รายการอ้างอิง

1. Elmasri, R. and Navathe, S. B. Fundamentals of database systems. 4th ed. Reading, MA: Addison-Wesley, 2003.
2. Lenzerini, M. Data integration: A theoretical perspective. Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2002), Madison, WI, June 3-5, 2002, pp. 233-246.
3. Ram, S. and Park, J. Semantic conflict resolution ontology (SCROL): An ontology for detecting and resolving data and schema-level semantic conflicts. IEEE Transactions on Knowledge and Data Engineering 16, 2, (February 2004): 189-202.
4. Gruber, T. R. A translation approach to portable ontology specifications. Knowledge Acquisition 5, 2, (June 1993): 199-220.
5. 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. Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Workshop: Ontologies and Information Sharing, Seattle, WA, August 4-6, 2001, pp. 108-117.
6. Arens, Y., Hsu, C. -N. and Knoblock, C. A. Query processing in the SIMS information mediator. In Tate, A. (ed.), Advanced Planning Technology, pp. 61--69. Menlo Park, CA: AAAI Press, 1996.
7. Cruz, I. F. and Rajendran, A. Semantic data integration in hierarchical domains. IEEE Intelligent Systems 18, 2, (2003): 66-73.
8. Mena, E. Illarramendi, A. Kashyap, V. and Sheth, A. P. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. DISTRIBUTED AND PARALLEL DATABASES 8, 2, (April 2000): 223-271.
9. Goh, C. H., Bressan, S., Madnick, S. and Siegel, M. Context interchange: New features and formalisms for the intelligent integration of information. ACM Transactions on Information Systems 17, 3, (July 1999): 270-293.
10. Wache, H., Scholz, T., Stieghahn, H. and Konig-Ries, B. An integration method for the specification of rule-oriented mediators. Proceedings of International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), Kyoto, Japan, November 28-30, 1999, pp. 109-112.

11. Pinto, H. S. and Martims, J. P. A methodology for ontology integration. Proceedings of the 1st International Conference on Knowledge Capture (K-CAP'01), British Columbia, Canada, October 22 – 23, 2001, pp. 131-138.

12. D'Antonio, F., Lenzerini, M., Jeusfeld, M. and Johannesson, P. Deliverable D8.1: State of the art and state of the practice including initial possible research orientations [PDF file]. INTEROP: Interoperability Research for Networked Enterprises Applications and Software, Version 1.2, November 18, 2004. Available from: http://interop-noe.org/deliv/D8/attach/D8.1.pdf [February 20, 2006]

13. Noy, N. F. Semantic integration: A survey of ontology-based approaches. ACM SIGMOD Record, Special Issue on Semantic Integration 33, 4, (December 2004): 65-70.

14. Zipparo, M., Turgut, D. and Bölöni, L. A survey of merging techniques and tools for ontologies. Proceedings of The 2004 International Conference on Information and Knowledge Engineering (IKE'04), Las Vegas, NV, June 21-24, 2004.

15. Abels, S., Haak, L. and Hahn, A. Identifying ontology integration methods and their applicability in the context of product classification and knowledge integration task. Technical Report of Department of Business Information Systems, University of Oldenburg, Oldenburg, Germany, Report No. WI-OL-TR-01-2005, October, 2005.

16. Noy, N. F. and Musen, M. A. PROMPT: Algorithm and tool for automated ontology merging and alignment. Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, August 2000, pp. 450-455.

17. McGuinness, D. L., Fikes, R., Rice, J. and Wilder, S. An environment for merging and testing large ontologies. Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR 2000), Breckenridge, CO, April 12-15, 2000, pp. 483-493.

18. Stumme, G. and Maedche, A. FCA-MERGE: Bottom-up merging of ontologies. Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Seattle, WA, August 4-10, 2001, pp. 225-230.

19. McComb, D. Semantics in business systems: The savvy managers guide. Amsterdam, Netherlands: Morgan Kaufmann, 2004.

20. สุพัตรา สวัสดิ์รักเกียรติ. วิธีการรวมแบบจำลองข้อมูลเชิงวัตถุโดยใช้วิทยาการศึกษาสำนึกและการวิเคราะห์ความสัมพันธ์ระหว่างคลาส. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2542.

21. DSTC. Ontology Definition MetaModel: DSTC initial submission. Submitted to the ADTF's RFP for an Ontology Definition Metamodel (ad/03-03-40), Object Management Group, Inc. (OMG), ad/2003-08-01, August 18, 2003.

22. ศิรินทร เอนกพีระศักดิ์. การออกแบบและพัฒนาระบบพจนานุกรมข้อมูลสำหรับระบบสารสนเทศเพื่อการบริหารมหาวิทยาลัย. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2534.

23. อุรฉัตร จินตนาวงศ์. การวางแนวทางเพื่อพัฒนาระบบสารสนเทศเพื่อการจัดการทะเบียนนักศึกษาและรายวิชา. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2534.

24. จินดารัตน์ วิเศษเรืองโรจน์. การพัฒนาโมเดลของข้อมูลสำหรับงานบริหารฝ่ายบุคลากรของมหาวิทยาลัย. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2535.

25. พิบูลย์ศรี พิณรัตน์. การวิเคราะห์ระบบการเงินในเชิงข้อมูลเพื่อเสริมระบบสารสนเทศสำหรับบริหารสถาบันการศึกษา. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2535.

26. Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies (IJHCS) 43, 5-6, (November 1995): 907-928.

27. Noy, N. F. and McGuinness, D. L. Ontology development 101: A guide to creating your first ontology [PDF file]. Stanford Knowledge Systems Laboratory Technical Report and Stanford Medical Informatics Technical Report, Stanford University, Stanford, CA, Report No. KSL-01-05 and SMI-2001-0880, March, 2001. Available from: http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf [February 20, 2006]

28. Resource Description Framework (RDF): Concepts and abstract syntax [Online]. In Klyne, G. and Carroll, J. J. (eds.), W3C Recommendation 10 February 2004, Available from: http://www.w3.org/TR/rdf-concepts/ [February 20, 2006]

29. OWL Web Ontology Language guide [Online]. In Smith, M. K., Welty, C. and McGuinness, D. L. (eds.), W3C Recommendation 10 February 2004, Available from: http://www.w3.org/TR/2004/REC-owl-guide-20040210/ [February 20, 2006]

30. Bechhofer, S., Harmelen, F. V., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. and Stein, L. A. OWL Web Ontology Language reference [Online]. In

Dean, M. and Schreiber, G. (eds.), W3C Recommendation 10 February 2004, Available from: http://www.w3.org/TR/owl-ref/ [February 20, 2006]

