** A text node must have only one parent and have no child nodes. A text node cannot contain an empty string as its content.

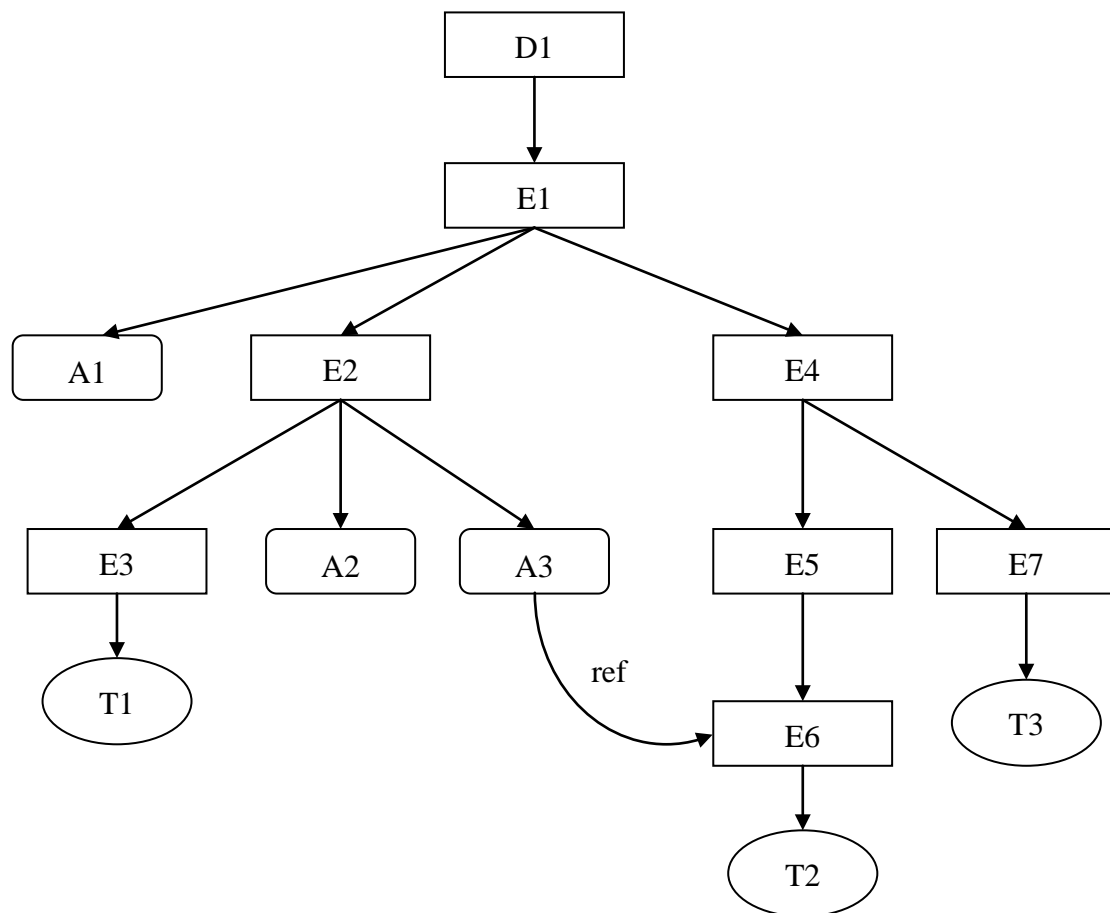


Figure 2: Graphic Representation of XML Data Model

3.1.2 XML Query Model

The query model is specified in generic constructs. We further identify key factors that influence the complexity of each query. This would help users to evaluate performance of the system with increasing complex queries.

The query model we defined can be classified into ten categories. Each of them challenges different aspects of XML processing. Besides, users can specify queries according to their

requirements which is called “user-driven query”. Figure 3 shows the XML query model. The following will describe each category briefly, and express each query in generic constructs. In each query, the generic term is written in italics. Then we illustrate them in XQuery. We use *E1*, *E2* etc. to denote a certain element, and *A1*, *A2* etc. to denote a certain attribute. The number of them does not indicate their order in a XML document, just for representing convenience. Finally, the complexity factors will be discussed.

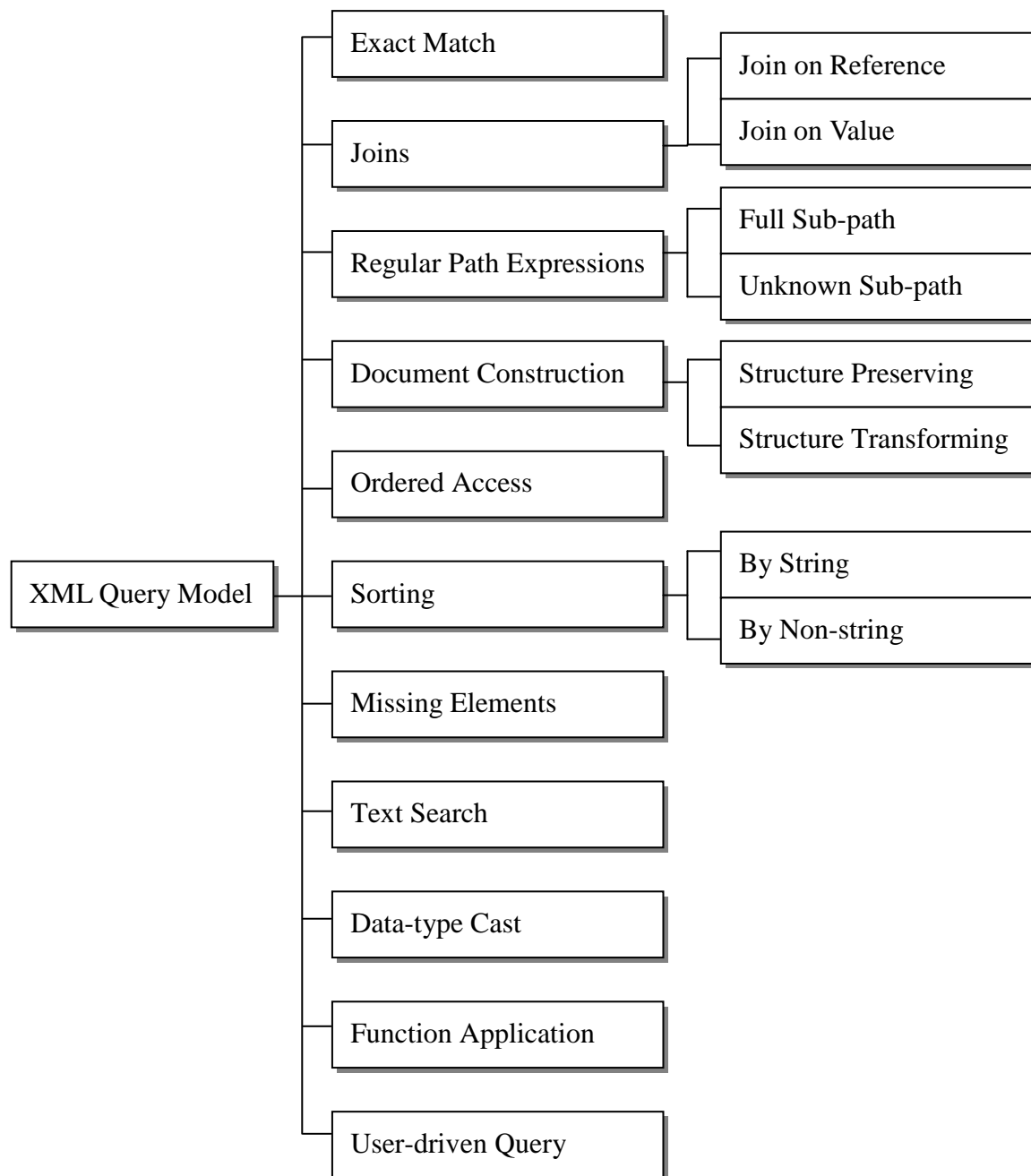


Figure 3: XML Query Model

3.1.3 Exact Match

This type of queries specifies a full path expression. One main concept of XQuery is the use of path expressions for selecting nodes. The length of the path expression depends on the levels of predicates being queried in XML

documents. This is the simplest query type. We can use this type of queries to establish a simple “metric” comparing performance of the following queries. It tests the database ability to handle simple string lookups with a fully specified path.

Generic constructs are: *Given a full path expression, find elements E1 that have an attribute A1 in a certain value X.* XQuery expression is:

```
FOR      $a IN input()/SUBPATH/E1[@A1 = "X"]
RETURN  $a
```

3.1.4 Joins

References are an integral part of XML identifying the relationship between related data. With using of reference, richer relationships can be represented than just hierarchical element structures. The system must be able to combine separate information together using joins. Horizontal

traversals are defined in this type of queries. Joins can be on the basis of references and values. References are specified in the DTD and may be optimized with logical OIDs for example. The system should make use of the cardinalities of the sets to be joined. Joins based on values test the database's ability to handle large (intermediate) results.

Join on Reference

Generic constructs are: *Find element E1 by the reference attribute A1 of E2. The*

reference attribute A1 of E2 refer to E1. XQuery expression is:

```
FOR      $a IN input()/E1
         $b IN input()/E2
WHERE    $a/@A2 = $b/@A1
RETURN  $a
```

Join on Value

Generic constructs are: *This time reference is based on join of the data*

values. Find element E1 whose attribute A1 is equal to the attribute A2 of E2. XQuery expression is:

```
FOR      $a IN input()/E1
         $b IN input()/E2
WHERE    $a/@A1 = $b/@A2
RETURN  $a
```

3.1.5 Regular Path Expressions

Regular path expressions are a

basic building block of almost every XML language including XPath, XQuery, and XSLT. The system should

be capable of optimizing path expressions and reducing traversals of irrelevant parts of the tree. We often use wildcards in regular path expressions and the system should realize that it is not necessary to traverse the complete

document tree to execute such expressions. This type of queries tries to quantify the costs of long path traversals that do not include wildcards, and the costs of path traversals that include wildcards.

Full Sub-path

Generic constructs are: *Find element E1*

with a long path expression. XQuery expression is:

```
FOR      $a IN input()/SUBPATH/E1
RETURN  $a
```

Unknown Sub-path

Generic constructs are: *Find element E1*

with a regular path expression include wildcards. XQuery expression is:

```
FOR      $a IN input()//E1
RETURN  $a
```

3.1.6 Document Construction

Structure is very important to XML documents. But XML documents storing in relational DBMSs often need to be broken down. Reconstructing the original document is a big challenge to

systems. We might retrieve fragments of original documents with original structures. But sometimes we may want to construct document fragments with new structures. These queries tests for the ability of the system to reconstruct portions of the original XML document.

