References

Altova GmbH and Altova, Inc, xmlspy® 5, Available from: http://www.xmlspy.com/ [2002].

Arch-int N, Sophatsathit P, and Li Y, "Ontology-based Metadata Dictionary for Integrating Heterogeneous Information Sources on the WWW," Journal of Research and Practice in Information Technology 35,4 (November 2003): 285-302.

Arch-int N. and Sophatsathit P., "A Reference Architecture for Integrating Heterogeneous Information Sources using XML and Agent Model," Proceedings of the Joint Conference on Information Sciences, NC, USA (8-14 March 2002): 235-239.

Arch-int N. and Sophatsathit P., "A Semantic Information Gathering Approach for Heterogeneous Information Source on WWW," Journal of Information Science 29,5 (October 2003): 357-374.

Arch-int N., Li Y., Roe P., and Sophatsathit P. "Query Processing the Heterogeneous Information Sources using Ontology-based Approach," Proceedings of the 18th International Conference on Computers and Their Applications, Honolulu, Hawaii, USA (26-28 March, 2003): 438-441.

Arens Y., Chee C.Y., Hsu C.-N. and Knoblock C., "Retrieving and Integrating Data from Multiple Information Sources," International Journal of Intelligent and Cooperative Information Systems 2,2 (1993): 127-158.

Arens Y., Hsu C.-N. and Knoblock C., "Query processing in the SIMS information mediator," In: A. Tate (ed.), Advanced Planning Technology, California (1996): 61-69.

Batini C. and Lenzirini M., "A Methodology for Data Schema Integration in Entity-Relationship Model," IEEE Transactions on Software Engineering 10,6 (1984): 650-654.

Batini C., Lenzirini M. and Navathe S.B., "A Comparative Analysis of Methodologies for Database Schema Integration," ACM Computing Surveys 18,4 (1986): 323-364.

Benjamins V. R. and Gomez-Perez A., "Knowledge System Technology: Ontologies and Problem solving Methods," (2000): 1-11.

Borgida A., "Description Logics in Data Management," IEEE Transactions on Knowledge and Data Engineering 7,5 (1995): 671-682.

Borgida A., Brachman R.J., Mcguinness D.L. and Resnick L.A., "Classic: A structural data model for objects," Proceedings of ACM SIGMOD International Conference on Management of Data, Portland, Oregon, USA (June 1989): 58-67.

Borst W. N. and Akkermans J. M., "Engineering Ontologies," International Journal of Human-Computer Studies 46,2 (1997): 365-406.

Brickley D. and Guha R., Resource Description Framework (RDF) Schema Specification, Available from: http://www.w3.org/TR/2000/CR-rdf-schema-20000327/, W3C Recommendation [March 2000].

Busse S., Kutsche R.-D., Leser U. and Weber H., "Federated Information Systems: Concepts, Terminology and Architectures," Forschungsberichte des Fachbereichs Informatik *Nr. 99-9*, (1999).

Caglayan A. and Harrison C., Agent Sourcebook – A Complete Guide to Desktop, Internet, and Intranet Agents. John Wiley & Sons, 1997.

Castano S., Antonellis V.D. and Vimercati S.D.C., "Global Viewing of Heterogeneous Data Sources," IEEE Transactions on Knowledge and Data Engineering 13,2 (2001): 277-297.

Chen P.P., "The Entity-Relationship Model – Toward a Unified View of Data," ACM Transactions on Database Systems 1,1 (1976): 9-36.

Cheung W. and Cheng H., "The Model-Assisted Global query system for multiple databases in distributed enterprises," ACM Transactions on Information Systems 14,4 (1996): 421-470.

Cluet S., Delobel C., Simeon J. and Smaga K., "Your Mediators Need Data Conversion!," Proceeding of ACM-SIGMOD International Conference on Management of Data (June 1997): 177-188.

Cornel G. and Horstman C., Core Java. SunSoft Press, 1996.

Cui Z. and O'Brien P., "Domain Ontology Management Environment," Proceedings of the 33rd Hawaii International Conference on System Sciences (2000).

Decker S., Melnik S., Harmelen F.V, Fensel D., Klein M., Broekstra J., Erdmann M. and Horrocks I., "The Semantic Web: The Roles of XML and RDF," IEEE Internet Computing 4 (September/October 2000): 63-74.

Farquhar A., Fikes R. and Rice J., "The Ontolingua Server: A Tool for Collaborative Ontology Construction," International Journal of Human-Computer Studies 46 (1997): 707-728.

Farquhar A., Fikes R., Pratt W. and Rice J., "Collaborative ontology construction for information integration," Technical Report KSL-95-63, Stanford University Knowledge Systems Laboratory (1995).

Fensel D., Harmelen F.V., Horrocks I., McGuinness D.L., and Patel-Schneider P.F., "OIL: An ontology infrastructure for the Semantic Web," IEEE Intelligent Systems 16,2 (2001): 38-44.

Fensel D., Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Berlin: Springer-Verlag, 2001.

Florescu D., Levy A. and Mendelzon A., "Database Techniques for the World-Wide Web: A Survey," ACM SIGMOD Record 27,3 (1998).

Fraser J., Tate A. and Uschold M., "The enterprise toolset – an open enterprise architecture," in The Impact of Ontologies on Reuse, Interoperability and Distributed Processing, pp. 42-50. Unicom Seminars, London, 1995.

Fridman-Noy N. and Hafner C.D., "The State of the Art in Ontology Design," AI Magazine 18,3 (1997): 53-74.

Gale T., and Island M., "SmallTalksAgents," MagTech Magazine 10,4 (1993). Available from: http://www.mactect.com/articles/mactech/Vol.10/10.04/SmalltalkAgents/.

Garcia-molina H., Papakonstantinou Y., Quass D., Rajaraman A., Sagiv Y., Ullman J., Vassalos V. and Wisom J., "The TSIMMIS approach to mediation: data models and languages," Journal of Intelligent Information Systems 8,2 (1997): 117-132.

Genesereth M. and Fikes R., "Knowledge interchange format reference manual," Available from: http://logic.standford.edu/sharing/papers/kif.ps [1994].

Genesereth M.R., "Interoperability: An Agent-Based Framework," AI Expert (1995): 34-40.

Gray R., "Agent Tcl: Alpha Release 1.1 (AFOSR contract F49620-93-1-0266 and ONR contract N00014-95-1-1204)," Available from: http://www.cs.dartmouth.edu/~rgray/transportable.html [1995].

Greem R., Article: Java Access to SQL Databases. Canadian Mind Products, 1997.

Gruber T.R., "A translation approach to portable ontology specifications," Knowledge Acquisition 4,2 (1993): 199-220.

Gruber T.R., "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," International Journal of Human-Computer Studies 43,5/6 (1995): 907-928.

Guarino N., Formal Ontology and Information Systems, in Formal Ontology in Information Systems. In Guarino N (ed.), pp. 3-15. Amsterdam, Netherlands: IOS Press, 1998.

Guarino N., "Formal Ontology, Conceptual Analysis and Knowledge Representation," International Journal of Human-Computer Studies, 43,2/3 (1995): 625-640.

Hamilton G., Cattell R. and Fisher M., JDBC Database Access with Java. A Tutorial and Annotated Reference, Addison-Wesley, 1997.

Heijst G.V., Schreiber A.T. and Wielinga B. J., "Using explicit ontologies in KBS development," International Journal of HumanComputer Studies 46,2/3 (1997): 183-292.

Hendler J., "Agents and the Semantic Web," IEEE Intelligent Systems 16,2 (2001): 30-37.

Ho H. C. T. and Tian X.. "Automatic Classification of Database Columns Using Feature Analysis," Submitted for publication, (2001).

Horrocks I., "DAML+OIL: A description logic for the Semantic Web," IEEE Bulletin of the Technical Committee on Data Engineering 25,1 (2002): 4-9.

Jakobovits R., "Integrating Autonomous Heterogeneous Information Sources," Univ. of Washington Technical Report (UW-CSE-971205) (July 1997).

Jepson B., Database Connectivity: The Lure of Java. Java Report. Whiley Computer Publishing, 1997.