31. Supekar, K. and Dameron, O. Advanced reasoning with OWL [Presentation]. The 2005 Protégé Conference, Available from: http://protege.stanford.edu/conference/2005/slides/T5_supekar-dameron_AdvancedOWLReasoningTutorial.pdf [February 20, 2006]
32. The Protégé ontology editor and knowledge acquisition system [Online]. Available from: http://protege.stanford.edu/ [February 20, 2006]
33. Horridge, M., Knublauch, H., Rector, A., Stevens, R. and Wroe, C. A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools [PDF file]. The University Of Manchester, Available from: http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf [February 20, 2006]
34. Knublauch, H., Musen, M. A. and Rector, A. L. Editing description logic ontologies with the Protégé OWL plugin. Proceedings of the 2004 International Workshop on Description Logics (DL2004), British Columbia, Canada, 2004.
35. Jena – A semantic web framework for Java [Online]. Available from: http://jena.sourceforge.net/ [April 20, 2006]
36. Jena2 database interface - Release notes [Online]. Hewlett-Packard Development Company, 2006. Available from: http://jena.sourceforge.net/DB/index.html [April 20, 2006]
37. Reynolds, D. Jena 2 inference support [Online]. Hewlett-Packard Development Company, LP, 2006. Available from: http://jena.sourceforge.net/inference/index.html [April 20, 2006]
38. Schoening, J. IEEE P1600.1 Standard Upper Ontology Working Group (SUO WG) home page [Online]. IEEE, 2003. Available from: http://suo.ieee.org/ [February 20, 2006]
39. Calì, A., Calvanese, D., De Giacomo, G. and Lenzerini, M. Data integration under integrity constraints. Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02), Toronto, Ontario, Canada, May 27-31, 2002, pp. 262-279.
40. MySQL AB :: The world's most popular open source database [Online]. Available from: http://www.mysql.com/ [April 20, 2006]
41. Wang, H., Horridge, M., Rector, A., Drummond, N. and Seidenberg, J. Debugging OWL-DL ontologies: A heuristic approach. Proceedings of the 4th International Semantic

Web Conference (ICWC 2005), Galway, Ireland, November 6-10, 2005, pp. 745-757.

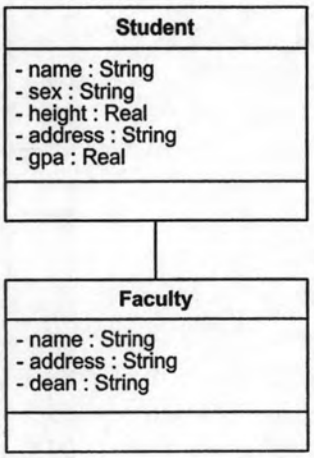
42. Racer systems GmbH & Co. KG [Online]. Available from: http://www.racer-systems.com/index.phtml [April 20, 2006]
43. Jamadhvaja, M. and Senivongse, T. An integration of data sources with UML class models based on ontological analysis. Proceedings of the 1st International ACM Workshop on Interoperability of Heterogeneous Information Systems (IHIS'05), Bremen, Germany, November 4, 2005, pp. 1-8.
44. Oracle database 10g [Online]. Available from: http://www.oracle.com/technology/products /database/oracle10g/index.html [April 20, 2006]
45. PHP: Hypertext preprocessor [Online]. Available from: http://www.php.net/ [April 20, 2006]
46. Blaha, M. and Premerlani, W. Object-oriented modeling and design for database applications. Upper Saddle River, NJ: Prentice Hall, 1998.

ภาคผนวก

ภาคผนวก ก
ตัวอย่างการรวมแบบจำลองข้อมูลคู่ที่ 1

ก.1 แบบจำลองข้อมูลเชิงวัตถุที่นำมารวม

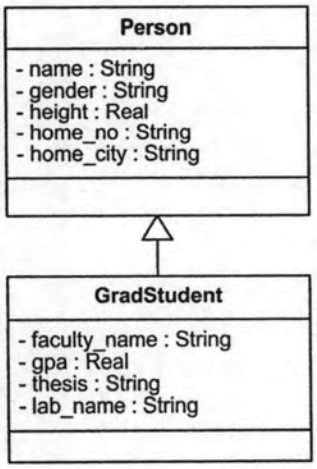
1. แบบจำลองข้อมูลเชิงวัตถุชุดที่ 1



นำมาจากแบบจำลองข้อมูลคู่ที่ 6 ในงานวิจัย [20]

Class semantics
Student: <Faculty = {'science', 'engineer'}
Major = {'computer', 'math'}>
Attribute semantics
Height: <Unit = {'centimetre'}>
Class Integrity constraint
Student: <Sex = {'F', 'M'}>

2. แบบจำลองข้อมูลเชิงวัตถุชุดที่ 2



นำมาจากแบบจำลองข้อมูลคู่ที่ 6 ในงานวิจัย [20]

Class semantics
GradStudent: <Faculty = {'science', 'account', 'engineer'}
Major = {'computer', 'stat', 'math', 'electronics'}>
Attribute semantics
Height: <Unit = {'metre'}>
Class Integrity constraint
Person: <Gender = {'female', 'male'}>

3. คำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1 และ 2

Synonym-list:
Data Model1.Student.sex = Data Model2.Person.gender
Data Model1.Student.faculty.name = Data Model2.GradStudent.faculty_name
Hypernym-list:
Data Model1.Student.address = Data Model2.Person.home_no + Data Model2.Person.home_city

ก.2 โลคัลออนโทโลจี

1. โลคัลออนโทโลจีที่ 1 สร้างจากแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://www.local-ontology.com/local6A.owl#"
  xmlns:upper="http://www.upper-ontology.com/upper.owl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
 xml:base="http://www.local-ontology.com/local6A.owl">
 <owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
 </owl:Ontology>
 <owl:Class rdf:ID="StudentSex">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Student.sex
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="math">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
   <owl:Class rdf:ID="StudentMajorSemantic"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="M">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
  <rdfs:subClassOf>
   <owl:Class rdf:ID="StudentSexConstraint"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="StudentFacultySemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
 </owl:Class>
 <owl:Class rdf:ID="science">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#StudentFacultySemantic"/>
 </owl:Class>
 <owl:Class rdf:ID="StudentFaculty">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Student.faculty
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 </owl:Class>
 <owl:Class rdf:ID="Faculty">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Faculty
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
 </owl:Class>
 <owl:Class rdf:ID="StudentHeight">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Student.height
```