Structure Preserving

Generic constructs are: *Return a XML document constructed by element E1*

and its sub-element E2. Retrieve E2 of E1 that has an attribute A1 equal to a certain value X. XQuery expression is:

```
FOR      $a IN input()//E1[@A1 = X]
RETURN  <$a> $a/E2 </$a>
```

Structure Transforming

Generic constructs are: *Construct a new XML document. Find element E1 with an attribute A1 equal to a certain value*

X, and select several sub-element of E1 to construct a new XML document. XQuery expression is:

```

FOR      $a IN input()//E1[@A1 = X]
RETURN  <output>
        {$a/E2/E3}
        {$a/E2/E4}
        {$a/E2/E3/E5}
        {$a/E6}
</output>

```

3.1.7 Ordered Access

Order of elements is important in XML documents. Because documents will sometimes be fragmented when they are stored on disk, it is important

that the order of these fragments in the original document is preserved. The system should be able to preserve these intrinsic orders. This type of queries attempts to test how efficient the system handle queries with order constraints.

Generic constructs are: *Find element E1 with attribute A1 in certain value X, and*

return the first sub-element E2 of E1. XQuery expression is:

```

FOR      $a IN input()//E1[@A1 = X]
RETURN  $a/E2[1]

```

3.1.8 Sorting

The order by clause is the only facility provided by XQuery for specifying an order other than document order. In XML documents, the generic data type of element content is string,

but users may cast the string type to other types. Therefore, the system should be able to sort values both in string and in non-string data types. This type of queries tests whether the system can do sorting efficiently.

By String

Generic constructs are: *List sub-element*

E3 of element E1 sorted by sub-element E2. XQuery expression is:

```

FOR      $a IN input()//E1
ORDER BY $a/E2
RETURN  $a/E3

```

By Non-string

Generic constructs are: *List sub-element E3 of element E1 sorted by sub-element*

E2. This time E2 is a non-string value.

XQuery expression is:

```
FOR      $a IN input()//E1
ORDER BY $a//E2
RETURN   $a/E3
```

3.1.9 Missing Elements

In XML, schemas are more flexible and may have a number of irregularities. Queries in this type are to test how well

the system knows to deal with the semi-structured aspect of XML data, especially elements that are declared optional in the schemas.

Generic constructs are: *Find element E1 whose sub-element E2 has NULL value.*

XQuery expression is:

```
FOR      $a IN input()//E1
WHERE    EMPTY($a/E2/text())
RETURN   $a
```

3.1.10 Text Search

Text search plays a very important part in XML document systems. This

type of queries conducts a full-text search in the form of keyword search. They will challenge the textual nature of XML documents.

Generic constructs are: *Find element E1 whose sub-element E2 contains a*

specific text Y. XQuery expression is:

```
FOR      $a IN input()//E1
WHERE    CONTAINS ($a/E2, "Y")
RETURN   $a
```

3.1.11 Data-type Cast

Strings are the generic data type in XML documents. But we often need to

cast strings to another data type that carries more semantics. These queries challenge the system's ability to transform between data types.

Generic constructs are: *Find element E1 with a constraint that contain operations need to transform data value of sub-element E2 to other data-type.*

Retrieve element E1 whose sub-element E2 is bigger than a certain number X.
XQuery expression is:

```
FOR      $a IN input()//E1
WHERE    $a/E2 > X
RETURN   $a
```

3.1.12 Function Application

constructs are: *Group element E1 by sub-element E2, and calculate the total number of elements for each group.*
XQuery expression is:

The following query challenges the system with aggregate functions such as count, avg, max, min and sum. Generic

```
FOR      $a IN DISTINCT-VALUES (input()//E1/E2)
LET      $b = input()//E1[E2 = $a]
RETURN   count($b)
```

Table 3 summarizes the complexity factors of each query type.

Table 3: Complexity Factors Listing

Complexity	Low	High
Exact Match	Shallow Path Expression	Deep Path Expression
Joins	Two-way Join Small Result Size	N-way Join Large Result Size
Regular Path Expressions	Shallow Path Expression Few Unknown Elements	Deep Path Expression Many Unknown Elements
Document Construction	Simple Original Structure Simple Output Structure	Complex Original Structure Complex Output Structure
Ordered Access	No Index	With Index
Sorting	Few Qualified Tuples Single Condition	Many Qualified Tuples Multiple Condition
Missing Elements	Few Generated Tuples	Many Generated Tuples
Text Search	One Text Few Generated Tuples	Multiple Text Many Generated Tuples
Data-type Cast	Few Generated Tuples	Many Generated Tuples

	Single Casting	Multiple Casting
Function Application	Few Generated Tuples	Many Generated Tuples

3.2 Ontology Data Model

An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information. Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms (Ushold, King, Moralee, & Zorgios, 1998). Generally speaking, an ontology consists of the following main constructs (Stevens, Goble, & Bechhofer, 2000; Weißenberg & Gartmann, 2003).

- **Facts** represent explicit knowledge, consisting of:
 - **Classes or concepts** are generalizations of instances. Concepts are the focus of most ontologies. A concept is a representation for a conceptual grouping of similar terms. A concept can have subconcepts that represent concepts that are

more specific than the superconcept. Concepts fall into two kinds of (1) primitive concepts are those which only have necessary conditions (in terms of their properties) for membership of the class. (2) Defined concepts are those whose description is both necessary and sufficient for a thing to be a member of the class.

- **Properties** can be subdivided into scalar attributes and non-scalar relations. The property can be defined to be a specialization (subproperty) of an existing property. An attribute is a property of a concept that refers to a datatype (integer, string, float, boolean etc.). An example of an attribute is “has-name” related to a string. A relation is a property of a concept that refers to another concept. Specialization / Generalization are one of the standard relations. For instance, “is a kind of” defines a relation that may be applied to the concepts “Enzyme” and “Protein”.
- **Instances** represent individual entities and are connected by type-of relation to at least one

class; some authors only consider facts about instances as real facts. Strictly speaking, an ontology should not contain any instances, because it is supposed to be a conceptualization of the domain. The combination of an ontology with associated instances is what is known as a knowledge base. However, deciding whether something is a concept of an instance is difficult, and often depends on the application.

- **Axioms** are rules used to add semantics and to infer knowledge from facts. In contrast to facts, they represent implicit knowledge about concepts and relations, e.g., whether a relation is transitive or symmetric.

3.3 Ontology Query Model

Initially, ontologies are introduced as an “explicit specification of a conceptualization”. In an intelligent information integration system,

ontologies can be used to establish common vocabularies and semantic interpretations of terms from information sources. With respect to the integration of data sources, they can be used for the identification and association of semantically corresponding information concepts. People can share and exchange information in a semantically consistent way.

Using ontology basic components described in the previous section, users can define their own ontology in any application domain. Then we conduct a series of tests to see how the system handles such ontologies. The operation model in the ontology workload model is a set of queries, and the answers are generated by inferring from the ontology. The queries we present here are representative for different application domains. We conclude the reasoning tasks and construct six basic reasoning queries for the ontology benchmark. Figure 4 shows the ontology query model in this research.

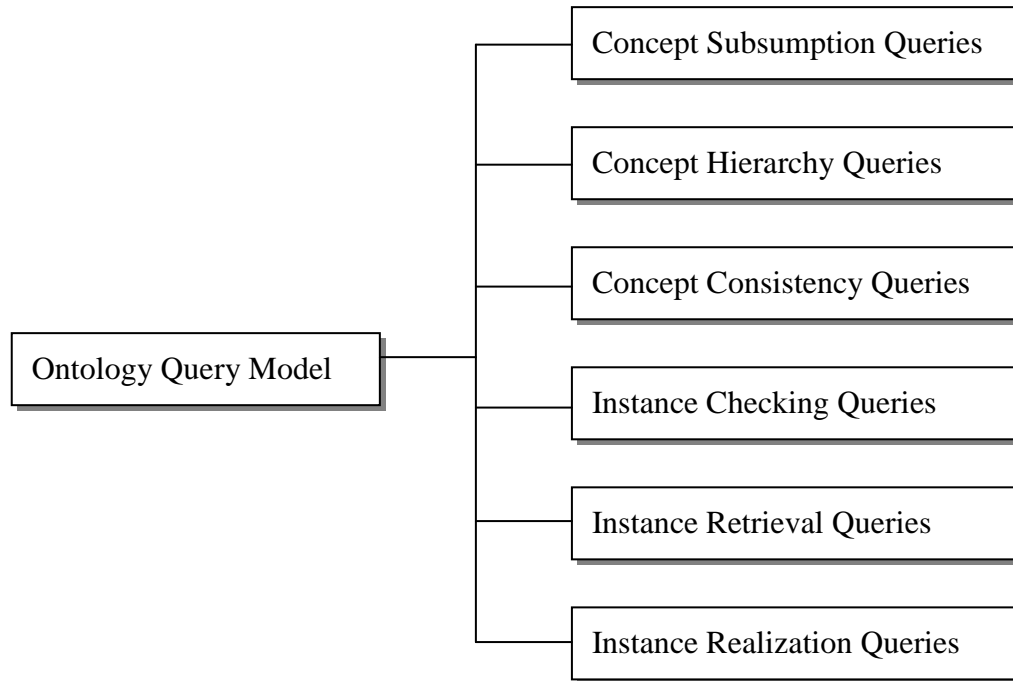


Figure 4: Ontology Query Model

- **Concept Subsumption Queries:** checks if one concept is a subconcept of another. Generic constructs are: *Given concepts C and D , determine if C is a subconcept of D with respect to ontology O .*
- **Concept Hierarchy Queries:** determines the concepts that immediate subsume or are subsumed by a given concept. Generic constructs are: *Given a concept C return all/most-specific superconcepts of C and/or all/most-general subconcepts of C .*
- **Concept Consistency Queries:** checks for (in)consistency of concept definitions. Generic constructs are: *Given a concept C , determine if the definition of C is generally satisfiable (consistent).*
- **Instance Checking Queries:** given a partial description of an individual (instance) and a concept description, finds whether the concept describes the instance. Generic constructs are: *Given a concept C , determine whether a given individual A is an instance of C .*
- **Instance Retrieval Queries:** finds all instances that are described by a given concept. Generic constructs are: *Given a concept C , determine all the individuals in ontology O that are instances of C .*
- **Instance Realization Queries:** given a partial description of an instance, finds the most specific concepts that describe it. Generic constructs are: *Given an individual A , determine all the concepts in ontology O that A is an instance of.*

When querying the intelligent information integration system, the reasoning service may not be so straightforward. We need to evaluate the ontology with increasing complexity. When formulating the complex benchmark queries, several factors should be taken into account (Guo, Heflin, & Pan, 2003). More complex queries may be formulated according to these factors. These allow the system be evaluated under different level of workloads.