Jeuring J., Meertens L., Pemberton S., Schrage M., Steen G.V.D. and Swierstra D., "Proxima – a generic presentation-oriented XML editor," Available from: http://www.cs.uu.nl/research/projects/proxima/ [2002].

Jones D.M., Bench-capon T.J.M. and Visser P.R.S., "Methodologies for Ontology Development," Proceedings of ITi and KNOWS Conference of the 15th IFIP World Computer Congress, Budapest, Hungary (August 1998): 62-75.

Kifer M., Lausen G. and Wu J., "Logical foundations of object-oriented and frame-based languages," Journal of the ACM 42 (1995).

Knoblock C. A. and Ambite J. L., Agents for Information Gathering. In Bradshaw J. M. (ed.), Software Agents. pp.347-373. AAAI Press/The MIT Press, 1997.

Lange D. B. and Oshima M., Programming and Deploying Java Mobile Agents with Aglets Addison-Wesley, 1998.

Lassila O. and Swick R. R., "Resource Description Framework (RDF) Model and Syntax Specification," W3C (World Wide Web Consortium), Available from: http://ww.w3.org/TR/1999/REC-rdf-syntax-19990222, [1999].

Levy A.Y., Rajaraman A. and Ordille J.J., "Querying Heterogeneous Information Sources using Source Descriptions," Proceedings of the 22nd International Conference on Very Large Databases, Bombay, India (September 1996): 251-262.

Li Y., Zhang C. and Swan J. R., "An information filtering model on the Web and Its Application in JobAgent," Knowledge-Based Systems 13,5 (2000): 285-296.

Mena E., Kashyap V., Sheth A., and Illarramendi A., "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies," Proceedings of the 1 st IFCIS International Conference on Cooperative Information Systems (CoopIS '96), Brussels, Belgium (June 1996).

Miller R.J., Hernandez M.A., Haas L.M., Yan L.-L., Ho C.T.H., Fagin R., and Popa L.. "The Clio Project: Managing Heterogeneity", SIGMOD Record 30,1 (2001): 78-83.

OEL-url "Ontology Exchange Language (XOL)," Available from: http://www.ai.sri.com/~pkarp/xol/.

OML+-url "Ontology Markup Language/Conceptual Knowledge Markup Language (OML/CKML)," Available from http://www.ontologos.org/OML/CKML-Grammar.html.

Özsu M.T. and Valduriez P., Principles of Distributed Database System. 2nd ed. New Jersey: Prentice-Hall, 1999.

Papakonstantinou Y., Garcia-Molina H. and Widom J., "Object exchange across heterogeneous information sources," IEEE International Conference on Data Engineering, Taipei, Taiwan (March 1995): 251-260.

Papastavrou S., Samaras G. and Pitoura E., "Mobile Agents for WWW Distributed Database Access," IEEE Transactions on Knowledge and Data Engineering 12,5 (2000): 802-820.

Paton N.W., Goble C.A. and Bechhofer S., "Knowledge based information integration systems," International Journal of Information and Software Technology 42,5 (2000): 299-312.

Pitoura E. and Samaras G., Data Management for Mobile Computing. Kluwer Academic Publishers, 1997.

Reddy M.P., Prasad B.E., Reddy P.G. and Gupta A., "A Methodology for Integration of Heterogeneous Databases," IEEE Transactions on Knowledge and Data Engineering 6,6 (1994): 920-933.

Sheth A.P. and Larson J.A., "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," ACM Computing Surveys 22,3 (1990): 183-236.

Sheth A.P., Larson J.A., Cornellio A. and Navathe S., "A Tool for Integrating Conceptual Schemas and User Views," Proceedings of the 4th International Conference on Data Engineering, Los Angeles, CA (February 1988): 176-183.

Stuckenschmidt H., Wache H., Vogele T. and Visser U., "Enabling technologies for interoperability," In Ubbo Visser and Hardy Pundt, editors, Workshop on the 14th International Symposium of Computer Science for Environmental Protection, TZI, University of Bremen, Bonn, Germany (2000): 35-46.

Studer R., Benjamins R. and Fensel D., "Knowledge Engineering: Principles and Methods," Data and Knowledge Engineering 25,1/2 (1998): 161-197.

Sun-url "Sunopsis™ XML Driver (JDBC™ Edition)," Available from: http://www.sunopsis.com/corporate/us/products/jdbcforxml/.

Sure Y., Erdmann M., Angele J., Staab S., Studer R. and Wenke D., "OntoEdit: Collaborative ontology development for the semantic web", Proceedings of the International Semantic Web Conference (ISWC) (2002).

Uschold M. and Gruninger M., "ONTOLOGIES: Principles, Methods and Applications," The knowledge Engineering Review 11,2 (1996): 93-155.

Uschold M., "Building ontologies: Towards a unified methodology," Technical Report AIAI-TR-197, University of Edinburgh (1996).

Vet P.E.V. and Mars N.J.I., "Bottom-Up Construction of Ontologies," IEEE Transactions on Knowledge and Data Engineering 10,4 (1998): 513-526.

W3C, "World Wide Web Consortium, Extensible Markup Language (XML) 1.0 (Second Edition)," Available from: http://www.w3.org/TR/2000/REC-xml-20001006, W3C Recommendation [6-Oct-2000].

Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H. and Hübner S., "Ontology-Based Integration of Information- A Survey of Existing Approaches," ed. Stuckenschmidt H, IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA (2001): 108-117.

Weibel S., Godby J., Miller E. and Danierl R., "OCLC/NCSA Metadata Workshop Report, Dublin, Ohio," Available from: http://www.oclc.org:5046/conferences/metadata/dublin_core_report.html, [1995].

White J.E., "An Introduction to Safety and Security in Telescript," Available from: http://www.genmagic.com, [1995].

Wiederhold G., "Mediators in the Architecture of Future Information Systems," IEEE computer 25,3 (1992): 38-49.

Yang H. and Bennett K., "Acquisition of ERA models from Data Intensive Code", Proceeding of the IEEE International Conference on Software Maintenance (1995): 116-123.

Yang H., Cui Z. and O'Brien P., "Extracting Ontologies from Legacy Systems for Understanding and Re-engineering", Proceeding of the 23rd IEEE International Conference on Computer Software and Applications (1999).

APPENDICES

APPENDIX A

The XML-DTD of the Metadata dictionary

```
<xml id="xml">
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE   MetadataDictionary [
  <!ELEMENT      MetadataDictionary    (VConcepts, PhysicalSourceConfs)>
  <!ATTLIST      MetadataDictionary    MetadataName ID       #REQUIRED>
  <!ELEMENT      VConcepts             (VConcept)+>
  <!ELEMENT      VConcept              (VRelationships?, VProperties)>
  <!ATTLIST      VConcept              VCname       ID       #REQUIRED>
  <!ELEMENT      VRelationships        (VRelationship)+>
  <!ELEMENT      VRelationship         (AssocConcept)+>
  <!ATTLIST      VRelationship         VRelname     (IS-A|IS-PART-OF|Associative)
#REQUIRED>
  <!ELEMENT      AssocConcept          (#PCDATA)>
  <!ATTLIST      AssocConcept          VConcept     IDREF #IMPLIED>
  <!ELEMENT      VProperties           (VPoid|VPord|VPref)+>
  <!ELEMENT      VPoid                 (VDataType, VUnitType, PProperties)>
  <!ATTLIST      VPoid                 VPname       ID       #REQUIRED>
  <!ELEMENT      VPord                 (VDataType, VUnitType, PProperties)>
  <!ATTLIST      VPord                 VPname       CDATA        #IMPLIED>
  <!ELEMENT      VPref                 (#PCDATA)>
  <!ATTLIST      VPref                 VPoid        IDREF #IMPLIED>
  <!ELEMENT      VDataType             (#PCDATA)>
  <!ELEMENT      VUnitType             (#PCDATA)>
  <!ELEMENT      PProperties           (PProperty)+>
  <!ELEMENT      PProperty             (PDataType, PUnitType)>
  <!ATTLIST      PProperty             PPname       CDATA        #REQUIRED
                                       PCname       IDREFS       #REQUIRED
                                       PSname       IDREF        #REQUIRED
                                       CVCname      IDREFS       #REQUIRED>
  <!ELEMENT      PDataType             (#PCDATA)>
```

```
    <!ELEMENT     PUnitType            (#PCDATA)>
    <!ELEMENT     PhysicalSourceConfs  (PSource)+>
    <!ELEMENT     PSource              (PConcept)+>
    <!ATTLIST     PSource              PSname      ID      #REQUIRED>
    <!ELEMENT     PConcept             (PDataModel, Permission, Owner)>
    <!ATTLIST     PConcept             PCname      ID      #REQUIRED>
    <!ELEMENT     PDataModel           (#PCDATA)>
    <!ELEMENT     Permission           (#PCDATA)>
    <!ELEMENT     Owner                (#PCDATA)>
]>
```