```
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="StudentGpa">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model1.Student.gpa
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="FacultyName">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model1.Faculty.name
 &lt;/p>
 &lt;p style="margin-top: 0">
  Data Model1.Student.Faculty.name
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="StudentName">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model1.Student.name
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="FacultyDean">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model1.Faculty.dean
 &lt;/p>
 &lt;p style="margin-top: 0">
  Data Model1.Student.Faculty.dean
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="F">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#StudentSexConstraint"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="centimetre">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf>
  <owl:Class rdf:ID="StudentHeightUnitSemantic"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="StudentAddress">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model1.Student.address
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="engineer">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#StudentFacultySemantic"/>
</owl:Class>
<owl:Class rdf:about="#StudentMajorSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="Student">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model1.Student
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
</owl:Class>
<owl:Class rdf:ID="FacultyAddress">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model1.Faculty.address
 &lt;/p>
 &lt;p style="margin-top: 0">
```

```
  Data Model1.Student.Faculty.address
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#StudentSexConstraint">
 <owl:equivalentClass>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#M"/>
    <owl:Class rdf:about="#F"/>
   </owl:unionOf>
  </owl:Class>
 </owl:equivalentClass>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassIntegrityConstraint"/>
</owl:Class>
<owl:Class rdf:ID="computer">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#StudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:about="#StudentHeightUnitSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasStudentGpa">
 <rdfs:range rdf:resource="#StudentGpa"/>
 <rdfs:domain rdf:resource="#Student"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentSemantic">
 <rdfs:range>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#StudentFacultySemantic"/>
    <owl:Class rdf:about="#StudentMajorSemantic"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:range>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:domain rdf:resource="#Student"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFacultyAltType">
 <rdfs:domain rdf:resource="#StudentFaculty"/>
 <rdfs:range rdf:resource="#Faculty"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeAltType"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentName">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#StudentName"/>
 <rdfs:domain rdf:resource="#Student"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyDean">
 <rdfs:domain rdf:resource="#Faculty"/>
 <rdfs:range rdf:resource="#FacultyDean"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyAddress">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Faculty"/>
 <rdfs:range rdf:resource="#FacultyAddress"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentHeightSemantic">
 <rdfs:domain rdf:resource="#StudentHeight"/>
 <rdfs:range rdf:resource="#StudentHeightUnitSemantic"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFaculty">
 <rdfs:range rdf:resource="#StudentFaculty"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Student"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentAddress">
 <rdfs:domain rdf:resource="#Student"/>
 <rdfs:range rdf:resource="#StudentAddress"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentSex">
```

```
  <rdfs:range rdf:resource="#StudentSex"/>
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasStudentIntegrityConstraint">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassIntegrityConstraint"/>
  <rdfs:range rdf:resource="#StudentSexConstraint"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasStudentHeight">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#StudentHeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasFacultyName">
  <rdfs:range rdf:resource="#FacultyName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Faculty"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentAddressType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#StudentAddress"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentSexType">
  <rdfs:domain rdf:resource="#StudentSex"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasFacultyNameType">
  <rdfs:domain rdf:resource="#FacultyName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasFacultyAddressType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#FacultyAddress"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentGpaType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#StudentGpa"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasFacultyDeanType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#FacultyDean"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentNameType">
  <rdfs:domain rdf:resource="#StudentName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentHeightType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#StudentHeight"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

2. โลคัลออนโทโลจีที่ 2 สร้างจากแบบจำลองข้อมูลเชิงวัตถุชุดที่ 2

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns="http://www.local-ontology.com/local6B.owl#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.local-ontology.com/local6B.owl">
```

```
<owl:Ontology rdf:about="">
 <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
</owl:Ontology>
<owl:Class rdf:ID="PersonGender">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person.gender
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="male">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
 <rdfs:subClassOf>
  <owl:Class rdf:ID="PersonGenderConstraint"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="PersonName">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person.name
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="GradStudentMajorSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="stat">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#GradStudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:ID="female">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#PersonGenderConstraint"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GradStudentLabName">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.GradStudent.lab_name
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="account">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf>
  <owl:Class rdf:ID="GradStudentFacultySemantic"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#PersonGenderConstraint">
 <owl:equivalentClass>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#male"/>
    <owl:Class rdf:about="#female"/>
   </owl:unionOf>
  </owl:Class>
 </owl:equivalentClass>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassIntegrityConstraint"/>
</owl:Class>
<owl:Class rdf:ID="GradStudent">
 <rdfs:subClassOf>
  <owl:Class rdf:ID="Person"/>
 </rdfs:subClassOf>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.GradStudent
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PersonHomeCity">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person.home_city
 &lt;/p></rdfs:comment>
```

```
</owl:Class>
<owl:Class rdf:ID="science">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#GradStudentFacultySemantic"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="computer">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#GradStudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:about="#Person">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
</owl:Class>
<owl:Class rdf:about="#GradStudentFacultySemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="PersonHeight">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person.height
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="metre">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf>
  <owl:Class rdf:ID="PersonHeightUnitSemantic"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GradStudentFacultyName">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.GradStudent.faculty_name
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="PersonHomeNo">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person.home_no
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="GradStudentGpa">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.GradStudent.gpa
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#PersonHeightUnitSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="math">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#GradStudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:ID="electronics">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#GradStudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:ID="engineer">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#GradStudentFacultySemantic"/>
</owl:Class>
<owl:Class rdf:ID="GradStudentThesis">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.GradStudent.thesis
 &lt;/p></rdfs:comment>
```

```
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasPersonName">
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:range rdf:resource="#PersonName"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentSemantic">
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:range>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#GradStudentMajorSemantic"/>
    <owl:Class rdf:about="#GradStudentFacultySemantic"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHeightSemantic">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeSemantic"/>
 <rdfs:range rdf:resource="#PersonHeightUnitSemantic"/>
 <rdfs:domain rdf:resource="#PersonHeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentThesis">
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:range rdf:resource="#GradStudentThesis"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHomeCity">
 <rdfs:range rdf:resource="#PersonHomeCity"/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonIntegrityConstraint">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassIntegrityConstraint"/>
 <rdfs:range rdf:resource="#PersonGenderConstraint"/>
 <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonGender">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#PersonGender"/>
 <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHomeNo">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:range rdf:resource="#PersonHomeNo"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentGpa">
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#GradStudentGpa"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHeight">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:range rdf:resource="#PersonHeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentFacultyName">
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#GradStudentFacultyName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentLabName">
 <rdfs:range rdf:resource="#GradStudentLabName"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#GradStudent"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasGradStudentLabNameType">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 <rdfs:domain rdf:resource="#GradStudentLabName"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonGenderType">
```