- **Input size:** This is measured as the proportion of the class instances involved in the query to the total class instances in the benchmark data.
- **Selectivity:** This is measured as the estimated proportion of the class instances involved in the query that satisfy the query criteria.
- **Complexity:** We use the number of classes and properties that are involved in the query as an indication of complexity.
- **Hierarchy information assumed:** This considers whether information of class hierarchy or property hierarchy is required to achieve the complete answer. Besides, the depth and width of class hierarchies should also be considered.

3.4 Test Database Generation

In order to evaluate the performance of the intelligent information integration system, we must define the test database. The workload consists of a test operation and a test database. The test database identifies what data must be loaded into the data sources, as well as the volume of the test data. Information integration system data sources are disparate and heterogeneous. Information comes from various sources (including structured, semi-structured and unstructured sources) and formats (such as database tables, XML files, PDF files, streaming media, internal documents, and Web pages). For this research, the data sources can be divided into three kinds: relational databases, object-oriented databases, and Web pages. For each data source, we must analyze the actual data and extract statistical data. Data analysis characterizes data in terms of the size of the database, the number of records, the length of records, the types of fields, and the value distributions.

- **Determine data values:** A number of data types are supported in this research, including long integer number, double precision floating point number, decimal number, money, datetime, fixed-length and variable-length character strings. We must conduct extensive studies to characterize each data source with several distribution parameters. Frequency distributions are

computed and standard probability distributions are fit to the data in order to generate the value of test data. Data values are created with these common data distributions such as exponential, normal, discrete, rotating, zipfian² or uniform distribution.

- **Determine scaling factors:** After determining the value of the test data, we must define how much data should be generated, i.e. defining the database scaling factor. Generally speaking, the logical size for the test database used for the benchmark is at least equal to the logical size of physical memory on the host(s). For this research, we refer to the AS³AP benchmark standard.
- **Open data source:** We must determine the test data of the open data source on the Web, but this is problematic. There is in excess of 10 billion pages on the Web, which include HTML files, text documents, PDF files, Microsoft Office documents and other similar data files. We cannot possibly download every page from the Web much less adequate sample size. Even the most comprehensive search engine currently indexes just a small fraction of the entire Web.

As such, it is important to carefully select the so-called “important” pages, so that the fraction of the Web that is

visited becomes more meaningful. In order to select these important pages, we can use several metrics for prioritizing them. For any given web page, we must define its importance using the following methods (Arasu, Cho, Garcia-Molina, Paepcke, & Raghavan, 2001):

- **Interest-driven.** The goal is to obtain pages of interest to a particular user or set of users. Important pages are those that match user interest. One particular way to define this notion is through what we call a *driving* query. For any given query, the importance of a page is defined by the “textual similarity” between the page and the driving query. Assuming that query represents the user’s interest, this metric shows how relevant the page is. Another interest-driven approach is based on a hierarchy of topics. Interest is defined by a topic, and we attempt to guess the page topics that will be visited by analyzing the link structure that leads to the candidate pages.
- **Popularity-driven.** Page importance depends on how popular a page is. For instance, one way to define popularity is to use a page’s backlink count. Intuitively, a page that is linked to by many pages is more important than one that is seldom referenced.
- **Location-driven.** The importance

of a page is a function of its location, not its contents. For example, URLs ending with “.com” may be deemed more useful than URLs with other endings, or URLs containing the string “home” may be of more interest than other URLs. Another location metric that is sometimes used considers URLs with fewer slashes more useful than those with more slashes.

(3) Control Model

The control model defines the environment setup variables to execute the experiment. A common set of parameters including the steady state, the test mode, the test duration, the test sequence and the number of repetitions should be specified as follows.

- **Steady State:** The benchmark test must be executed in a steady state, in order to return the sustained system performance.
- **Test Mode:** There are three kinds of test mode, that is, cold mode, warm mode, and hot mode. In cold mode, there is no data in the cache. The system cannot retrieve data from the cache directly. Therefore, the performance in cold mode is usually slower than the other two modes. In warm mode, the data is left in the cache from the prior query. Because of that, the test response time decreases. In hot mode, a query is

executed in cold mode first, and then is executed with the cache data for several times. The average response time is computed.

- **Test Duration:** Test duration means time intervals of the benchmark. Each interval must begin after the system has reached steady state and is long enough to generate the reproducible throughput. Each interval must extend uninterruptedly for a period of time.
- **Test Sequence:** Test sequence indicates the order of the queries to be executed.
- **Number of Repetitions:** Number of repetitions means execution repeated times.

(4) Performance Metrics

Performance metrics can be divided into two types, i.e., the speed-specific metrics and the relevance-specific metrics. The former consists of the metrics of response time and throughput. The latter has the metrics of relative recall and precision.

- **Response time:** Response time refers to the time interval between when a request is made and when the response is received by the requester.
- **Throughput:** Throughput refers to the number of operations completed by the system per unit time.
- **Recall and precision:** Recall and

precision refer to two important measures of evaluation of information retrieval. However, it is very difficult, if not impossible, to directly apply these measurements to the evaluation of Web information retrieval systems due to the unique nature of the Web. There is no proper method of calculating absolute recall of search engines as it is impossible to know the total number of relevant in huge databases. The relative recall value is defined in (Clarke & Willet, 1997).

3.5 Experimental Design

We use the university campus as the test scenario of the benchmark prototype. It describes universities and departments and the activities that occur at them. The global schema for the intelligent information integration prototype system is shown on Appendix

A. For the Tamino XML server, we have defined six schemas: Department, Faculty, Student, Grad Student, Course, and Publication. These schema details are shown on Appendix C.

3.6 Prototype Functions

3.6.1 Data Loader

In the data loader, users can define the logical database size. The default logical database sizes for each data source are uniform and users can modify them individually. The data sources and data schema in the prototype are described in the previous section. Furthermore, in the SQL Server users can adjust the number of tuples loaded into each relation. Similarly, users can alter the number of instances generated for each schema in Tamino XML server as shown in Figure 5.

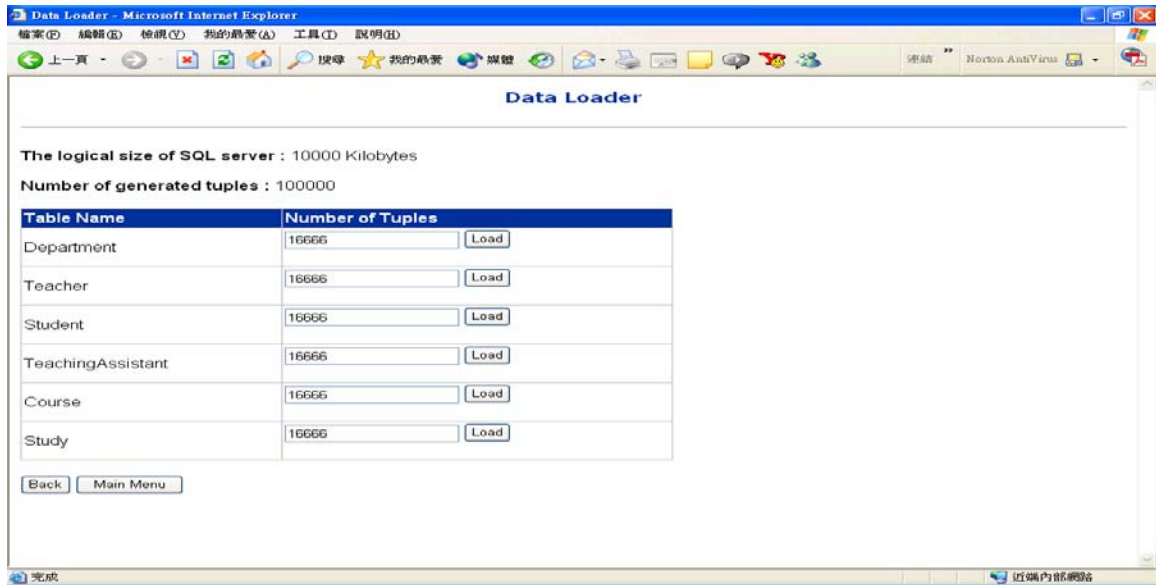


Figure 5: Data Loader

3.6.2 Query Generator

The query generator can be divided into two parts: pure XML query and XML query combined ontology reasoning as shown in Figure 6. Users can evaluate pure XML processing performance of intelligent information integration systems or combined ontology reasoning services. In each of them, users can input the queries they want to test or simply select the standard query types predefined in the query selector. The queries are generated according to the XML query model. In

open query input, users can specify several queries according to global schema and their requirements. All specified queries will be executed in one test, and users can delete any one in the open query list as shown in Figure 7-Figure 9. In the standard query selector, after choosing the standard test queries, users can specify the complexity of each query further. It would help users to evaluate the system performance at different complexity levels. The complexity of query is determined by the complexity factor.