APPENDIX B

Examples of valid XML Document conforming to the proposed XML-DTD

```
<MetadataDictionary MetadataName="University">
 <VConcepts>
   <VConcept VCname="Staff">
        <VRelationships>
            <VRelationship VRelname="Associative">
                <AssocConcept>Dept</AssocConcept>
            </VRelationship>
        </VRelationships>
        <VProperties>
            <VPoid VPname="st_id">
                <VDataType>String</VDataType>
                <VUnitType>Null</VUnitType>
                <PProperties>
                    <PProperty PPname="Staff_id"
                          PCname = "Staff_Member"
                          PSname = "Source1"
                          CVCname = "Staff">
                        <PDataType>String</PDataType>
                        <PUnitType>Null</PUnitType>
                    </PProperty>
                    <PProperty PPname="Inst_id"
                          PCname = "Instructor_Member Course_Teach"
                          PSname = "Source2"
                          CVCname = "Instructor">
                        <PDataType>String</PDataType>
                        <PUnitType>Null</PUnitType>
                    </PProperty>
                </PProperties>
```

```
</VPoid>
<VPord VPname="st_name">
        <VDataType>String</VDataType>
        <VUnitType>Null</VUnitType>
        <PProperties>
                <PProperty PPname="Staff_name"
                            PCname = "Staff_Member"
                            PSname = "Source1"
                            CVCname = "Staff">
                        <PDataType>String</PDataType>
                        <PUnitType>Null</PUnitType>
                </PProperty>
                <PProperty PPname="Inst_name"
                            PCname = "Instructor_Member"
                            PSname = "Source2"
                            CVCname = "Instructor">
                        <PDataType>String</PDataType>
                        <PUnitType>Null</PUnitType>
                </PProperty>
        </PProperties>
</VPord>
<VPord VPname="st_salary">
        <VDataType>Float</VDataType>
        <VUnitType>US$</VUnitType>
        <PProperties>
                <PProperty PPname="Salary"
                            PCname = "Staff_Member"
                            PSname = "Source1"
                            CVCname = "Staff">
                        <PDataType>Float</PDataType>
                        <PUnitType>US$</PUnitType>
                </PProperty>
                <PProperty PPname="Salary"
                            PCname = "Instructor_Member"
```

```
                                                PSname = "Source2"
                                                CVCname = "Instructor">
                                            <PDataType>Float</PDataType>
                                            <PUnitType>AUS$</PUnitType>
                                    </PProperty>
                            </PProperties>
                    </VPord>
                    <VPref>dept_id</VPref>
            </VProperties>
    </VConcept>
    <VConcept VCname="Instructor">
            <VRelationships>
                    <VRelationship VRelname="IS-A">
                            <AssocConcept>Staff</AssocConcept>
                    </VRelationship>
            </VRelationships>
            <VProperties>
                    <VPord VPname="Position">
                            <VDataType>String</VDataType>
                            <VUnitType>Null</VUnitType>
                            <PProperties>
                                    <PProperty PPname="Ac_Position"
                                                PCname = "Instructor_Member"
                                                PSname = "Source2"
                                                CVCname = "Instructor">
                                            <PDataType>String</PDataType>
                                            <PUnitType>Null</PUnitType>
                                    </PProperty>
                            </PProperties>
                    </VPord>
            </VProperties>
    </VConcept>
    <VConcept VCname="CourseTeach">
            <VRelationships>
    <VRelationship VRelname="Associative">
        <AssocConcept>Instructor</AssocConcept>
    </VRelationship>
    <VRelationship VRelname="Associative">
        <AssocConcept>Crs</AssocConcept>
    </VRelationship>
</VRelationships>
<VProperties>
    <VPref>st_id</VPref>
    <VPref>crs_id</VPref>
    <VPord VPname="crs_time">
        <VDataType>String</VDataType>
        <VUnitType>Null</VUnitType>
        <PProperties>
            <PProperty PPname="Course_time"
                  PCname = "Course_Teach"
                  PSname = "Source2"
                  CVCname = "Course_Teach">
                <PDataType>String</PDataType>
                <PUnitType>Null</PUnitType>
            </PProperty>
        </PProperties>
    </VPord>
    <VPord VPname="num_stu">
        <VDataType>Integer</VDataType>
        <VUnitType>Null</VUnitType>
        <PProperties>
            <PProperty PPname="Num_students"
                  PCname = "Course_Teach"
                  PSname = "Source2"
                  CVCname = "Course_Teach">
                <PDataType>Integer</PDataType>
                <PUnitType>Null</PUnitType>
            </PProperty>
            </PProperties>
        </VPord>
    </VProperties>
</VConcept>
<VConcept VCname="Crs">
    <VProperties>
        <VPoid VPname="crs_id">
            <VDataType>String</VDataType>
            <VUnitType>Null</VUnitType>
            <PProperties>
                <PProperty PPname="Course_id"
                        PCname = "Course Course_Teach"
                        PSname = "Source2"
                        CVCname = "Course Course_Teach">
                    <PDataType>String</PDataType>
                    <PUnitType>Null</PUnitType>
                </PProperty>
            </PProperties>
        </VPoid>
        <VPoid VPname="crs_name">
            <VDataType>String</VDataType>
            <VUnitType>Null</VUnitType>
            <PProperties>
                <PProperty PPname="Course_name"
                        PCname = "Course"
                        PSname = "Source2"
                        CVCname = "Course">
                    <PDataType>String</PDataType>
                    <PUnitType>Null</PUnitType>
                </PProperty>
            </PProperties>
        </VPoid>
    </VProperties>
</VConcept>
```

```
   <VConcept VCname="Dept">

       <VProperties>
           <VPoid VPname="dept_id">
               <VDataType>Integer</VDataType>
               <VUnitType>Null</VUnitType>
               <PProperties>
                   <PProperty PPname="Dept_id"
                         PCname = "Department Staff_Member"
                         PSname = "Source1"
                         CVCname = "Department Staff">
                       <PDataType>Integer</PDataType>
                       <PUnitType>Null</PUnitType>
                   </PProperty>
               </PProperties>
           </VPoid>
           <VPord VPname="dept_name">
               <VDataType>String</VDataType>
               <VUnitType>Null</VUnitType>
               <PProperties>
                   <PProperty PPname="Dept_name"
                         PCname = "Department"
                         PSname = "Source1"
                         CVCname = "Department">
                       <PDataType>String</PDataType>
                       <PUnitType>Null</PUnitType>
                   </PProperty>
               </PProperties>
           </VPord>
       </VProperties>
   </VConcept>
 </VConcepts>

<PhysicalSourceConfs>
```

```
        <PSource PSname = "Source1">
                <PConcept PCname= "Staff_Member">
                        <PDataModel>Relation</PDataModel>
                        <Permission>RW-</Permission>
                        <Owner>AAA</Owner>
                </PConcept>
                <PConcept PCname= "Department">
                        <PDataModel>Relation</PDataModel>
                        <Permission>RW-</Permission>
                        <Owner>AAA</Owner>
                </PConcept>
        </PSource>

        <PSource PSname = "Source2">
                <PConcept PCname= "Instructor_Member">
                        <PDataModel>XML</PDataModel>
                        <Permission>RW-</Permission>
                        <Owner>BBB</Owner>
                </PConcept>
                <PConcept PCname= "Course_Teach">
                        <PDataModel>XML</PDataModel>
                        <Permission>RW-</Permission>
                        <Owner>BBB</Owner>
                </PConcept>
                <PConcept PCname= "Course">
                        <PDataModel>XML</PDataModel>
                        <Permission>RW-</Permission>
                        <Owner>BBB</Owner>
                </PConcept>
        </PSource>
</PhysicalSourceConfs>
</MetadataDictionary>

</xml>
```

APPENDIX C

Algorithms for mapping virtual schema to physical schema of the metadata dictionary

Input: 1) A labeled tree T of XML-based metadata dictionary, such that $T = <t, n>$.