```
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonGender"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasGradStudentThesisType">
  <rdfs:domain rdf:resource="#GradStudentThesis"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonHomeCityType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonHomeCity"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasGradStudentFacultyNameType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#GradStudentFacultyName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonHeightType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#PersonHeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonHomeNoType">
  <rdfs:domain rdf:resource="#PersonHomeNo"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonNameType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasGradStudentGpaType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#GradStudentGpa"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
</owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

3. ออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีที่ 1 และโลคัลออนโทโลจีที่ 2

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:local1="http://www.local-ontology.com/local6A.owl#"
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:local2="http://www.local-ontology.com/local6B.owl#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://www.semantic-ontology.com/semantic6.owl#"
  xml:base="http://www.semantic-ontology.com/semantic6.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.local-ontology.com/local6B.owl"/>
    <owl:imports rdf:resource="http://www.local-ontology.com/local6A.owl"/>
    <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <rdf:Description rdf:about="http://www.local-ontology.com/local6A.owl#StudentSex">
    <owl:equivalentClass>
      <rdf:Description rdf:about="http://www.local-ontology.com/local6B.owl#PersonGender">
        <owl:equivalentClass rdf:resource="http://www.local-ontology.com/local6A.owl#StudentSex"/>
      </rdf:Description>
    </owl:equivalentClass>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.local-ontology.com/local6A.owl#StudentAddress">
    <owl:equivalentClass>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.local-ontology.com/local6B.owl#PersonHomeCity"/>
          <rdf:Description rdf:about="http://www.local-ontology.com/local6B.owl#PersonHomeNo"/>
```

```
      </owl:unionOf>
     </owl:Class>
    </owl:equivalentClass>
   </rdf:Description>
   <rdf:Description rdf:about="http://www.local-ontology.com/local6B.owl#GradStudentFacultyName">
    <owl:equivalentClass>
     <rdf:Description rdf:about="http://www.local-ontology.com/local6A.owl#FacultyName">
      <owl:equivalentClass rdf:resource="http://www.local-
ontology.com/local6B.owl#GradStudentFacultyName"/>
     </rdf:Description>
    </owl:equivalentClass>
   </rdf:Description>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

ก.3 ออนโทโลจีรวม

1. ออนโทโลจีรวมพื้นฐาน (IntegratedOntology.owl)

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns="http://www.integrate-ontology.com/IntegratedOntology6.owl#"
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.integrate-ontology.com/IntegratedOntology6.owl">
  <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Faculty">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#Faculty
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="computer">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentMajorSemantic"/>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentMajorSemantic"/>
   </rdfs:subClassOf>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  </owl:Class>
  <owl:Class rdf:ID="FacultyDean">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#FacultyDean
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="science">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentFacultySemantic"/>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentFacultySemantic"/>
   </rdfs:subClassOf>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  </owl:Class>
  <owl:Class rdf:ID="GradStudentLabName">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local6B.owl#GradStudentLabName
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="engineer">
```

```
 <rdfs:subClassOf>
  <owl:Class rdf:about="#GradStudentFacultySemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#StudentFacultySemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="GradStudent">
 <rdfs:subClassOf>
  <owl:Class rdf:ID="Student"/>
 </rdfs:subClassOf>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6B.owl#GradStudent
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="StudentFaculty">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#StudentFaculty
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="metre">
 <rdfs:subClassOf>
  <owl:Class rdf:ID="PersonHeightUnitSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="PersonGender">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#StudentSex
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local6B.owl#PersonGender
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="electronics">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#GradStudentMajorSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#StudentMajorSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="FacultyAddress">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#FacultyAddress
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="GradStudentThesis">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6B.owl#GradStudentThesis
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="math">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#GradStudentMajorSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#StudentMajorSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="PersonHeight">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
>&lt;p style="margin-top: 0">
  equivalent to local6A.owl#StudentHeight
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local6B.owl#PersonHeight
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="F">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
 <rdfs:subClassOf>
  <owl:Class rdf:ID="PersonGenderConstraint"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="M">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#PersonGenderConstraint"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#StudentFacultySemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="stat">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#GradStudentMajorSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#StudentMajorSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:about="#Student">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#Student
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf>
  <owl:Class rdf:ID="Person"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#GradStudentFacultySemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="account">
 <rdfs:subClassOf rdf:resource="#GradStudentFacultySemantic"/>
 <rdfs:subClassOf rdf:resource="#StudentFacultySemantic"/>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:about="#Person">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6B.owl#Person
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PersonName">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#StudentName
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local6B.owl#PersonName
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#PersonHeightUnitSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="StudentGpa">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#StudentGpa
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local6B.owl#GradStudentGpa
```

```
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#StudentMajorSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:about="#GradStudentMajorSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="FacultyName">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#FacultyName
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local6B.owl#GradStudentFacultyName
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#PersonGenderConstraint">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassIntegrityConstraint"/>
</owl:Class>
<owl:Class rdf:ID="PersonAddress">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#StudentAddress
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local6B.owl#PersonHomeNo + local6B.owl#PersonHomeCity
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasPersonHeight">
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#PersonHeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentThesis">
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#GradStudentThesis"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonAddress">
 <rdfs:range rdf:resource="#PersonAddress"/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentSemantic">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:range>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#GradStudentFacultySemantic"/>
    <owl:Class rdf:about="#StudentMajorSemantic"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyDean">
 <rdfs:domain rdf:resource="#Faculty"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#FacultyDean"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFaculty">
 <rdfs:domain rdf:resource="#Student"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#StudentFaculty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFacultyAltType">
 <rdfs:domain rdf:resource="#StudentFaculty"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeAltType"/>
 <rdfs:range rdf:resource="#Faculty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonIntegrityConstraint">
```

```
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassIntegrityConstraint"/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:range rdf:resource="#PersonGenderConstraint"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyAddress">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Faculty"/>
 <rdfs:range rdf:resource="#FacultyAddress"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyName">
 <rdfs:range rdf:resource="#FacultyName"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Faculty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonGender">
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:range rdf:resource="#PersonGender"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHeightSemantic">
 <rdfs:range rdf:resource="#PersonHeightUnitSemantic"/>
 <rdfs:domain rdf:resource="#PersonHeight"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentSemantic">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:domain rdf:resource="#Student"/>
 <rdfs:range>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#StudentFacultySemantic"/>
    <owl:Class rdf:about="#StudentMajorSemantic"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentGpa">
 <rdfs:domain rdf:resource="#Student"/>
 <rdfs:range rdf:resource="#StudentGpa"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#PersonName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentLabName">
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:range rdf:resource="#GradStudentLabName"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasPersonAddressType">
 <rdfs:domain rdf:resource="#PersonAddress"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasStudentGpaType">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 <rdfs:domain rdf:resource="#StudentGpa"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasFacultyDeanType">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 <rdfs:domain rdf:resource="#FacultyDean"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonHeightType">
 <rdfs:domain rdf:resource="#PersonHeight"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonGenderType">
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 <rdfs:domain rdf:resource="#PersonGender"/>
```