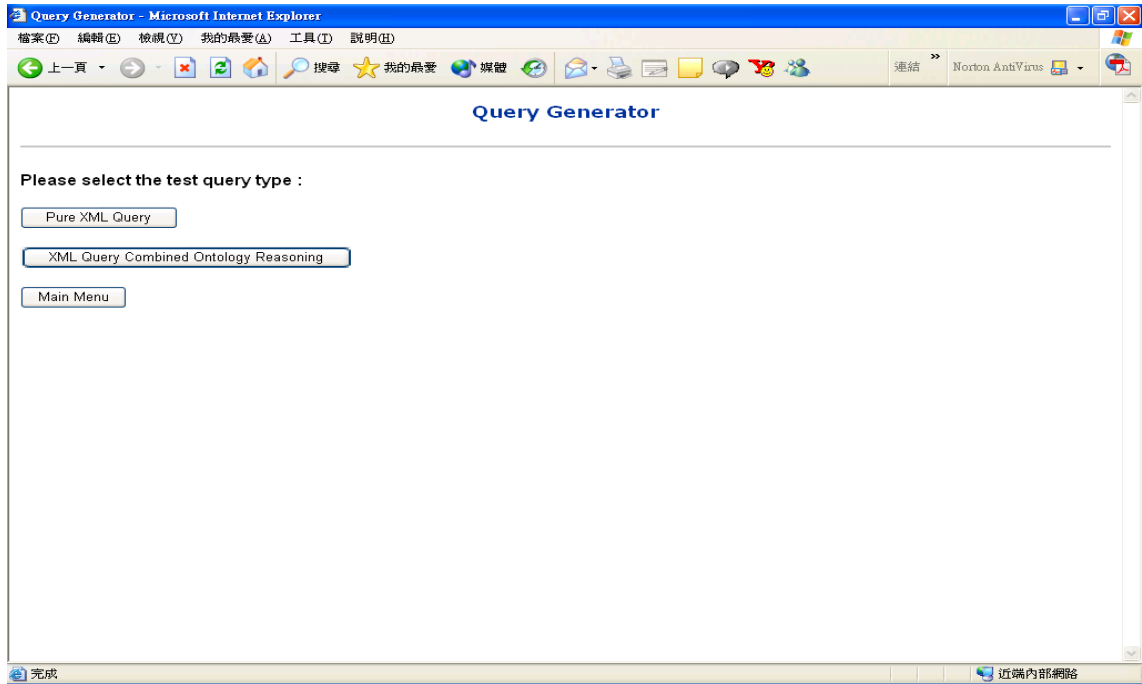


Figure 6: Query Generator

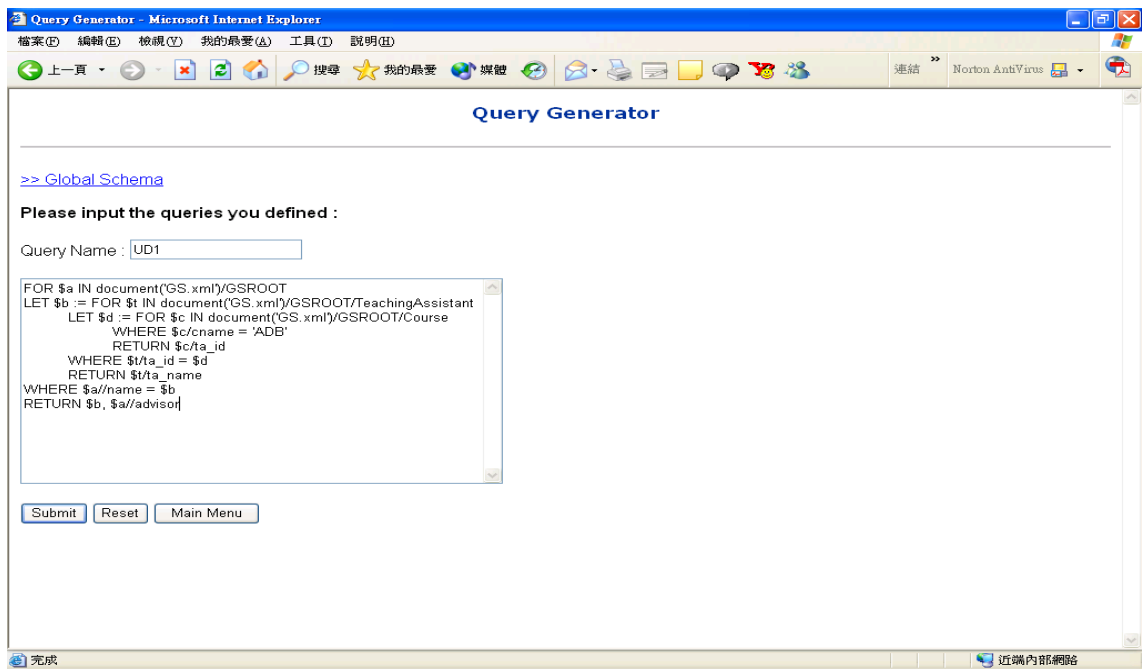


Figure 7: Query Generator - Input

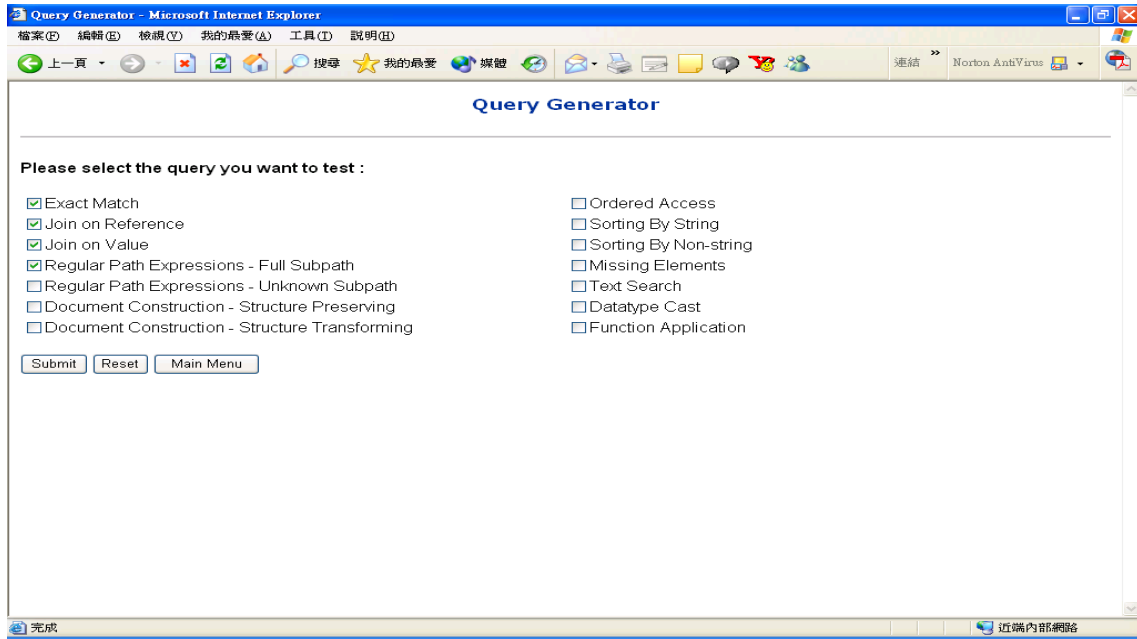


Figure 8: Query Generator – Select Standard Query Type

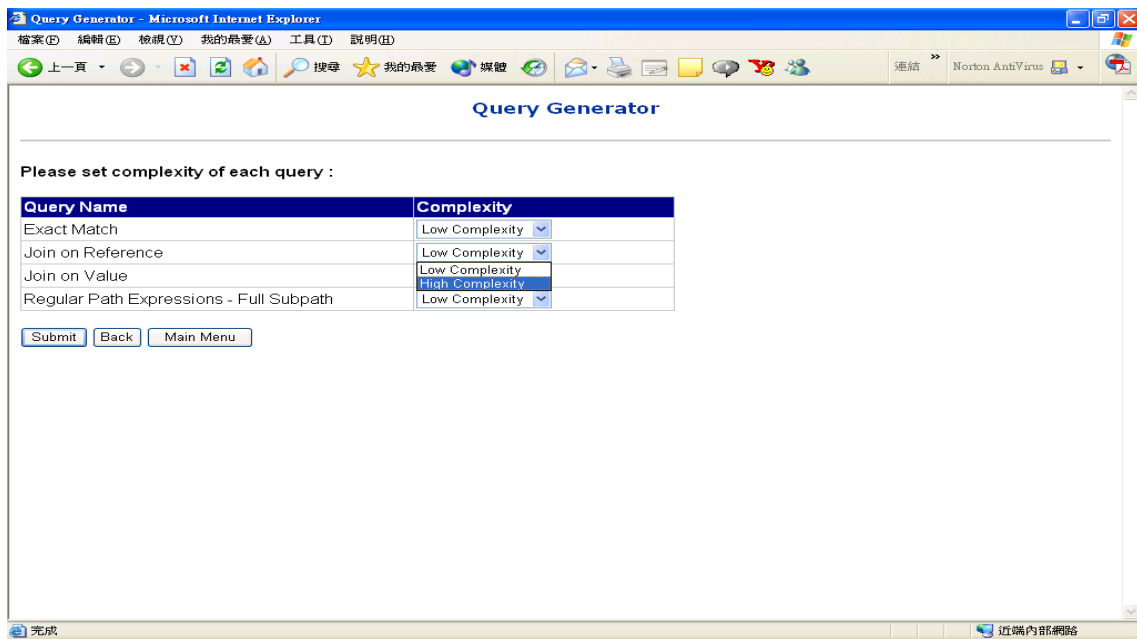


Figure 9: Query Generator – Set Standard Query Complexity

3.6.3 Scheduler

According to the control model, several parameters should be set to

execute the benchmark. The parameters we implement in the prototype are test sequence and number of repetitions. Both in the open query input and in the standard query selector, once the test

query set has been determined, users can set up the executed sequence and the repetitions of each query in the scheduler.

3.6.4 Result Collector

The result collector shows the test results of the queries we specified. The test results can be divided into three parts: the total execution result, the XML query test result, and test result of each data source. If the test query needs to combine ontology reasoning service, the ontology query test result will be

shown in the test result. In the total execution result, the total query response time and throughput are illustrated. In the XML query test result, it shows the query response time and throughput of the XML query processing as shown Figure 10. In the ontology query test result, it adds two extra performance metrics: recall and precision. This can help users evaluate the quality of answer entailed by the ontology. Finally, the result collector lists the query response time and throughput of each data sources.