2) A pair of $<vc, vp>$, where $vc \in C$ is a virtual concept name specified in the FROM clause of a global transaction Q and $vp \in S$ is a virtual property name of vc specified in the SELECT or the WHERE clause.

Output: A finite set $Result = \{r_i \mid i = 1 \ldots n\}$ of virtual and physical mapping information, such that $r_k \in Result$ is defined as a 6-tuple < *vc, vp*, *PPname*, *PCname*, *PSname, CVCname*>, where *PPname*, *PCname*, *PSname, CVCname* are physical property name, concept name, source name, and corresponding virtual concept name, respectively.

Remarks: 1) If the input $vc_c \in C$ is a subconcept of another concept vc_k and $vp_{ci} \in S$ is a virtual property inherited from vc_k, we will use a pair of $<vc_k, vp_{ci}>$ as an input to the algorithm.

2) For a queue Q, we assume that the function *RemoveFromQ(Q)* returns the value of the designated item on Q and removes that item from Q.

```
Procedure InQueue (v: VertexType; var Q: Queue);
begin
          initialize Q to empty;
          for each unmarked vertex w adjacent to v do
          begin
                    visit and mark w;
                    insert w in Q;
          end for
          return (Q);
end Procedure
```

```
Function SEARCH(v : VertexType; A, B, C : String; var x : VertexType);
begin
```

```
        InQueue(v, q);
        x = RemoveFromQ(q) ;
        visit x;
        if (l(x) == A) and (d(x) == B)) then
        begin
            while q is nonempty do
            begin
                x = RemoveFromQ(q);
                visit x;
                if l(x) == C then
                  return (true);
            end while
        end if
        return (false);
end function

begin  /* main */
      Let Result = $\phi$;
      Let v = root(T);
      for each unmarked vertex w adjacent to v do   /* Search the word "VConcepts"  */
      begin
          visit and mark w;
          if (l(w) == "VConcepts") then
          begin
              v = w;

              /* Search for each node labeled "VConcept" */
              for each unmarked vertex w adjacent to v do
              begin
                  visit and mark w;

        /* Search if node labeled "VCname" = vc then mark the node labeled "VProperties" to x  */
                  if SEARCH(w, "VCname", vc, "VProperties", x) then
            /* Search for each node labeled "VPoid", "VPord" and "VPref" */
            for each unmarked vertex w adjacent to x do
            begin
              visit and mark w;

    /* Search if node labeled "VPname" = vp then mark the node labeled "PPproperties" to y */
              if SEARCH(w, "VPname", vp, "PProperties", y) then
                /* for each node labeled "PProperty" */
                for each unmarked vertex w adjacent to y do
                 begin
                   visit and mark w;
                   v =w;
                 /* retrieve the value of nodes labeled "PPname", "PSname", and
                "PCname" to variables PPname, PSname, and PCname  */
                   while there is an unmarked vertex w adjacent to v do
                   begin
                     visit and mark w;
                      if l(w) == "PPname" then
                        PPname = d(w);
                      else if l(w) == "PSname" then
                          PSname = d(w);
                        if l(w) == "PCname" then
                          PCname = d(w);
                          if l(w) == "CVCname" then
                            CVCname = d(w);
                   end while
                 /* Add the results to set Result */
                   Add (vc, vp, PPname, PSname, PCname,CVCname) to Result;
                 end for
              end for
        end for
    end if /* if l(w) == "VConcepts" */
  end for
end  /* main */
```

APPENDIX D

Algorithms for transforming the XML results returned from a single source into a unified XML-based data

Input: 1) A labeled tree T of XML-based metadata dictionary, such that $T = <t, n>$.

2) A labeled tree R representing an XML result returning from a search agent to the managing agent, such that $R = <r, n>$.

Output: A finite set $UXML = \{X_i \mid \forall i=1...n\}$of integrating XML results, such that each $X_k = \{<VP_j, VPD_j> \mid \forall j = 1...m\}$ is a finite set of a virtual property pair VP_c and data value of VP_c obtained from converting the value of a physical property in the physical source. The set $UXML$ is used to generate a unified XML-based data corresponding to a global transaction of a user's request.

Remarks: The root node of labeled tree R denotes the physical source name.

```
Procedure MarkTreeNew(v: VertexType)
begin
        visit and mark $v$;
        for each unmark vertex $w$ adjacent to $v$ do
        begin
                visit $w$ and mark "new";
                $v = w$;
        end for
end procedure
```

```
Procedure Search_VP(PPN: String, Var VP: String)
begin
      $r$ = root($T$);
      MarkTree($r$);
      $v = r$;
      visit and mark $v$;
      for each vertex $w$ marked "new" adjacent to $v$ do
      begin
```

```
            visit and mark w "old";
            if l(w) == "VPname" then
                VP = d(w);
            else if (l(w) == "PPname") and (d(w) == PPN) then
                    return (VP);
            v = w;
        end for
    end procedure

    begin  /* main */
        U = ϕ;
        Initialize array X to Null;
        I = 1;
        r = root(R);
        MarkTreeNew(r);
        visit and mark r;
        for each w marked "new" adjacent to r do
        begin
            visit and mark w "old";
            v = w; J = 1;
            for each w marked "new" adjacent to v do
            begin
                visit and mark w "old";
                PPN[J] = l(w);
                PPD[J] = d(w);

      /* Search the associated virtual property name VP of physical property name PPN[J] from the
         metadata dictionary*/
                Search_VP(PPN[J], VP);
      /* Check if PPN[J] is a non-key property or if PPN[J] is a key property and VP of
      PPN[J] is selected from a global transaction (or VP ∈ S) or not */
                if not k(w) or (k(w) and (VP ∈ S)) then
                begin
```

```
// Convert the physical data value conforming to the data type, unit type defined in the virtual
         property VP
                            ConvertPPDtoVPD(VP, PPD[J], VPD);

     /* Add the returned virtual property name VP and value of virtual property VPD to array X[I] */
                            Add (VP, VPD) to X[I];
                      end if
                      J++;
                end for

                Add X[I] to UXML;
                I++;
                Initialize array PPN, PPD to Null;
         end for
end main
```

APPENDIX E

Algorithms for multiple source integration using merging technique

Input: 1) A labeled tree T of XML-based metadata dictionary, such that $T = <t, n>$.

2) The labeled trees R_i of each XML result returning from each search agent to the managing agent, such that $R_i = <r_i, n_i>$, $i = 1 \ldots m$, and m = no. of search agents.

3) The set S of a global transaction.

Output: A finite set $UXML = \{X_i \mid \forall i{=}1 \ldots n\}$ of integrating XML results, such that each $X_k = \{<VP_j, VPD_j> \mid \forall j = 1 \ldots m\}$ is a finite set of a virtual property pair VP_c and data value of VP_c.

Remarks: 1) The root node of each tree R_i labels the source name for each physical information source.