```
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasGradStudentThesisType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#GradStudentThesis"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasFacultyAddressType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#FacultyAddress"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasGradStudentLabNameType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#GradStudentLabName"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonNameType">
  <rdfs:domain rdf:resource="#PersonName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasFacultyNameType">
  <rdfs:domain rdf:resource="#FacultyName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

2. ผลการรวมออนโทโลจี (IntegratedOntology.txt)

```
[Class] : Person (equivalent to local6B.owl#Person)
<subClassOf> upper.owl#Class
<hasClassAttribute>
     [ClassAttribute] : PersonName (equivalent to local6A.owl#StudentName, local6B.owl#PersonName)
     <subClassOf> upper.owl#ClassAttribute
     <hasClassAttributeType> String
     Type conversion : no
     [ClassAttribute] : PersonGender (equivalent to local6A.owl#StudentSex, local6B.owl#PersonGender)
     <subClassOf> upper.owl#ClassAttribute
     <hasClassAttributeType> String
     Type conversion : no
     [ClassAttribute] : PersonHeight (equivalent to local6A.owl#StudentHeight, local6B.owl#PersonHeight)
     <subClassOf> upper.owl#ClassAttribute
     <hasClassAttributeType> String
     Type conversion : no
     <hasClassAttributeSemantic>
          [SemanticName] : PersonHeightUnitSemantic
          <subClassOf> upper.owl#SemanticName
               [SemanticWord] : metre
               <subClassOf> upper.owl#SemanticWord, PersonHeightUnitSemantic
     [ClassAttribute] : PersonAddress (equivalent to local6A.owl#StudentAddress, local6B.owl#PersonHomeNo
+ local6B.owl#PersonHomeCity)
     <hasClassAttributeType> String
     Type conversion : no
<hasClassIntegrityConstraint>
     [ClassIntegrityConstraint] : PersonGenderConstraint
     <subClassOf> upper.owl#ClassIntegrityConstraint
          [ConstraintValue] : F
          <subClassOf> upper.owl#ConstraintValue, PersonGenderConstraint
          Mapping function : yes
          [ConstraintValue] : M
          <subClassOf> upper.owl#ConstraintValue, PersonGenderConstraint
          Mapping function : yes
<hasClassSemantic> none

[Class] : Student (equivalent to local6A.owl#Student)
<subClassOf> upper.owl#Class, Person
<hasClassAttribute>
     [ClassAttribute] : StudentGpa (equivalent to local6A.owl#StudentGpa, local6B.owl#GradStudentGpa)
     <subClassOf> upper.owl#ClassAttribute
     <hasClassAttributeType> Float
     Type conversion : no
     [ClassAttribute] : StudentFaculty (equivalent to local6A.owl#StudentFaculty)
```

<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeAltType> Faculty
Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
[SemanticName] : StudentFacultySemantic
<subClassOf> upper.owl#SemanticName
[SemanticWord] : science
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticWord] : account
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticWord] : engineer
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticName] : StudentMajorSemantic
<subClassOf> upper.owl#SemanticName
[SemanticWord] : computer
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : stat
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : electronics
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : math
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic

[Class] : GradStudent (equivalent to local6B.owl#GradStudent)
<subClassOf> upper.owl#Class, Student
<hasClassAttribute>
[ClassAttribute] : GradStudentThesis (equivalent to local6B.owl#GradStudentThesis)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : no
[ClassAttribute] : GradStudentLabName (equivalent to local6B.owl#GradStudentLabName)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
[SemanticName] : GradStudentFacultySemantic
<subClassOf> upper.owl#SemanticName
[SemanticWord] : science
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticWord] : account
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticWord] : engineer
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticName] : GradStudentMajorSemantic
<subClassOf> upper.owl#SemanticName
[SemanticWord] : computer
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : stat
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : electronics
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : math
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic

[Class] : Faculty (equivalent to local6A.owl#Faculty)
<subClassOf> upper.owl#Class
<hasClassAttribute>
[ClassAttribute] : FacultyName (equivalent to local6A.owl#FacultyName,
local6B.owl#GradStudentFacultyName)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : yes
[ClassAttribute] : FacultyAddress (equivalent to local6A.owl#FacultyAddress)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : no
[ClassAttribute] : FacultyDean (equivalent to local6A.owl#FacultyDean)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic> none

ก.4 ตัวอย่างวิวที่กำกับลงบนฐานข้อมูล

1. วิวที่กำกับลงบนแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

```
CREATE FUNCTION Sex_to_Gender (old_sex in VARCHAR2)
    RETURN VARCHAR2
    IS new_sex          VARCHAR2(50);
BEGIN
    IF (old_sex='F') THEN
        new_sex:='female';
    ELSIF (old_sex='M') THEN
        new_sex:='male';
    END IF;
    RETURN new_sex;
END;

CREATE FUNCTION Height_to_Metre (old_height in NUMBER)
    RETURN NUMBER
    IS new_height NUMBER;
BEGIN
    new_height:= old_height /100;
    RETURN new_height;
END;

CREATE OR REPLACE VIEW Person(name, gender, height, address)
    AS
    SELECT a.name, SEX_TO_GENDER(a.sex), Height_to_Metre(a.height), a.address
    FROM StudentTab a;

CREATE OR REPLACE VIEW Student(name, gender, height, address, gpa, faculty_name, faculty_address,
faculty_dean)
    AS
SELECT a.name, SEX_TO_GENDER(a.sex), Height_to_Metre(a.height), a.address, a.gpa, a.faculty.name,
a.faculty.address, a.faculty.dean
    FROM StudentTab a;

CREATE OR REPLACE VIEW Faculty(name, address, dean)
    AS
    SELECT a.name, a.address, a.dean
    FROM FacultyTab a;
```