Sequence	Query Name	Response Time (seconds)	Throughput (queries/per second)
1	Join on Reference		
	Response Time (seconds)	3.495	0.286
	XML Processing Time		
	Response Time (seconds)	0.591	1.692
	This Query Needs Ontology Reasoning - Concept Hierarchy Query		
	Response Time (seconds)	2.834	0.353
	Precision (%)	100	Recall (%)
			100
	SQL Server Processing Time		
	Response Time (seconds)	0.03	33.33
	Tamino XML Server Processing Time		
	Response Time (seconds)	0.04	25
2	Exact Match		
	Response Time (seconds)	0.461	2.169
	XML Processing Time		
	Response Time (seconds)		

Figure 10: Result Collector

4. Research Implications and Concluding Remarks

4.1 Research Implications

In this research, we have accomplished four main tasks. First, an analysis framework of web search and benchmark literature to lay the basis of generic construct development is developed. Our aim is to collect all related literature on the classic web search algorithms and the benchmark methods. We collected and compiled the key web search algorithms and the benchmark methods summarized to be representative. Secondly, a set of heuristics to formulate the generic constructs of web search algorithms are presented. Generic constructs are extracted from the main web search algorithms and the benchmark methods. We analyzed the algorithms and find the essential constructs. For instance, PageRank is based on inlinks and outlinks of the page so these become the key components of the algorithm where the “web page” is a generic construct and the “tag” is the operation of the generic constructs. Thirdly, a more representative and reproducible workload model of web search is created. The generic constructs of a web page is extracted into the page model. The generic constructs of the search types are extracted into the query model. Designed as such, this benchmark meets the desired characteristics of scalability,

portability and simplicity. Fourthly, a computer-assisted benchmarking process is implemented in a prototype system. The prototype system is designed to help prove the feasibility and validity of the research method.

In this research, we have described a detailed approach to model workload requirements from the user's perspective. This results in a more realistic environment of workload representation, transformation, and generation. We have delineated the main components of the method. They include the workload specification scheme, the scheme translator, and the data and operation generators.

The method is domain-representative and workload-representative because we model from the user problem domain and characterize from the user application. The benchmark method is scalable because we can scale up or down the problem size and the problem complexity by changing the data definition and the operation definition via specification. It is reproducible because we use a high-level specification scheme to describe the general workload requirements. The method enables a custom benchmark where users can control the execution through requirements specification instead of manual manipulation.

In the new benchmark method, we have presented a common carrier

concept to capture and compose the user requirements into three carrier components of model. They are the data component model, the operation component model, and the control component model. Web search experiment requires a page model similar to the object model to abstract web as a directed labeled graph in which the nodes model objects and the outgoing edges of an object model the attributes of the object. Designed as such, the benchmark conforms to the desirable characteristics of relevance and rigorousness.

There are several limitations in this research due to the time and resource constraints. We did not verify all performance indicators through the prototype system. The validity of this research can be further improved. The data generator of the prototype system is primitive.

- Due to the infinity of the Internet, we cannot precisely verify all performance variables. Thus, the validity of this research is limited which can be further improved.
- In the prototype, the query generator developed is primitive. So far, it depends on the extent of functions supported by the web search service APIs.
- The experiments are mainly the basic and synthetic tests. Thus, the comprehensiveness and

completeness of experiments can be enhanced.

The future research will continue to augment the experimental prototype in order to accommodate a larger set of data, more complicated operations, more data distribution types, and a wider collection of performance metrics. We plan to develop an expert system to analyze benchmark results, pinpoint performance bottlenecks, provide possible reasons for the test results and advise on the actions to take. In addition, we will further quantify the advantages of the method in the form of metrics on cost and quality. In the future, we further enhance the method to provide users and managers the means to diagnose and detect the strength and weakness of each benchmark.

- Add more advanced generic constructs: Web search benchmark and development for new algorithms is a continuing effort. Continuously collecting the new web search-related literature can help find more advanced generic constructs. Adding more advanced generic constructs to a workload model can advance the generality.
- Enhance the complexity of tests: The experiments we have performed only include the baseline test suites of algorithms. In order to completely verify the workload model, we need to test more new

algorithms. Due to the limitations of the web search APIs, we have only designed ten simple tests to be applied with the APIs. If more web search APIs can be available, we can perform more complicated tests in the future. Another direction is to provide a comparison of experimental results with those of different search engines besides Google and Yahoo.

- Enhance the features of the prototype: The prototype system can be expanded to include the rest of the features of the research method in the future.

4.2 Concluding Remarks

In this research, we have developed the XML and ontology benchmark workload model in intelligent information integration, and built a workload generation prototype. We have reviewed the XML and ontology related literature to motivate the design of the workload model. The objective of this research is to develop a workload model to test whether the intelligent information integration system under EB environment can overcome the diverse formats of content and derive meaning from this content. In order to apply the workload model to different scenarios easier, it is designed in generic constructs. Finally, we validate the research model through the prototype

implementation.

- Enhancing the ontology query model. The development of an ontological standard presents many opportunities and challenges. New reasoning tasks may arise in the future. Retrieval (instances of a concept) and realization (most specific class of instance) may not be sufficient. In order to make the ontology query model more comprehensive, further study to keep track of ontology progression is needed.
- Improving the complexity factors of the XML query model. The complexity factors we analyze in the XML query model are still too rough. Each query type can be analyzed more carefully to refine the query model.
- Implementing various data distributions. In this research, only uniform distribution is implemented. It cannot evaluate performance under different distributions. Implementation of diverse data distributions will become a user requirement.
- Applying the workload model to other applications. Ontology and XML are complementary technologies, and there are other applications that can apply. In this research, we assume the intelligent information

integration system is used on Intranets, such as enterprise information integration (EII), electronic business (EB), and enterprise application integration (EAI). There are other applications between enterprises that may need to integrate heterogeneous information, such as business-to-business integration (B2Bi), collaborative commerce (C-Commerce), and electronic commerce (EC). We can modify the workload model of this research to create other benchmarks that are based on XML and ontology with different characteristics.

5. References

1. Anon et al. (1985). A measure of s processing power. *Datamation*, 17(3), April 1, 112-118.
2. Bapna, R., Goes, P., Gupta, A., & Jin, Y. (2004). User heterogeneity and its impact on electronic auction market design: an empirical exploration. *MIS Quarterly*, 28(1), March, 21-43.
3. Bitton, D., DeWitt, D. J., & Turbyfill, C. (1983). Benchmarking data management Systems - a systematic approach. *Proceedings of the 9th International Conference on Very Large Data Bases*, August, 8-19.
4. Bitton, D. & Turbyfill, C. (1985). Design and analysis of multiuser benchmarks for data management system. *Proceedings of the HICSS-18 Conference*.
5. Bohme, T. & Rahm, E. (2001). Xmach-1: benchmark for XML data management. In *Proceedings of the German Data management Conference*.
6. Cardenas, A. F. (1973). Evaluation and selection of file organization - a model and system. *Communications of the ACM*, 16(9), September, 540-548.
7. Carey, M. J., DeWitt, D. J., & Naughton, J. F. (1993). The 007 benchmark. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, May, 12-21.
8. Carey, J. Michael, David J. DeWitt, Jeffrey F. Naughton, Mohammsd Asgarian, Paul Brown, Johannes E. Gehrke, & Dhaval N. Shah. (1997). The BUCKY object-relational benchmark. *ACM SIGMOD*, 135-146.
9. Cattell, R. G. G. & Skeen, J. (1992). Engineering data management benchmark. *ACM Operations on Data management Systems*, 17(1), March, 1-31.
10. Cetintemel, U., Zimmerman, J., Ulusoy, O., & Buchmann, A. (1999). Objective: a benchmark for object-oriented active data management systems. in *Jthenal of Systems and Software*, 45, 31-43.
11. Clarke, S., & Willett, P. (1997). Estimating the recall performance of search engines. *ASLIB Proceedings*, 49 (7), 184-189.
12. DeWitt, D. J., Fattersack, P., Maier, D., & Velez, F. (1990). A study of three alternative workstation server architectures for object-oriented data management systems. *Proceedings of the 16th International Conference on Very Large Data Bases*, August, 07-121.
13. DeWitt, D. J., Ghandeharizadeh, S., & Schneider, D. (1988). A performance analysis of the gamma data management machine. *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, May,

- 350-360.
14. DeWitt, David, Jeffrey Naughton, Jayavel Shanmugasundaram, Kristin Tufte, Gang He, & Chun Zhang (1999). Relational data managements for querying XML documents: limitations and opportunities in *Proceedings of the 25th VLDB Conference*, 25-32.
 15. Florescu, Daniela & Donald Kossmann (1999). A performance evaluation of alternative mapping schemes for sorting XML data in a relational data management. *Technical Report* No. 3680, INRIA, Le Chesnay Cedex.
 16. Gray, J. (1992). Data management and operation processing benchmarks. *Proceedings of the 1992 ACM SIGMOD international conference on Management of data*.
 17. Gray, J. N. (1993) *The Benchmark Handbook for Data management and Operation Processing Systems*, Ed., Morgan Kaufmann, Inc.
 18. Hevner, A., March, S. T., Park, J. & Ram, S. (2004). Design science research in information systems. *MIS Quarterly* 28(1), March, 75-105.
 19. March, S. T. & Smith, G. (1995). Design and natural science research on information technology. *Decision Support Systems* 15(4), December, 251-266.
 20. Markus, M. L., Majchrzak, A. & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *MIS Quarterly* 26(3), September, 179-212.
 21. Nambiar, U., Lacroix, Z., Bressan, S., Lee, M., Li, Y. (2002). Current approaches to XML management. *IEEE Internet Computing*, July/August, 6(4), 43-51.
 22. Schmidt, A., M. Kersten, D. Florescu, M. Carey, I. Manolescu & F. Waas (2000). The XML store benchmark project. <http://www.xmlperformanc e.org>.
 23. Schmidt, A. R., F. Waas, M.L. Kersten, D. Florescu, I. Manolescu, M.J. Carey & R. Busse (2001). The XML benchmark project. *CWI Technical reports*.
 24. Schmidt, A., F. Waas, M.L. Kersten, D. Florescu, I. Manolescu, M.J. Carey & R. Busse (2001). Why and how to benchmark XML data management. *SIGMOD Record*, 30(3), 41-50.
 25. Seng, J. L. (1998). Comparing object-oriented database systems benchmarks. *Proceedings of the 31st Hawaii International Conference on Systems Sciences*, January.
 26. Seng, J. L., Yao, S. B. & Hevner, A. R. (2005). Requirements-driven database benchmark method. *Decision Support Systems*, 38, 629-648.
 27. Shanmugasundaram, Shekita, Kiernan, Krishnamurthy, Viglas,

- Naughton & Tatarinov. (2001). A general technique for querying XML documents using a relational data management system. *ACM SIGMOD RECORD*.
28. Storey, V. C., Chiang, R. H. L., Dey, D., Goldstein, R. C., & Sundaresan, S. (1997). Database design with common sense business reasoning and learning. *ACM Transactions on Database Systems*, 22(4), December, 471-512.
 29. Tillquist, J., King, J. & Woo, C. (2002). A representational scheme for analyzing information technology and organization dependency. *MIS Quarterly*, 26(2), Jun, 91-118.
 30. Transaction Processing Performance Council. (2006).
 31. Yao, S. B., Hevner, A. R. & Myers, H. Y. (1987). Analysis of data management system architectures using benchmarks. *IEEE Transactions on Software Engineering*, SE-13(6), June, 709-725.
 32. Yu, P. S., Chen, M. S., Heiss, H. U., & Lee, S. (1992). On workload characterization of relational data management environments. *IEEE Transactions on Software Engineering*, SE-18(4), April, 347-355.
 33. Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information*

Systems Research 3(1), March, 36-59.