2) The type of variables P, A, B, Xa, and Xb are the set of a pair of values $<VP_j, VPD_j>$, where $\forall j = 1 \ldots m$.

```
Procedure GetKey_VP_VPD(v: VertexType; var K, P: Set of a pair of values)
begin
        $I = 1$;
        $A = \phi$;
        for each w marked "old" adjacent to v do
        begin
                visit and mark w "old";
                $PPN[I] = l(w)$;
                $PPD[I] = d(w)$;
                Search_VP($PPN[I]$, $VP$);
                ConvertPPDtoVPD($VP$, $PPD[I]$, $VPD$);

  /* k(w) return "true" or "false", if k(w) = true means node w is a key node */
                if k(w) then
                  Add (VP, VPD) to K;
                else
                  Add (VP, VPD) to P;
```

```
            $I$++;
        end for
end procedure
Procedure MergeTreesToUXML(v: VertexType; Ka, Kb, Xa, Xb: Set of a pair of values;
var UXML: Set of a pair of values, merge: boolean)
begin
        merge = false;
        if ($Ka == Kb$) and ($Xa == Xb$) then
        begin
/* check if each $VP_i$ of set $Ka$ is selected from a global transaction or $VP_i \in S$ or not */
              if ChkKeySelect($Ka$) then
                 Add ($Ka \cup Xa$) to $UXML$;
              else Add $Xa$ to $UXML$;
              merge = true;
        end if
        else if ($Ka == Kb$) and (($\forall x$ in $Xa$) $\in Xb$) then
            begin
                if ChkKeySelect($Kb$) then
                    Add ($Kb \cup Xb$) to $UXML$;
                else Add $Xb$ to $UXML$;
                merge = true;
            end if
              else if ($Ka == Kb$) and ($\forall x$ in $Xb$) $\in Xa$) then
                  begin
                      if ChkKeySelect($Ka$) then
                          Add ($Ka \cup Xa$) to $UXML$;
                      else Add $Xa$ to $UXML$;
                      merge = true;
                  end if

        if merge then
```

```
            MarkTreeOld($v$);
        return ($UXML$);
end procedure

begin  /* main */
        $UXML = \phi$; $i = 1$;
        Initialize array $Xa$ and $Xb$ to Null;
        while there is a tree $R_i$ of XML results do
        begin
                $r$ = root($R_i$);
                MarkTreeNew($r$);
        end while
        for each $R_i$ of XML results do
        begin
              $r$ = root($R_i$);
              visit and mark $r$;
              for each $w$ marked "new" adjacent to $r$ do
              begin
                    visit and mark $w$ "old";
                    $j = i$;

    /* Add a pair of <$VP$, $PPD$> for PPN that is a key node to set $Ka$  */
    /* and Add a pair of <$VP$, $PPD$> for each PPN that is not a key node to set $Xa$ */
                    GetKey_VP_VPD($w$, $Ka$, $Xa$);

                    for each  $R_{j+1}$ of XML results do  /* Search $Xa$ in other labeled trees */
                    begin
                          $r$ = root($R_{j+1}$);
                          visit and mark $r$;
                          for each $w$ marked "new" adjacent to $r$ do
                          begin
                                visit and mark $w$;
```

```
   /* Add a pair of <VP, PPD> for PPN that is a key node to set Kb  */
   /* and Add a pair of <VP, PPD> for each PPN that is not a key node to set Xb */
                      GetKey_VP_VPD(w, Kb, Xb);

     /* Compare members in set Xa and Xb and merge into set UXML */
                      MergeTreesToUXML(w, Ka, Kb, Xa, Xb, UXML, merge);
                      if merge then
                        exit for
                  end for
                  if merge then
                    exit for
                  j++;
              end for

              /*The set of key Ka does not match with any key in other labeled trees*/
              if not merge then
              begin
                  if ChkKeySelect(Ka) then
                    Add (Ka ∪ Xa) to UXML
                  else Add Xa to UXML;
              end if
          end for
          i++;
     end for
end main
```

APPENDIX F

Algorithms for multiple source integration using join technique

Input: 1) A labeled tree T of XML-based metadata dictionary, such that $T = <t, n>$.

2) The labeled trees R_i of each XML result returning from each search agent to the managing agent, such that $R_i = <r_i, n_i>$, $i = 1...m$, and m = no. of search agents.

3) The set S of a global transaction.

Output: A finite set $UXML = \{X_i \mid \forall i=1...n\}$ of integrating XML results, such that each $X_k = \{<VP_j, VPD_j> \mid \forall j = 1...m\}$ is a finite set of a virtual property pair VP_c and data value of VP_c.

Remarks: 1) The root node of each tree R_i labels the source name for each physical information source.

2) The type of variables P, A, B, Xa, and Xb are the set of a pair of values $<VP_j, VPD_j>$, where $\forall j = 1...m$.

```
Procedure GetKey_VP_VPD(v: VertexType; Var K, P: Set of a pair of values)
begin
    $I = 1$;
    $A = \phi$;
    for each $w$ marked "old" adjacent to $v$ do
    begin
        visit and mark $w$ "old";
        $PPN[I] = l(w)$;
        $PPD[I] = d(w)$;
        Search_VP($PPN[I]$, $VP$);
        ConvertPPDtoVPD($VP$, $PPD[I]$, $VPD$);

        /* $k(w)$ return "true" or "false", if $k(w)$ = true means node $w$ is a key node */
        if $k(w)$ then
          Add ($VP$, $VPD$) to $K$;
```

```
            else
              Add ($VP$, $VPD$) to $P$;
            $I$++;
        end for
end procedure

Procedure JoinTreesToUXML(v: VertexType; Ka, Kb, Xa, Xb: Set of a pair of values; Var
UXML: Set of a pair of values, join: boolean)
begin
    $join$ = false;
    if ($Ka$ == $Kb$) then
    begin
/* check if each $VP_i$ of set $Ka$ is selected from a global transaction or $VP_i \in S$ or not */
            if ChkKeySelect($Ka$) then
               Add ($Ka \cup Xa \cup Xb$) to $UXML$;
            else Add ($Xa \cup Xb$) to $UXML$;
            $join$ = true;
    end if
    if $join$ then
        MarkTreeOld($v$);
    return ($UXML$);
end procedure

begin  /* main */

    $UXML = \phi$; $i$ = 1;
    Initialize array $Xa$ and $Xb$ to Null;
    while there is a tree $R_i$ of XML results do
    begin
            $r$ = root($R_i$);
            MarkTreeNew($r$);
    end while
    for each $R_i$ of XML results do
    begin
```

```
                    r = root(R_i);
                    visit and mark r;
                    for each w marked "new" adjacent to r do
                    begin
                         visit and mark w "old";
                         j = i;

     /* Add a pair of <VP, PPD> for PPN that is a key node to set Ka  */
     /* and Add a pair of <VP, PPD> for each PPN that is not a key node to set Xa */
                         GetKey_VP_VPD(w, Ka, Xa);
                         for each  R_{j+1} of XML results do  /* Search Xa in other labeled trees */
                         begin
                              r = root(R_{j+1});
                              visit and mark r;
                              for each w marked "new" adjacent to r do
                              begin
                                   visit and mark w;
     /* Add a pair of <VP, PPD> for PPN that is a key node to set Kb  */
     /* and Add a pair of <VP, PPD> for each PPN that is not a key node to set Xb */
                                   GetKey_VP_VPD(w, Kb, Xb);

          /* Compare members in set Xa and Xb and join to the set UXML */
                                   JoinTreesToUXML(w, Ka, Kb, Xa, Xb, UXML, join);
                                   if join then
                                     exit for
                              end for
                              if join then
                                exit for
                              j++;
                         end for
                    end for
                    i++;
               end for
          end main
```

APPENDIX G

Algorithms for verifying result correctness

Input: 1) A finite set unified XML-based Data $UXML = \{X_i \mid \forall i=1\ldots n\}$, such that each $X_k = \{<VP_j, VPD_j> \mid \forall j = 1\ldots m\}$ is a finite set of a virtual property pair VP_c and data value of VP_c.

2) A finite set of selected virtual properties $SEL = \{s_j \mid \forall j=1\ldots m\}$, such that each s_c (or attributes) is a virtual property in the SELECT clause of a global transaction Q.