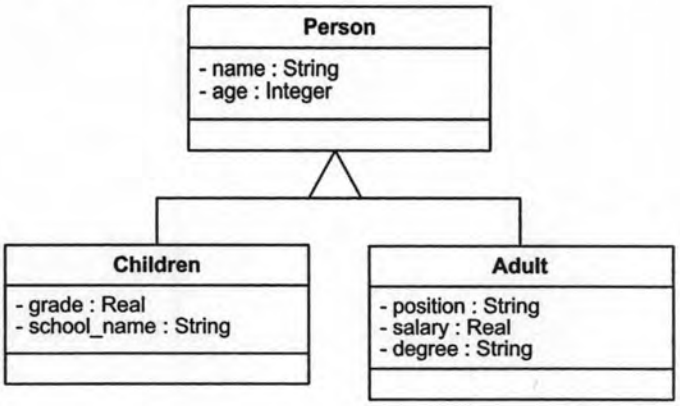
2. วิวที่กำกับลงบนแบบจำลองข้อมูลเชิงวัตถุชุดที่ 2

```
CREATE OR REPLACE VIEW Person(name, gender, height, address)
    AS
    SELECT a.name, a.gender, a.height, a.home_no || ', ' || a.home_city
    FROM PersonTab a
    UNION
SELECT b.person.name, b.person.gender, b.person.height, b.person.home_no || ', ' || b.person.home_city
    FROM GradStudentTab b;

CREATE OR REPLACE VIEW Student(name, gender, height, address, gpa, faculty_name, faculty_address,
faculty_dean)
    AS
SELECT a.person.name, a.person.gender, a.person.height, a.person.home_no || ', ' || a.person.home_city,
a.gpa, a.faculty_name, '—', '—'
    FROM GradStudentTab a;

CREATE OR REPLACE VIEW GradStudent(name, gender, height, address, gpa, faculty_name,
faculty_address, faculty_dean, thesis, lab_name)
    AS
SELECT a.person.name, a.person.gender, a.person.height, a.person.home_no || ', ' || a.person.home_city,
a.gpa, a.faculty_name, '—', '—', a.thesis, a.lab_name
    FROM GradStudentTab a;

CREATE OR REPLACE VIEW Faculty(name, address, dean)
    AS
    SELECT a.faculty_name, '—', '—'
    FROM GradStudentTab a;
```

ภาคผนวก ข
ตัวอย่างการรวมแบบจำลองข้อมูลคู่ที่ 2

ข.1 แบบจำลองข้อมูลเชิงวัตถุที่นำมารวม

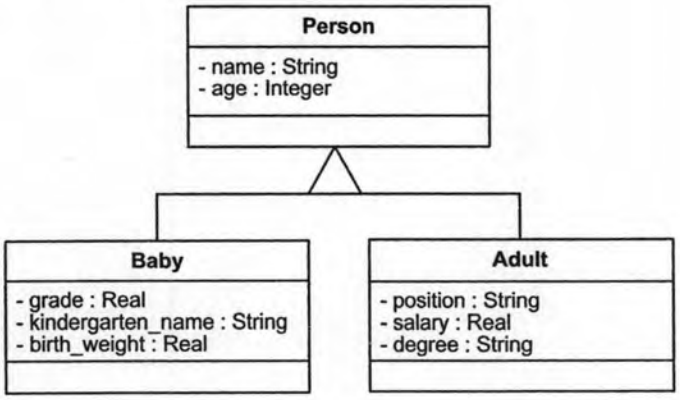
1. แบบจำลองข้อมูลเชิงวัตถุชุดที่ 1



นำมาจากแบบจำลองข้อมูลคู่ที่ 4 ในงานวิจัย [20]

Class semantics
Children: <Age = [0, 20]>
Adult: <Age = [30, 150]>

2. แบบจำลองข้อมูลเชิงวัตถุชุดที่ 2



นำมาจากแบบจำลองข้อมูลคู่ที่ 4 ในงานวิจัย [20]

Class semantics
Baby: <Age = [0, 2]>
Adult: <Age = [20, 120]>

3. คำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1 และ 2

Synonym-list:
Data Model1.Childten.school_name = Data Model2.Baby.kindergarten_name

ข.2 โลคัลออนโทโลจี

1. โลคัลออนโทโลจีที่ 1 สร้างจากแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:upper="http://www.upper-ontology.com/upper.owl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.local-ontology.com/local4A.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
 xml:base="http://www.local-ontology.com/local4A.owl">
 <owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
 </owl:Ontology>
 <owl:Class rdf:ID="Adult">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Adult
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Person"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="AdultSalary">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Adult.salary
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="GreaterThanOrEqual30">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="AdultAgeSemantic"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:about="#Person">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Person
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
 </owl:Class>
 <owl:Class rdf:ID="AdultDegree">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Adult.degree
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 </owl:Class>
 <owl:Class rdf:ID="LessThanOrEqual20">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="ChildrenAgeSemantic"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:about="#AdultAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
 </owl:Class>
 <owl:Class rdf:ID="PersonName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Person.name
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="AdultPosition">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Adult.position
```

```
 &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#ChildrenAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="PersonAge">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Person.age
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Children">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Children
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="ChildrenSchoolName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Children.school_name
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="ChildrenGrade">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Children.grade
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual0">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual150">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#AdultAgeSemantic"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasChildrenGrade">
  <rdfs:range rdf:resource="#ChildrenGrade"/>
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultPosition">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:range rdf:resource="#AdultPosition"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSemantic">
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:range rdf:resource="#ChildrenAgeSemantic"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSemantic">
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:range rdf:resource="#AdultAgeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSchoolName">
  <rdfs:range rdf:resource="#ChildrenSchoolName"/>
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSalary">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:range rdf:resource="#AdultSalary"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#PersonName"/>
```

```
  <rdfs:domain rdf:resource="#Person"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasAdultDegree">
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#AdultDegree"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasPersonAge">
  <rdfs:range rdf:resource="#PersonAge"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonAgeType">
  <rdfs:domain rdf:resource="#PersonAge"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultSalaryType">
  <rdfs:domain rdf:resource="#AdultSalary"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultDegreeType">
  <rdfs:domain rdf:resource="#AdultDegree"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultPositionType">
  <rdfs:domain rdf:resource="#AdultPosition"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasChildrenGradeType">
  <rdfs:domain rdf:resource="#ChildrenGrade"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasChildrenSchoolNameType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#ChildrenSchoolName"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonNameType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonName"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

2. โลคัลออนโทโลจีที่ 2 สร้างจากแบบจำลองข้อมูลเชิงวัตถุชุดที่ 2

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://www.local-ontology.com/local4B.owl#"
  xml:base="http://www.local-ontology.com/local4B.owl">
 <owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
 </owl:Ontology>
 <owl:Class rdf:ID="Adult">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Adult
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf>
   <owl:Class rdf:ID="Person"/>
  </rdfs:subClassOf>
 </owl:Class>
```