6. 計畫成果自評

資訊科技和網際網路的蓬勃發展，異質資訊整合在企業電子化與電子商務環境中已是一項普遍存在且重要的議題。在缺乏整合的情形下，個別地存取異質資訊來源，已造成資訊的混雜和錯誤及浪費，尤其不能提供即時管理決策分析給企業主管。本研究發展出一項能夠整合不同資料模型，以及這資料模型中衍生出的語意，依照延伸標記語言與本體論以及學名式的資料結構進行建模，並達到負載量模型具有可攜性和延展性的趨近一般化的績效評估方法。此資料模型結合本體和延伸標籤語言的查詢方法，提高對底層異質資訊來的結構互動性和語意互動。基於學名結構方法，發展單一的標準轉換模式和交換格式，促成分散地的資料倉儲間形成多對多的系統化映對模式，輔以多維度中介資料管理功能，可形成網路上通用並能兼顧效率與品質的雛型系統。對於在電子商務環境中，企業對於大量的資料和資訊，賴有效系統工具來協助進行淬取、分析和預測的重要商業智慧利器。本研究的部份研究成果已發表在 *Information & Management, Information Processing & Management, Information Sciences, Expert Systems with Applications* 與 *國科會 A 級國際學術研討會論文* (附錄) 。

(附錄) 論著述 :

A. 期刊論文

1. Seng*, Jia-Lang and Chen, C. T. (2010), An analytic approach to select data mining method for business decision, *Expert Systems with Applications*, Volume 37, Issue 12, December 2010, Pages 8042-8057. (SCI , IF=2.908)
2. Seng*, Jia-Lang and Lai, S. T. (2010), An intelligent information segmentation approach to extract financial data for business valuation, *Expert Systems with Applications*, Volume 37, Issue 9, September 2010, Pages 6515-6530. (SCI , IF=2.908)
3. Seng*, Jia-Lang and Lin, M. I. (2010), Benchmarking intelligent information integration - a generic construct-based model, *Expert Systems with Applications*, Volume 37, Issue 6, June 2010, Pages 4242-4255. (SCI , IF=2.908)
4. Tu, I. L. and Seng*, Jia-Lang (2009), Research intelligence involving information retrieval - an example of conferences and journals, *Expert Systems with Applications*, Volume 36, Issue 10, December 2009, Pages 12151-12166. (SCI , IF=2.908)
5. Seng*, Jia-Lang and Kong, I. L. (2009), A schema and ontology-aided intelligent information integration, *Expert Systems with Applications*, Volume 36, Issue 7, September 2009, Pages 10538-10550. (SCI , IF=2.908)
6. Seng*, Jia-Lang, Ka, I. and Lin, B. (2009), A generic construct based workload model for web search, *Information Processing & Management*, Volume 45, Issue 5, September 2009, Pages 529-554. NSC-95-2416-H-004-006-. (國科會管二學門 A 級國際期刊 , SSCI/SCI , IF=1.783)
7. Chou, T. and Seng*, Jia-Lang (2009), An agility and agent-enabled e-services method: an international telecommunication example, *Information & Management*, Volume 46, Issue 6, August 2009, Pages 342-350. NSC-96-2416-H-004-055-MY3. (國科會管二學門 A 級國際期刊 , SSCI/SCI , IF=2.282)
8. Chou, T., Seng*, Jia-Lang, and Lin, B. (2008), eTOM and e-services-based trouble-management operations-a large scale telecom case study, *International Journal of Technology Management*, Vol. 43, No. 4, pp.383-403. NSC-93-2416-H-004-002-. (SSCI/SCI , IF=0.419)
9. Huang, S., Chou, T. and Seng*, Jia-Lang (2007), Data warehouse enhancement: a semantic cube model approach, *Information Sciences*, Vol.

- 177, Issue 11, 1 June, pp. 2238-2254. NSC-95-2416-H-004-055-MY3. (SCI , IF=3.291)
10. Seng*, Jia-Lang and Lin, T. (2007), An ontology-assisted analysis in aligning business process with ecommerce standards, *Industrial Management and Data Systems*, Vol. 107, No. 3, pp. 415-437. NSC-94-2416-H-004-018-. (SCI, IF=1.535)
 11. Seng*, Jia-Lang and Tsai, W. (2007), An IT capital performance indicator study: evidence from a Taiwan financial service industry case study, *International Journal of Accounting, Auditing, Performance Evaluation*, Vol. 4, No.4/5, pp. 501-528. NSC88-2412-H-004-021-. (SSCI , IF=0.02)
 12. Seng*, Jia-Lang and Kang, C. (2006), A case study on exploring IT capital indicators in Taiwan's high-tech industry, *International Journal of Learning and Intellectual Capital*. NSC87-3011-P-301-001-.
 13. Seng*, Jia-Lang, Yao, S. B, Hevner, A. R. (2005), Requirements-driven database benchmark method, *Decision Support Systems*, 38: 629-648. NSC-92-2416-H-004-018-. (國科會管二學門 A+級國際期刊 , SSCI/SCI , IF=2.622).
 14. Seng*, Jia-Lang and Yu, J. (2005), A Requirements-driven workload model for XML benchmark, *Industrial Management and Data Systems*. NSC93-2416-H-004-002-. (SCI , IF=1.535)
 15. Seng*, Jia-Lang and Wang, R. (2005) , 資料交換與查詢互動於 XML 與關聯資料庫之間 , *資訊管理學報*。(TSSCI)
 16. Seng*, Jia-Lang and Yu, M. (2005) , 一項具適性協同性建構性之互動式數位學習設計 , *電子商務研究* , 6 月。NSC92-2416-H-004-018-。(TSSCI)
 17. Seng*, Jia-Lang and Wu, G. (2005) , 技術創新策略、競爭環境、和績效之關聯性研究 , *管理與資訊學報*。NSC93-2416-H-004-026-。
 18. Seng*, Jia-Lang and Han, S. (2004), A transaction-centric distributed database benchmark workload model, *Data & Knowledge Engineering*. NSC-90-2416-H-004-030-. (SCI , IF=1.745)
 19. Seng*, Jia-Lang (2004), Toward a more generic workload model for xml benchmark, *Computer Systems Science and Engineering*, 19, 5:95, 95-114. NSC-89-2416-H-004-092-. (SCI , IF=0.222)
 20. Seng*, Jia-Lang and Yin, M. (2004), A genetic algorithm assisted hybrid approach to web information integration, *Journal of Information Technology and Information Management*, 13:1. NSC89-2412-H-004-021-.
 21. Seng*, Jia-Lang (2004), A representation of distributed object-oriented DBMS workload model for benchmark, *Journal of Database Management*, October-November. NSC-88-2416-H-004-040-. (SCI , IF=2.0)

22. Seng*, Jia-Lang (2004), An exploratory set of indicators of information technology and process and innovation capital, *Journal of Intellectual Capital*, NSC90-2416-H-004-030-.
23. Seng*, Jia-Lang and Lin, Y. (2004), A workload-generic and web-centric benchmark model design for multi- and hyper-media databases, *International Journal of Services and Standards*, 1:1, 18-38. NSC88-3011-P-301-001-.
24. Seng*, Jia-Lang and Lin, S. (2004), Mobility- and knowledge-centric e-learning application design method, *International Journal on Innovation and Learning*, 1:3, 293-311. NSC88-2412-H-004-021-.
25. Seng*, Jia-Lang and Hwang, M. (2004), 趨近泛用型之物件和物件關聯資料庫績效評估工作量模式設計和建立, *資訊管理學報*, 7月。NSC91-2416-H-004-011-。(TSSCI)
26. Seng*, Jia-Lang, Lin, Y., Wang, R., and Yu, J. (2003), An analytic study of xml database techniques, *Industrial Management and Data Systems*, 103:2, 111-120. NSC-87-2416-H-004-031-.(SCI, IF=1.535)
27. Seng*, Jia-Lang (2003), A study on industry and synthetic standard benchmarks in relational and object databases, *Industrial Management and Data Systems*, 103:7, 516-532. NSC-89-2416-H-004-044-.(SCI, IF=1.535)
28. Seng*, Jia-Lang (2003), A simple set representation of functional and non-functional requirements, *Journal of Information Technology and Information Management*. NSC89-2412-H-004-021.
29. Seng*, Jia-Lang and Lin, C. (2003), 資料檢索全文資料庫績效評估工作量模式之建立與應用, *資訊管理學報*, 7月。NSC89-2416-H-004-092。(TSSCI)
30. Seng*, Jia-Lang and Hwang, M. (2003), 初探高科技產業政策與產業技術創策略配合度之績效關連性, *資訊管理學報*, No.8。(TSSCI)
31. Seng*, Jia-Lang and Fang, H. (2003), UML 導向之主從式物件資料庫績效評估之工作量模式, *資訊管理學報*, 1月。NSC89-2416-H-004-044-。(TSSCI)
32. Seng*, Jia-Lang and Chou, H. (2003), 趨近泛用型之不動產估價資訊系統之設計, *中山管理評論*, 10:2, 12月。NSC88-2412-H-004-021-。(TSSCI)
33. Seng*, Jia-Lang and Wang, I. (2002), A database-centric and web-automatic hypertext application design method, *Journal of Computer Information Systems*, 43:1, 91-110. NSC-86-2213-E-004-005-.(SCI, IF=0.528)
34. Seng*, Jia-Lang (2002), A web-based data warehouse planning, *Global Perspective of Information Technology Management*, IRMA. NSC88-2412-H-004-021-.
35. Seng*, Jia-Lang and Han, S. (2002), A representation of distributed DBMS benchmark workload model, *Information Systems Evaluation Management*,

IRMA. NSC88-3011-P-301-001-.