Output: The return message to the user whether the result is validate or not.

```
Procedure getMaxLengthSet(U1:Set of a pair of value,VAR U2:Set of a pair of value}
begin

        /* find the max length of set UXML, such that to retrieve only the set required to generate
the set UDTD */

        max = 0; lg = 0;
        Initialize set U2 to null;
        for each X in U1 do
        begin
                lg = length(X);
                if lg > max then
                begin
                        Initialize U2;
                        max = lg;
                        Add X to U2;
                end if
        end for
end procedure

Procedure generateSetUDTD(U1: Set of a pair of value; VAR U2:Set of a pair of value)
begin
        Initialize set U2 to Null;
```

```
          for each element $E$ in $UMax$ do
          begin

/* Get each virtual property that is the first element of a pair of ($VP$, $VPD$) in $UMax$ */
                    $VP$ = get_First_Element($E$);
                    Add $VP$ to $U2$;
          end for
end procedure

begin  /* main */

          /* generate the set $UDTD$ from the set $UXML$ */
          getMaxLengthSet($UXML$, $UMax$);
          generateSetUDTD($UMax$, $UDTD$);
          if $UDTD$== $SEL$ then
                    return ("The return result is validate");
          else
                    return ("The return result is not validate");
end main
```

APPENDIX H

Case Study: Implementation and Installation Guide of SIGA

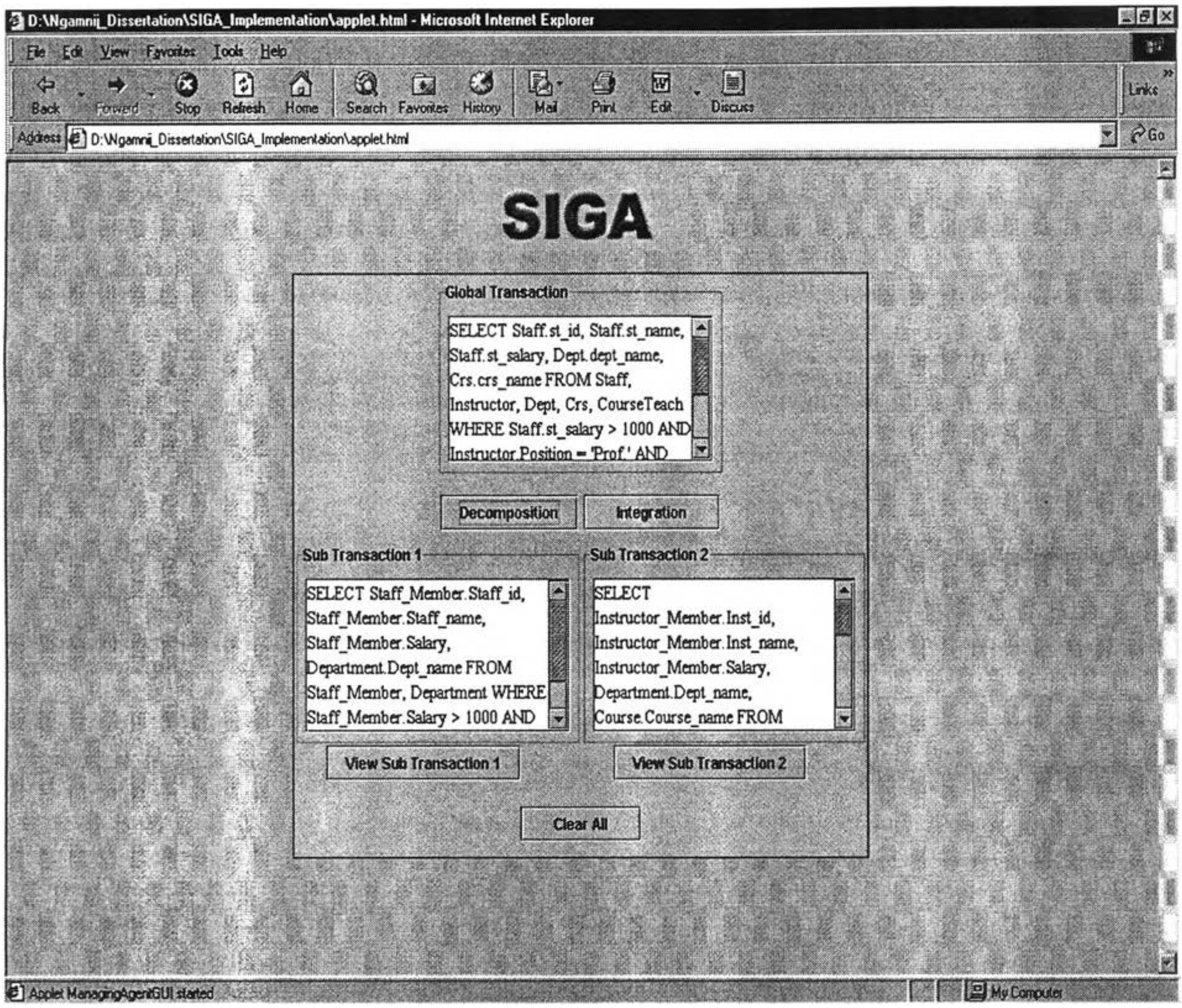
The fact that this research focuses on information access and retrieval on heterogeneous information sources using ontology-based metadata dictionary entails a methodically transformation process of user-level semantic to data processing semantic. A case study is thus contrived to demonstrate the viability of architectural independence, semantic interoperability, and flexible information interchange metadata dictionary through a set of ontology-based rendition and layered-architecture access/retrieval algorithms.

Parts of SIGA are implemented to demonstrate the transformation of theoretical foundation to actual implementation and to validate the proposed algorithms in a practical implementation. The implementation focuses on the decomposition process of a global transaction into sub-transactions and the integration process of the XML results returned from each sub-transaction into a unified XML-based data.

The applications of decomposition and integration processes are designed based on a case study given in Chapter 7 and conformed to the algorithms given in Appendix C to F. We choose Java language to develop these applications since Java is an object-oriented programming language that makes application easier to build, maintain, modify, and reuse. Besides, Java is designed to operate in heterogeneous environments which is ideal for developing the mobile agent programming. The SIGA application is developed as a Java applet in which a Java-enabled web browser can retrieve and execute applet on the local computer, making a displayed web page more dynamic.

H.1. Global Transaction Decomposition

The global transaction decomposition is illustrated via Java applet as shown in Figure H.1. The user can fill in the form to create a global transaction directly and activate the decomposition process to generate the desired sub-transactions.


D:\Ngamnij_Dissertation\SIGA_Implementation\applet.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Links
Address D:\Ngamnij_Dissertation\SIGA_Implementation\applet.html
Go
SIGA
Global Transaction
Decomposition
Integration
Sub Transaction 1
Sub Transaction 2
View Sub Transaction 1
View Sub Transaction 2
Clear All
Applet ManagingAgentGUI started
My Computer

```
SELECT Staff.st_id, Staff.st_name,
Staff.st_salary, Dept.dept_name,
Crs.crs_name FROM Staff,
Instructor, Dept, Crs, CourseTeach
WHERE Staff.st_salary > 1000 AND
Instructor.Position = 'Prof.' AND
```