```
<owl:Class rdf:ID="LessThanOrEqual20">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf>
  <owl:Class rdf:ID="AdultAgeSemantic"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="PersonAge">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person.age
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#AdultAgeSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual20">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#AdultAgeSemantic"/>
</owl:Class>
<owl:Class rdf:ID="AdultPosition">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Adult.position
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="BabyKindergartenName">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Baby.kindergarten_name
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="BabyAgeSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="BabyBirthWeight">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Baby.birth_weight
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual2">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#BabyAgeSemantic"/>
</owl:Class>
<owl:Class rdf:about="#Person">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual0">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#BabyAgeSemantic"/>
</owl:Class>
<owl:Class rdf:ID="AdultSalary">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Adult.salary
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="BabyGrade">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Baby.grade
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="AdultDegree">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
 >&lt;p style="margin-top: 0">
  Data Model2.Adult.degree
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="PersonName">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Person.name
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Baby">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  Data Model2.Baby
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasBabyKindergartenName">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Baby"/>
 <rdfs:range rdf:resource="#BabyKindergartenName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultDegree">
 <rdfs:range rdf:resource="#AdultDegree"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:range rdf:resource="#PersonName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabySemantic">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:range rdf:resource="#BabyAgeSemantic"/>
 <rdfs:domain rdf:resource="#Baby"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultPosition">
 <rdfs:domain rdf:resource="#Adult"/>
 <rdfs:range rdf:resource="#AdultPosition"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabyGrade">
 <rdfs:domain rdf:resource="#Baby"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#BabyGrade"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonAge">
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#PersonAge"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabyBirthWeight">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Baby"/>
 <rdfs:range rdf:resource="#BabyBirthWeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSemantic">
 <rdfs:range rdf:resource="#AdultAgeSemantic"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSalary">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#AdultSalary"/>
 <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasBabyBirthWeightType">
 <rdfs:domain rdf:resource="#BabyBirthWeight"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasBabyKindergartenNameType">
```

```
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#BabyKindergartenName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasBabyGradeType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#BabyGrade"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonAgeType">
  <rdfs:domain rdf:resource="#PersonAge"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultDegreeType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#AdultDegree"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultSalaryType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#AdultSalary"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultPositionType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#AdultPosition"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonNameType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonName"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

3. ออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีที่ 1 และโลคัลออนโทโลจีที่ 2

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:local1="http://www.local-ontology.com/local4A.owl#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:local2="http://www.local-ontology.com/local4B.owl#"
   xmlns="http://www.semantic-ontology.com/semantic4.owl#"
  xml:base="http://www.semantic-ontology.com/semantic4.owl">
  <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://www.local-ontology.com/local4B.owl"/>
   <owl:imports rdf:resource="http://www.local-ontology.com/local4A.owl"/>
   <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <rdf:Description rdf:about="http://www.local-ontology.com/local4A.owl#ChildrenSchoolName">
   <owl:equivalentClass>
    <rdf:Description rdf:about="http://www.local-ontology.com/local4B.owl#BabyKindergartenName">
     <owl:equivalentClass rdf:resource="http://www.local-ontology.com/local4A.owl#ChildrenSchoolName"/>
    </rdf:Description>
   </owl:equivalentClass>
  </rdf:Description>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

ข.3 ออนโทโลจีรวม

1. ออนโทโลจีรวมพื้นฐาน (IntegratedOntology.owl)

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:upper="http://www.upper-ontology.com/upper.owl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.integrate-ontology.com/IntegratedOntology4.owl#"
 xml:base="http://www.integrate-ontology.com/IntegratedOntology4.owl">
 <owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
 </owl:Ontology>
 <owl:Class rdf:ID="ChildrenAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
 </owl:Class>
 <owl:Class rdf:ID="Children">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
   equivalent to local4A.owl#Children
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf>
   <owl:Class rdf:ID="Person"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="LessThanOrEqual20">
  <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 </owl:Class>
 <owl:Class rdf:ID="PersonName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
   equivalent to local4A.owl#PersonName
  &lt;/p>
  &lt;p style="margin-top: 0">
   equivalent to local4B.owl#PersonName
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="ChildrenGrade">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
   equivalent to local4A.owl#ChildrenGrade
  &lt;/p>
  &lt;p style="margin-top: 0">
   equivalent to local4B.owl#BabyGrade
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="AdultSalary">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
   equivalent to local4A.owl#AdultSalary
  &lt;/p>
  &lt;p style="margin-top: 0">
   equivalent to local4B.owl#AdultSalary
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="AdultPosition">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
   equivalent to local4A.owl#AdultPosition
  &lt;/p>
  &lt;p style="margin-top: 0">
   equivalent to local4B.owl#AdultPosition
  &lt;/p></rdfs:comment>
 </owl:Class>
```

```
<owl:Class rdf:ID="GreaterThanOrEqual0">
 <rdfs:subClassOf>
  <owl:Class rdf:ID="BabyAgeSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual150">
 <rdfs:subClassOf>
  <owl:Class rdf:ID="AdultAgeSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="BabyBirthWeight">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4B.owl#BabyBirthWeight
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Baby">
 <rdfs:subClassOf rdf:resource="#Children"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4B.owl#Baby
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="ChildrenSchoolName">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4A.owl#ChildrenSchoolName
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local4B.owl#BabyKindergartenName
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#Person">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4A.owl#Person
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local4B.owl#Person
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PersonAge">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4A.owl#PersonAge
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local4B.owl#PersonAge
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="AdultDegree">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4A.owl#AdultDegree
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local4B.owl#AdultDegree
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="Adult">
 <rdfs:subClassOf rdf:resource="#Person"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4A.owl#Adult
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local4B.owl#Adult
```

```
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#AdultAgeSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual2">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#BabyAgeSemantic"/>
 </rdfs:subClassOf>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual20">
 <rdfs:subClassOf rdf:resource="#AdultAgeSemantic"/>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:about="#BabyAgeSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasChildrenGrade">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Children"/>
 <rdfs:range rdf:resource="#ChildrenGrade"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSemantic">
 <rdfs:domain rdf:resource="#Adult"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:range rdf:resource="#AdultAgeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSemantic">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:range rdf:resource="#ChildrenAgeSemantic"/>
 <rdfs:domain rdf:resource="#Children"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#PersonName"/>
 <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonAge">
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:range rdf:resource="#PersonAge"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabySemantic">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:domain rdf:resource="#Baby"/>
 <rdfs:range rdf:resource="#BabyAgeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultPosition">
 <rdfs:range rdf:resource="#AdultPosition"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultDegree">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#AdultDegree"/>
 <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabyBirthWeight">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Baby"/>
 <rdfs:range rdf:resource="#BabyBirthWeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSalary">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Adult"/>
 <rdfs:range rdf:resource="#AdultSalary"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSchoolName">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#ChildrenSchoolName"/>
 <rdfs:domain rdf:resource="#Children"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasAdultPositionType">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
```