36. Seng*, Jia-Lang and Chang, C. (2002), SVG 在地理資訊系統空間資料表達上的應用, *管理與資訊學報*, No.7。NSC89-2412-H-004-021-。
37. Seng*, Jia-Lang and Wei, K. (2001), A syntax-based gateway between sql3 and odmng languages, *Journal of Information Science and Engineering*, 17, 161-195, March. NSC-85-2213-E-004-003-. (SCI, IF=0.242)
38. Seng*, Jia-Lang and Cheng, C. (2001), 建構電子商務資料庫績效評估工作量模式之研究, *管理與資訊學報*, No. 6。NSC88-2416-H-004-040-。
39. Seng*, Jia-Lang and Hsu, C. (2000), 開放型分析導向式物件資料庫績效評估模式之研究, *資訊管理學報*, 7:1, 7 月, 43-70。NSC87-3011-P-301-001-。(TSSCI)
40. Seng*, Jia-Lang (2000), 建構整合模式之會計與審計資訊系統, *資訊管理學報*, 6:2, 1 月, 25-49。NSC86-2213-E-004-005-。(TSSCI)
41. Seng*, Jia-Lang and Hsieh, L. (2000), 建構網際網路資料庫選擇模式之研究, *管理與資訊學報*, No.5, pp.275-316。NSC87-2416-H-004-031-。
42. Seng*, Jia-Lang and Chang, C. (1999), 構成高科技產業標準政策因素之初探, *科技管理學刊*, Vol.4, No.2, pp.67-104。NSC85-2213-E-004-003-。
43. Seng*, Jia-Lang (1998), A hybrid design of cost accounting information systems, *政大學報*, 第 76 期, 6 月, 329-344。NSC87-3011-P-301-001-。(TSSCI Extended)
44. Seng*, Jia-Lang (1997), An integral client-server design of accounting information systems, *政大學報*, 第 75 期, 12 月, 213-228。NSC87-2416-H-004-031-。(TSSCI Extended)
45. Seng*, Jia-Lang (1994), Requirements-driven Database Management Systems Benchmark Method, University of Maryland, College Park, USA, Unpublished Dissertation.

B. 研討會論文

1. Seng*, Jia-Lang (2010), Information Segmentation and Extraction in Business Valuation – An Example of Non-financial Data Sources, The American Accounting Association (AAA) Annual Meeting, San Francisco, USA.
2. Seng*, Jia-Lang (2010), Exploring IT Capital Performance Indicators – An Example of High-tech Industry, The American Accounting Association (AAA) Annual Meeting, San Francisco, USA.
3. Seng*, Jia-Lang (2009), Benchmarking Data Warehousing and Data Mining, The American Accounting Association (AAA) Annual Meeting, New York, New York, USA.

4. Seng*, Jia-Lang (2009), An Exploratory Study in Computer-aided Consolidated Financial Reporting Preparation, The American Accounting Association (AAA) Annual Meeting, New York, New York, USA.
5. Seng*, Jia-Lang (2008), A Performance Indicator Study on Information Technology Capital in Financial Service Sector, The American Accounting Association (AAA) Annual Meeting, Anaheim, California, USA.
6. Seng*, Jia-Lang (2008), An Ontology-based B2Bi Information Integration, The American Accounting Association (AAA) Annual Meeting, Anaheim, California, USA.
7. Ka, Seng*, Jia-Lang, and Huang (2007), Web Search Benchmark Workload, Information Systems Application Conference, Taipei, Taiwan.
8. Lai, Seng*, Jia-Lang, and Lin (2007), Information Search in Financial Data, Innovation and Management Conference, Taipei, Taiwan.
9. Wu, Seng*, Jia-Lang, and Chi (2007), Chinese Information Segmentation in Non-financial Data, 21st Century Contemporary Management Conference, Taipei, Taiwan.
10. Seng*, Jia-Lang (2007), A Data Modeling for Cost Accounting Information Systems, The American Accounting Association (AAA) Annual Meeting, Chicago, USA.
11. Seng*, Jia-Lang (2007), A Generic Transformation between XML and UML, The American Accounting Association (AAA) Annual Meeting, Chicago, USA.
12. Seng*, Jia-Lang (2007), A Generic Construct based Workload Model for Business Intelligence, The 38th Decision Science Institute (DSI) Annual Meeting, Phoenix, Arizona, USA.
13. Lin and Seng*, Jia-Lang (2006), Information Integration in B2Bi EC, The 21st Century Management International Conference, Taipei, Taiwan. ,
14. Chen, Chiu, Seng*, Jia-Lang, and Chi (2006), A Selection Model of Data Mining Applications, Management Innovation and Vision International Conference, Taipei, Taiwan.
15. Kang, Tsai, Seng*, Jia-Lang, and Huang (2006), IT Capital Indicators – Evidence from Taiwan High-tech Industry, 2006 International Conference of Digital Technology and Innovation Management, Taipei, Taiwan.
16. Seng*, Jia-Lang (2006), A relational and integral design of cost accounting information systems, The 8th Annual AIS Educator Conference, Estes Park, CO USA.
17. Seng*, Jia-Lang (2006), A selection model for business data mining applications, The 37th Decision Science Institute (DSI) Annual Meeting, San

Antonio, Texas USA.

18. Seng*, Jia-Lang (2006), An integral data design of accounting information systems, The American Accounting Association (AAA) International Meeting, Emerging Learning Forum, Washington, D.C., USA.
19. Seng*, Jia-Lang and Lin (2005), B2Bi EC Alignment Study, Management Organization Annual Conference, Taipei, Taiwan.
20. Seng*, Jia-Lang (2005), A metadata OLAP exchange, The 11th Americas Conference on Information Systems (AMCIS), Omaha, Nebraska USA.
21. Seng*, Jia-Lang (2005), A more generic construct-based data exchange in data warehousing, The 36th Decision Science Institute (DSI) Annual Meeting, San Francisco, California, USA.
22. Seng*, Jia-Lang and Wu, G. (2005), An analytic study on DW benchmarks, International Association of Computer Information Systems, Taipei, Taiwan.
23. Seng*, Jia-Lang (2005), XML and Ontology-assisted Information Integration, National Science Council Annual Conference in Management, Taipei, Taiwan.
24. Seng*, Jia-Lang (2005), Client-server EC Workload Model, National Science Council Research Project Report, Taipei, Taiwan.
25. Seng*, Jia-Lang (2004), A More Generalized Decision Support Systems for Real Estate Valuation, National Science Council Research Project Report, Taipei, Taiwan.
26. Seng*, Jia-Lang, Kang, Lin, and Chiang (2004), A study in indicators on information technology with process capital and innovation capital, International Conference on Intellectual Capital, Taipei, Taiwan.
27. Seng*, Jia-Lang (2004), Toward a generic transformation between B2Bi standards XML and UML, The 10th Americas Conference on Information Systems (AMCIS), New York, USA.
28. Seng*, Jia-Lang (2004), A Process-centric and Innovation-centric Information Technology View on Intellectual Capital, 25th McMaster World Congress, Canda.
29. Seng*, Jia-Lang and Lin (2004), Heterogeneous Information Integration Benchmarking, Management Organization Annual Conference, Taipei, Taiwan,
30. Seng*, Jia-Lang and Kong (2004), Heterogeneous Information Integration using XML and Ontology, Management Organization Annual Conference, Taipei, Taiwan.
31. Seng*, Jia-Lang and Wang (2004), Data Warehouse Metadata Transformation, National Science Council Annual Conference in

Management, Taipei, Taiwan.

32. Seng*, Jia-Lang (2004), Tow ore generic construct-based data exchange in s XML and UML, The 10th Americas Conference on Information Systems (AMCIS), New York, USA.
33. Seng*, Jia-Lang (2003), A schema transformation between UML and XML, The 34th Decision Science Institute (DSI) Annual Meeting, Washington, D. C. USA.
34. Seng*, Jia-Lang (2003), A collaborative commerce process analysis on global logistics, The 34th Decision Science Institute (DSI) Annual Meeting, Washington, D. C. USA.
35. Seng*, Jia-Lang and Chang (2003), SVG-applied Geographic Information System Design, Management Organization Annual Conference, Taipei, Taiwan.
36. Seng*, Jia-Lang and Wang (2003), XML and SQL Interoperability Study, Management Organization Annual Conference, Taipei, Taiwan.
37. Seng*, Jia-Lang and Tsaih (2003), Information Technology Capital in Financial Service Industry, National Science Council Annual Conference in Management, Taipei, Taiwan.
38. Seng*, Jia-Lang and Wang (2003), A study in interaction between information technology and intellectual capital, International Conference on Intellectual Capital, Taipei, Taiwan.
39. Seng*, Jia-Lang (2003), XML and Ontology-assisted Heterogeneous Information Integration, National Science Council Research Project Report, Taipei, Taiwan.
40. Seng*, Jia-Lang (2002), International Conference on Intellectual Capital, Taipei, Taiwan.
41. Seng*, Jia-Lang, Lin, Wang, (2002), A comparative study of standard WEB-DB server benchmarks, The 31st Decision Science Institute (DSI) Annual Meeting, Orlando, Florida USA.
42. Seng*, Jia-Lang, Lin, and Wang (2002), A comparative study of db benchmark in internet commerce, The 35th Hawaii International Conference on Systems Science (HICSS), Hawaii USA.
43. Seng*, Jia-Lang (2002), A Web-Based Data Warehouse Planning, Global Perspective of Information Technology Management, IRMA.
44. Seng*, Jia-Lang and Han, S. (2002), A Representation of Distributed DBMS Benchmark Workload Model, Information Systems Evaluation Management, IRMA.
45. Seng*, Jia-Lang and Kang (2002), Information Technology Capital in