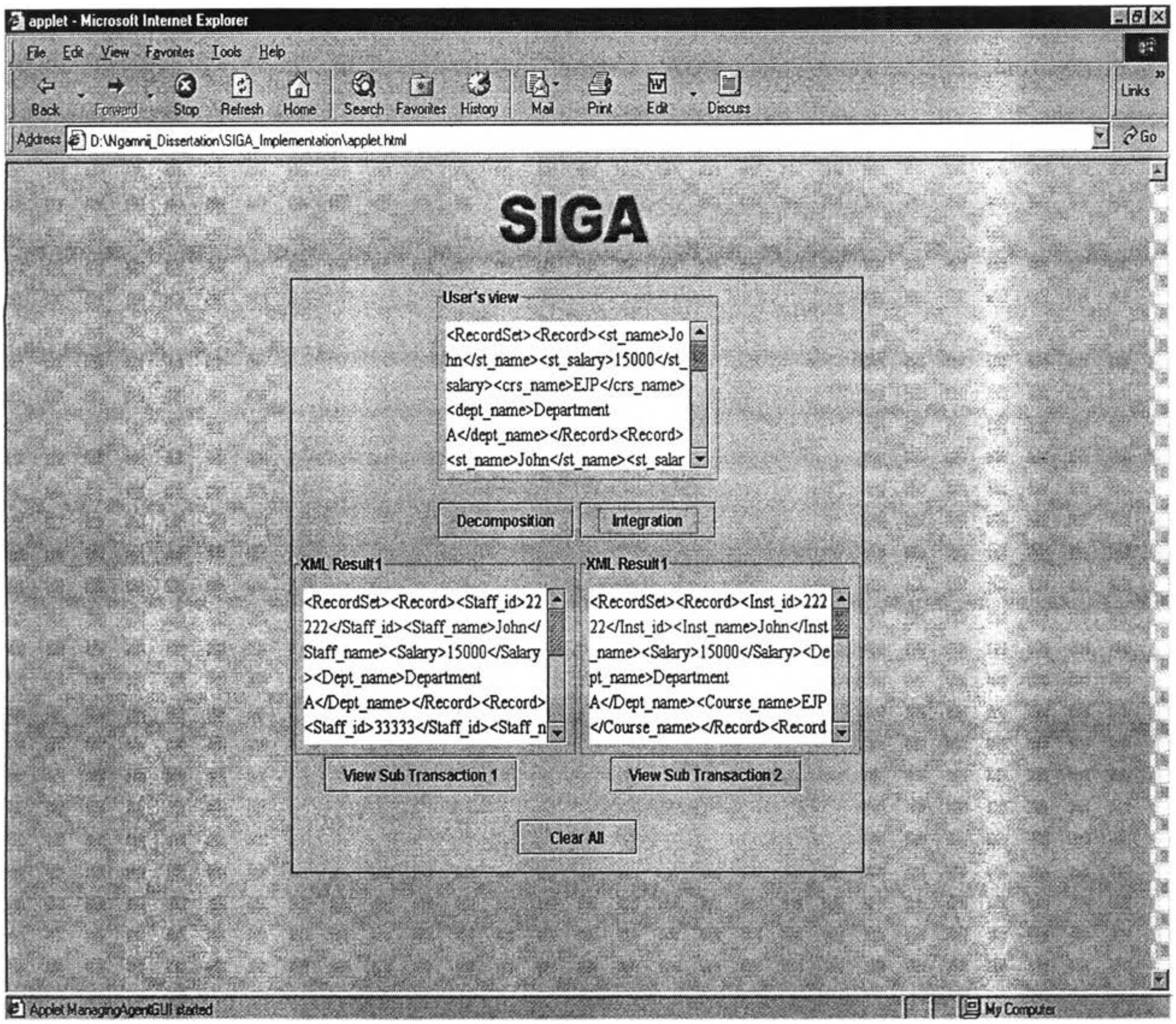
```
SELECT Staff_Member.Staff_id,
Staff_Member.Staff_name,
Staff_Member.Salary,
Department.Dept_name FROM
Staff_Member, Department WHERE
Staff_Member.Salary > 1000 AND
```

```
SELECT
Instructor_Member.Inst_id,
Instructor_Member.Inst_name,
Instructor_Member.Salary,
Department.Dept_name,
Course.Course_name FROM
```

Figure H.1. The global transaction decomposition.

H.2. XML Results Integration

Upon executing each sub-transaction at the destination physical source, the returned result from each source is converted into XML result. The user can view the returned XML result from view sub-transaction screen as illustrated in Figure H.2. The unified XML-based data corresponding to the user's request is generated through the integration process when the user presses the integration button.



applet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Links
Address D:\Ngamnij_Dissertation\SIGA_Implementation\applet.html
Go
SIGA
User's view
Decomposition
Integration
XML Result1
XML Result1
View Sub Transaction 1
View Sub Transaction 2
Clear All
Applet ManagingAgentGUI started
My Computer

User's view

```
<RecordSet><Record><st_name>Jo
hn</st_name><st_salary>15000</st_
salary><crs_name>EJP</crs_name>
<dept_name>Department
A</dept_name></Record><Record>
<st_name>John</st_name><st_salar
```

XML Result1

```
<RecordSet><Record><Staff_id>22
222</Staff_id><Staff_name>John</
Staff_name><Salary>15000</Salary
><Dept_name>Department
A</Dept_name></Record><Record>
<Staff_id>33333</Staff_id><Staff_n
```

XML Result1

```
<RecordSet><Record><Inst_id>222
22</Inst_id><Inst_name>John</Inst
_name><Salary>15000</Salary><De
pt_name>Department
A</Dept_name><Course_name>EJP
</Course_name></Record><Record
```

Figure H.2. The XML results Integration into the unified XML-based data.

H.3. Java Development Kit Environment and Installation

SIGA Application will operate under any Java Development Kit (JDK) environment that provides a JDK 1.1 or JDK 1.2 (also known as Java2 Standard Edition, or J2SDK) compatible platform. The SIGA application has been certified to work with Java 2 SDK, Standard Edition version 1.4.0. We need the full JDK installed, not just the Java Runtime Environment (JRE).

To enable the application run on the Web in applet form, we also need one of the following web browsers:

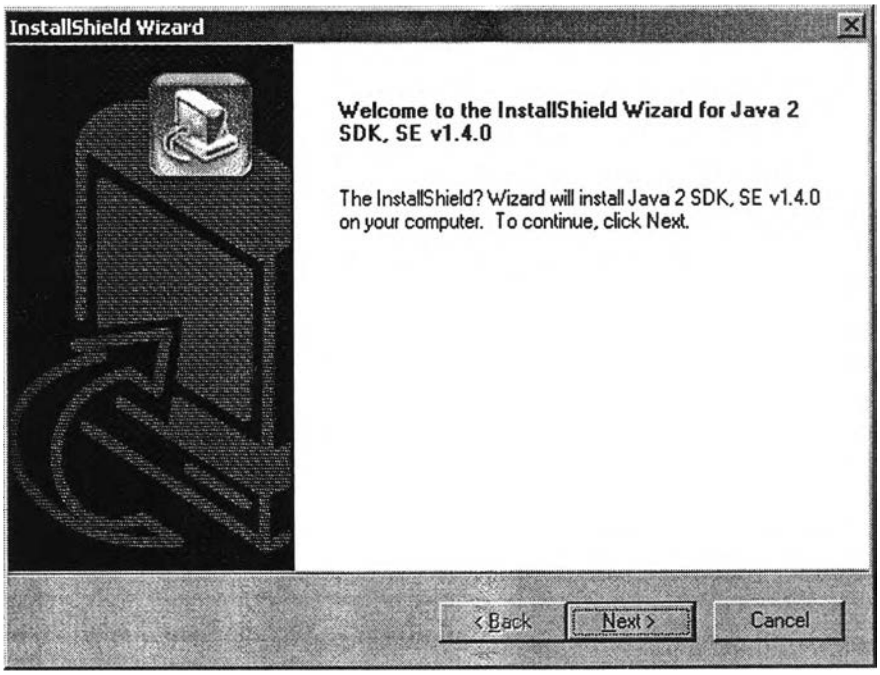
- Netscape Navigator, version 4.6 or later
- Microsoft Internet Explorer, version 5.0 or later

System requirements

- Windows NT 4.0 operating system, Service Pack 3 or 5, Windows 2000, or Windows 98
- 128 MB memory minimum; 160 MB or more is recommended
- 233MHz Pentium II processor
- 64 MB disk space requirements

J2SDK, SE V1.4.0 Installation

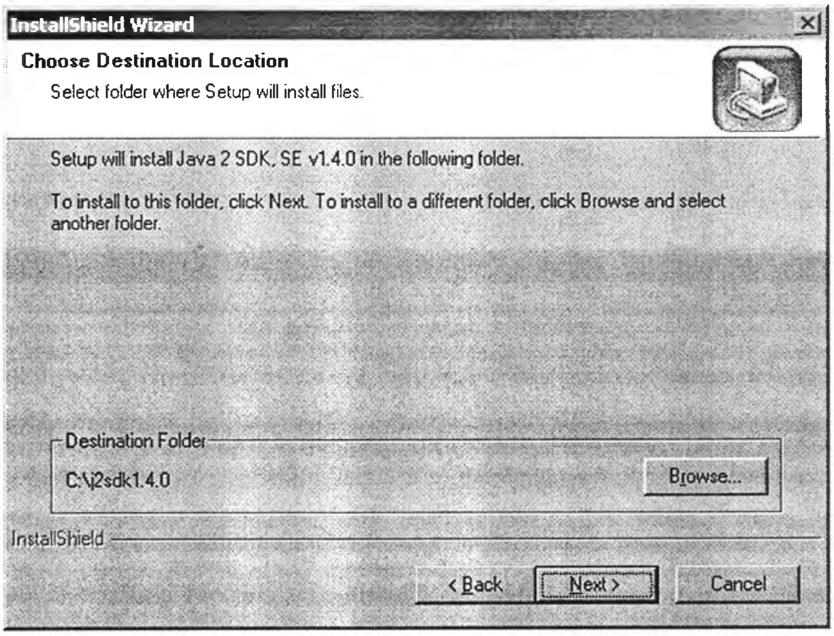
The installation of J2SDK, SE V1.4.0 is accomplished by invoking file "j2sdk-1_4_0-win.exe"



InstallShield Wizard

Welcome to the InstallShield Wizard for Java 2 SDK, SE v1.4.0

The InstallShield? Wizard will install Java 2 SDK, SE v1.4.0 on your computer. To continue, click Next.



Choose Destination Location

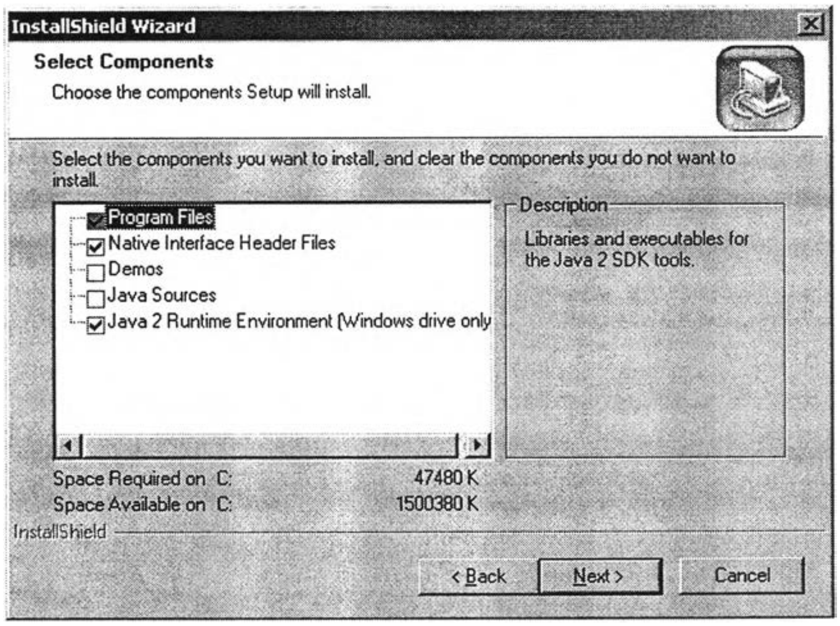
Select folder where Setup will install files.

Setup will install Java 2 SDK, SE v1.4.0 in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder

C:\j2sdk1.4.0



Select Components

Choose the components Setup will install.

Select the components you want to install, and clear the components you do not want to install.

- ☑ Program Files
- ☑ Native Interface Header Files
- □ Demos
- □ Java Sources
- ☑ Java 2 Runtime Environment (Windows drive only

Description

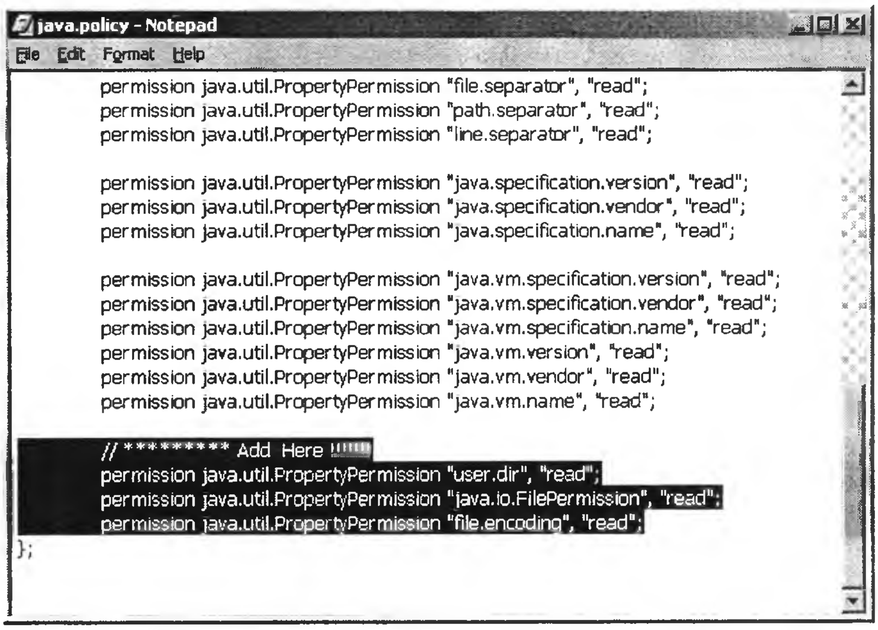
Libraries and executables for the Java 2 SDK tools.

Space Required on C:	47480 K
Space Available on C:	1500380 K

Figure H.3. Java 2 SDK, SE v1.4.0 Installation.

JDK Environment Setup

- Copy all files in the folder "lib" to "c:\Program Files\Java\j2re1.4.0\lib".
- Grant permission in accessing XML documents to applets by editing the file "java.policy" residing in the folder "c:\Program Files\Java\j2re1.4.0\lib\security" as follows:



```
        permission java.util.PropertyPermission "file.separator", "read";
        permission java.util.PropertyPermission "path.separator", "read";
        permission java.util.PropertyPermission "line.separator", "read";

        permission java.util.PropertyPermission "java.specification.version", "read";
        permission java.util.PropertyPermission "java.specification.vendor", "read";
        permission java.util.PropertyPermission "java.specification.name", "read";

        permission java.util.PropertyPermission "java.vm.specification.version", "read";
        permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
        permission java.util.PropertyPermission "java.vm.specification.name", "read";
        permission java.util.PropertyPermission "java.vm.version", "read";
        permission java.util.PropertyPermission "java.vm.vendor", "read";
        permission java.util.PropertyPermission "java.vm.name", "read";

        // ********* Add Here !!!!!
        permission java.util.PropertyPermission "user.dir", "read";
        permission java.util.PropertyPermission "java.io.FilePermission", "read";
        permission java.util.PropertyPermission "file.encoding", "read";
};
```

Figure H. 4. Editing file "java.policy".

BIOGRAPHY

Name: Mrs. Ngamnij Arch-int

Position: Assistant Professor

Address: Department of Computer Science
Faculty of Science
Khon Kaen University 40002
Thailand

Email : ngamnij@kku.ac.th

Born: July 27, 1966 in Bangkok, Thailand

Education: 1988 B.S. (Statistics), Kasetsart University, Bangkok,Thailand

1990 M.S. (Applied Statistics) (Major: Computer Science) National Institute of Development Administration (NIDA), Bangkok, Thailand

Research Interests: Agent Technology, XML Technology, Integrating of Heterogeneous Information Sources, Semantic Knowledge Representation with Ontologies, and Semantic Web.

Publications:

- **Journals**

Arch-int N. and Sophatsathit P., "A Semantic Information Gathering Approach for Heterogeneous Information Source on WWW," Journal of Information Science 29,5 (October 2003): 357-374.

Arch-int N, Sophatsathit P, and Li Y, "Ontology-based Metadata Dictionary for Integrating Heterogeneous Information Sources on the WWW," Journal of Research and Practice in Information Technology 35,4 (November 2003): 285-302.

- **Proceedings**

Arch-int N. and Sophatsathit P., "A Reference Architecture for Integrating Heterogeneous Information Sources using XML and Agent Model," Proceedings of the Joint Conference on Information Sciences, NC, USA (8-14 March 2002): 235-239.

Arch-int N., Li Y., Roe P., and Sophatsathit P. "Query Processing the Heterogeneous Information Sources using Ontology-based Approach," Proceedings of the 18th International Conference on Computers and Their Applications, Honolulu, Hawaii, USA (26-28 March, 2003): 438-441.