- High-Tech Industry, National Science Council Annual Conference in Management, Taipei, Taiwan, October.
46. Seng*, Jia-Lang and Chiu (2002), Business Intelligence Benchmark, National Chengchi University Thesis Workshop, Taipei, Taiwan.
 47. Seng*, Jia-Lang and Chen (2002), Business Intelligence Design, National Chengchi University Thesis Workshop, Taipei, Taiwan.
 48. Seng*, Jia-Lang and Chou (2002), GIS-based Real Estate Evaluation Design, Management Organization Annual Conference, Taipei, Taiwan.
 49. Seng*, Jia-Lang and Lin (2002), Toward A More Generalized Hypermedia DBMS Workload Model, Management Organization Annual Conference, Taipei, Taiwan.
 50. Seng*, Jia-Lang (2002), Ontology-assisted Information Integration, National Science Council Research Project Report, Taipei, Taiwan.
 51. Seng*, Jia-Lang and Lin (2001), B2Bi Business Process Model in Ontology, National Chengchi University Thesis Workshop, Taipei, Taiwan.
 52. Seng*, Jia-Lang and Chiang (2001), Business Intelligence in Construction Management, National Chengchi University Thesis Workshop, Taipei, Taiwan.
 53. Seng*, Jia-Lang and Wu (2001), Innovation and Environment in Telecommunication Industry, Management Organization Annual Conference, Taipei, Taiwan.
 54. Seng*, Jia-Lang and Huang (2001), Technology Innovation Study in Information Industry, Management Organization Annual Conference, Taipei, Taiwan.
 55. Seng*, Jia-Lang (2001), A More Generalization of Workload Model in Data Warehouse, National Science Council Research Project Report, Taipei, Taiwan.
 56. Seng*, Jia-Lang, Lin and Wang (2001), An analytic study in electronic commerce benchmarks of TPC-W and Web EC, The 32nd Decision Science Institute (DSI) Annual Meeting, San Francisco, California USA.
 57. Seng*, Jia-Lang, Wu, and Lin (2001), A comparative study of DSS/DW DB benchmarks, The 32nd Decision Science Institute (DSI) Annual Meeting, San Francisco, California USA.
 58. Seng*, Jia-Lang, (2001), A Schema Transformation between UML and Xml, The Decision Science Institute Annual Meeting 2001, USA.
 59. Seng*, Jia-Lang, (2001), A Web-based DW Planning- A Large Scale Project Management Information Systems in Taiwan, Global Perspective of Information Technology Management, IRMA.

60. Seng*, Jia-Lang, (2001), A Representation of Distributed Database Benchmark Workload Model, Information Systems Evaluation Management, IRMA.
61. Seng*, Jia-Lang, Lin and Wang (2001), An analytic study in electronic commerce benchmarks of TPC-W and Web EC, The 32nd Decision Science Institute (DSI) Annual Meeting, San Francisco, California USA.
62. Seng*, Jia-Lang and Wei, K (2001), Queue-driven MOM design, International Conference in Information Management and Practice, Taipei, Taiwan.
63. Seng*, Jia-Lang and Lin (2000), Exploring OPEN GL image benchmark tools, The 31st Decision Science Institute (DSI) Annual Meeting, Orlando, Florida USA.
64. Seng*, Jia-Lang and Tong, (2000), A Comparative Study of Standard Web-DB Server Benchmarks, The Decision Science Institute Annual Meeting.
65. Seng*, Jia-Lang and Lin (2000), Information Integration Benchmark, National Chengchi University Thesis Workshop, Taipei, Taiwan.
66. Seng*, Jia-Lang and Kuang (2000), Information Integration In Ontology, National Chengchi University Thesis Workshop, Taipei, Taiwan.
67. Seng*, Jia-Lang and Wang (2000), A Database-Centric Hypertext Application Design, Management Organization Annual Conference, Taipei, Taiwan.
68. Seng*, Jia-Lang and Lin (2000), Broadcasting Digital Library Search Model, Management Organization Annual Conference, Taipei, Taiwan.
69. Seng*, Jia-Lang (2000), XML and Relational Database Schema Transformation, National Science Council Research Project Report, Taipei, Taiwan.
70. Seng*, Jia-Lang and Wang (1999), Data Warehouse XML Metadata, National Chengchi University Thesis Workshop, Taipei, Taiwan.
71. Seng*, Jia-Lang and Kuo (1999), XML and UML Data Modeling, National Chengchi University Thesis Workshop, Taipei, Taiwan.
72. Seng*, Jia-Lang and Huang (1999), ORDBMS Workload Model, Management Organization Annual Conference, Taipei, Taiwan.
73. Seng*, Jia-Lang and Fang (1999), Client-Server DBMS Benchmarking, Management Organization Annual Conference, Taipei, Taiwan.
74. Seng*, Jia-Lang (1999), Object-relational Database Benchmarking, National Science Council Research Project Report, Taipei, Taiwan.
75. Seng*, Jia-Lang and Tsai (1999), A structured transformation approach to

- legacy information systems, The 32nd Hawaii International Conference on Systems Sciences (HICSS), Hawaii USA.
76. Seng*, Jia-Lang and Hsieh, L (1999), Selection model of Web DBMS, International Conference in Information Management, Taipei, Taiwan.
 77. Seng*, Jia-Lang, (1998), An Availability Study of Distributed Information Systems, The 9th International Conference on Information Management.
 78. Seng*, Jia-Lang, (1998), A Language-Base Gateway Between OODBMS and Web, The 9th International Conference on Information Management.
 79. Seng*, Jia-Lang (1998), Comparing object-oriented database systems benchmarks, The 31st Hawaii International Conference on Systems Sciences (HICSS), Hawaii USA.
 80. Seng*, Jia-Lang and Wu (1998), Collaborative Commerce Data Modeling, National Chengchi University Thesis Workshop, Taipei, Taiwan.
 81. Seng*, Jia-Lang and Lin (1998), Mobile e-Learning, National Chengchi University Thesis Workshop, Taipei, Taiwan.
 82. Seng*, Jia-Lang and Han (1998), Distributed DBMS Benchmark Study, Management Organization Annual Conference, Taipei, Taiwan.
 83. Seng*, Jia-Lang and Chang (1998), High-Tech Industry Benchmark Policy Study, Management Organization Annual Conference, Taipei, Taiwan.
 84. Seng*, Jia-Lang (1998), Object-oriented Database Systems Workload Model, National Science Council Research Project Report, Taipei, Taiwan.
 85. Seng*, Jia-Lang and Yu (1997), Interactive e-Learning Design, National Chengchi University Thesis Workshop, Taipei, Taiwan.
 86. Seng*, Jia-Lang and Yo (1997), XML Benchmarking, National Chengchi University Thesis Workshop, Taipei, Taiwan.
 87. Seng*, Jia-Lang and Hsieh (1997), Selection Study on Web DBMS, Management Organization Annual Conference, Taipei, Taiwan.
 88. Seng*, Jia-Lang and Cheng (1997), EC DBMS Benchmarking, Management Organization Annual Conference, Taipei, Taiwan.
 89. Seng*, Jia-Lang (1997), Electronic Commerce Benchmark Model, National Science Council Research Project Report, Taipei, Taiwan.
 90. Seng*, Jia-Lang (1997), Exploring the interaction between business process reengineering and activity-based costing, The International Conference on Decision Science Institute (IDSI), Sydney Australia.
 91. Seng*, Jia-Lang (1996), Using structured requirements techniques to model DB benchmark workload, The 2nd Annual Americans Conference on Information Systems (AMCIS), Phoenix, Arizona, USA.
 92. Seng*, Jia-Lang (1996), Client-Server multimedia database systems

- development, The 2nd Annual Americans Conference on Information Systems (AMCIS), Phoenix, Arizona, USA.
93. Seng*, Jia-Lang (1996), Comparing object oriented information systems analysis methods, The Asia Pacific Decision Science Institute Conference (APDSI), Hong Kong.
 94. Seng*, Jia-Lang (1996), Business Database Benchmarking Using Requirements Analysis, 1996 Pacific-Basin Business Conference.
 95. Seng*, Jia-Lang (1996), Requirements Modeling in Business Database Benchmarking, The Second Annual Americans Conference on Information Systems, AIS-1996.
 96. Seng*, Jia-Lang (1996), Client-Server multimedia database systems development, The 2nd Annual Americans Conference on Information Systems (AMCIS), Phoenix, Arizona, USA.
 97. Seng*, Jia-Lang and Hsu (1996), OODBMS Benchmark Method, Management Organization Annual Conference, Taipei, Taiwan.
 98. Seng*, Jia-Lang and Wei (1996), A Gateway Model Between SQL3 and ODMG, Management Organization Annual Conference, Taipei, Taiwan.
 99. Seng*, Jia-Lang (1996), Requirements-driven Database Benchmarking, National Science Council Research Project Report, Taipei, Taiwan.
 100. Seng*, Jia-Lang (1995), A comparative study in ERP experiences in US and Taiwan, International Conference on Multinational Management, Taipei, Taiwan.
 101. Seng*, Jia-Lang (1994), A literature study on requirements engineering, TIMS/ORSA Annual Conference, Orlando, Florida, USA.
 102. Seng*, Jia-Lang (1993), Extended Abstract, Pre-ICSE Workshop, Baltimore, Maryland, USA.
 103. Seng*, Jia-Lang (1993), Extended Abstract, Pre-SIGMOD Workshop, Washington D.C., USA.
 104. Seng*, Jia-Lang (1993), A mathematical model of requirements analysis, Pre-SIGMOD Workshop, Washington, D.C., USA.
 105. Seng*, Jia-Lang (1993) Considering the Non-Functional Requirements Analysis and Quality Evaluation in Software Design-A Methodological Approach, The Software Engineering Research Forum.
 106. Seng*, Jia-Lang (1992), Requirement Analysis Process Model in Information Systems, The TIMS/ORSA Annual Conference.
 107. Seng*, Jia-Lang (1992), Hierarchical Requirements Analysis Method, The TIMS/ORSA Annual Conference.

C. 專書及專書論文

- 1. Seng*, Jia-Lang, et al., 2006, Intellectual Capital Management, Hua Tai Press, Taipei, Taiwan.**
- 2. Seng*, Jia-Lang, 2002, Database Management and Systems - Theory and Practice, Best Wise Press, Taipei, Taiwan.**
- 3. Seng*, Jia-Lang, 2002, Structured Requirements Analysis Methodology with Object Extension, The Library of Congress, Washington, D.C., USA.**
- 4. Seng*, Jia-Lang, 2002, Structured Requirements Analysis Methodology, Best Wise, Taipei, Taiwan.**
- 5. Seng*, Jia-Lang, et al., 2001, Systems Analysis and Design, NOU Press, Taipei, Taiwan.**
- 6. Seng*, Jia-Lang, et al., 1998, Case Studies in Service Sector, Best Wise Press, Taipei, Taiwan.**
- 7. Seng*, Jia-Lang, et al., 1998, Accounting Information Systems, NOU Press, Taipei, Taiwan.**
- 8. Seng*, Jia-Lang, 1994, Requirements-Driven Database Systems Benchmark Methodology, University of Maryland at College Park, USA.**