```
  <rdfs:domain rdf:resource="#AdultPosition"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultDegreeType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#AdultDegree"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasChildrenGradeType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#ChildrenGrade"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonNameType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#PersonName"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasBabyBirthWeightType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#BabyBirthWeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonAgeType">
  <rdfs:domain rdf:resource="#PersonAge"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultSalaryType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#AdultSalary"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasChildrenSchoolNameType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#ChildrenSchoolName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

2. ผลการรวมออนโทโลจี (IntegratedOntology.txt)

```
[Class] : Person (equivalent to local4A.owl#Person, local4B.owl#Person)
<subClassOf> upper.owl#Class
<hasClassAttribute>
      [ClassAttribute] : PersonName (equivalent to local4A.owl#PersonName, local4B.owl#PersonName)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> String
      Type conversion : no
      [ClassAttribute] : PersonAge (equivalent to local4A.owl#PersonAge, local4B.owl#PersonAge)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> Int
      Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic> none

[Class] : Children (equivalent to local4A.owl#Children)
<subClassOf> upper.owl#Class, Person
<hasClassAttribute>
      [ClassAttribute] : ChildrenGrade (equivalent to local4A.owl#ChildrenGrade, local4B.owl#BabyGrade)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> Float
      Type conversion : no
      [ClassAttribute] : ChildrenSchoolName (equivalent to local4A.owl#ChildrenSchoolName,
local4B.owl#BabyKindergartenName)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeAltType> String
      Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
      [SemanticName] : ChildrenAgeSemantic
      <subClassOf> upper.owl#SemanticName
            [SemanticWord] : GreaterThanOrEqual0
```

```
        <subClassOf> upper.owl#SemanticWord, ChildrenAgeSemantic, BabyAgeSemantic
        [SemanticWord] : LessThanOrEqual20
        <subClassOf> upper.owl#SemanticWord, ChildrenAgeSemantic

[Class] : Baby (equivalent to local4B.owl#Baby)
<subClassOf> upper.owl#Class, Children
<hasClassAttribute>
    [ClassAttribute] : BabyBirthWeight (equivalent to local4B.owl#BabyBirthWeight)
    <subClassOf> upper.owl#ClassAttribute
    <hasClassAttributeType> Float
    Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
    [SemanticName] : BabyAgeSemantic
    <subClassOf> upper.owl#SemanticName
        [SemanticWord] : GreaterThanOrEqual0
        <subClassOf> upper.owl#SemanticWord, ChildrenAgeSemantic, BabyAgeSemantic
        [SemanticWord] : LessThanOrEqual2
        <subClassOf> upper.owl#SemanticWord, BabyAgeSemantic

[Class] : Adult (equivalent to local4A.owl#Adult, local4B.owl#Adult)
<subClassOf> upper.owl#Class, Person
<hasClassAttribute>
    [ClassAttribute] : AdultPosition (equivalent to equivalent to local4A.owl#AdultPosition,
local4B.owl#AdultPosition)
    <subClassOf> upper.owl#ClassAttribute
    <hasClassAttributeType> String
    Type conversion : no
    [ClassAttribute] : AdultSalary (equivalent to local4A.owl#AdultSalary, local4B.owl#AdultSalary)
    <subClassOf> upper.owl#ClassAttribute
    <hasClassAttributeType> Float
    Type conversion : no
    [ClassAttribute] : AdultDegree (equivalent to local4A.owl#AdultDegree, local4B.owl#AdultDegree)
    <subClassOf> upper.owl#ClassAttribute
    <hasClassAttributeType> String
    Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
    [SemanticName] : AdultAgeSemantic
    <subClassOf> upper.owl#SemanticName
        [SemanticWord] : GreaterThanOrEqual20
        <subClassOf> upper.owl#SemanticWord, AdultAgeSemantic
        [SemanticWord] : LessThanOrEqual150
        <subClassOf> upper.owl#SemanticWord, AdultAgeSemantic
```

3. ออนโทโลจีรวมที่ผ่านการอนุมานแล้ว

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:upper="http://www.upper-ontology.com/upper.owl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.integrate-ontology.com/IntegratedOntology4.owl#"
 xml:base="http://www.integrate-ontology.com/IntegratedOntology4.owl">
 <owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
 </owl:Ontology>
 <owl:Class rdf:ID="ChildrenAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
 </owl:Class>
 <owl:Class rdf:ID="Children">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty>
     <owl:ObjectProperty rdf:ID="hasPersonAge"/>
    </owl:onProperty>
    <owl:someValuesFrom>
     <owl:Class rdf:ID="LessThanOrEqual20"/>
    </owl:someValuesFrom>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
```

```
    <owl:Restriction>
     <owl:onProperty>
      <owl:ObjectProperty rdf:about="#hasPersonAge"/>
     </owl:onProperty>
     <owl:someValuesFrom>
      <owl:Class rdf:ID="LessThanOrEqual2"/>
     </owl:someValuesFrom>
    </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
    <owl:Restriction>
     <owl:onProperty>
      <owl:ObjectProperty rdf:about="#hasPersonAge"/>
     </owl:onProperty>
     <owl:someValuesFrom>
      <owl:Class rdf:ID="GreaterThanOrEqual0"/>
     </owl:someValuesFrom>
    </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Person"/>
   </rdfs:subClassOf>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#Children
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="#LessThanOrEqual20">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
   <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
  </owl:Class>
  <owl:Class rdf:ID="ChildrenGrade">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#ChildrenGrade
   &lt;/p>
   &lt;p style="margin-top: 0">
    equivalent to local4B.owl#BabyGrade
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="PersonName">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#PersonName
   &lt;/p>
   &lt;p style="margin-top: 0">
    equivalent to local4B.owl#PersonName
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="AdultSalary">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#AdultSalary
   &lt;/p>
   &lt;p style="margin-top: 0">
    equivalent to local4B.owl#AdultSalary
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  </owl:Class>
  <owl:Class rdf:ID="AdultPosition">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#AdultPosition
   &lt;/p>
   &lt;p style="margin-top: 0">
    equivalent to local4B.owl#AdultPosition
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="#GreaterThanOrEqual0">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
   <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
   <rdfs:subClassOf>
```

```
    <owl:Class rdf:ID="BabyAgeSemantic"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="LessThanOrEqual150">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
   <rdfs:subClassOf>
    <owl:Class rdf:ID="AdultAgeSemantic"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="BabyBirthWeight">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4B.owl#BabyBirthWeight
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="Baby">
   <owl:equivalentClass>
    <owl:Class>
     <owl:intersectionOf rdf:parseType="Collection">
      <owl:Restriction>
       <owl:someValuesFrom rdf:resource="#GreaterThanOrEqual0"/>
       <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasPersonAge"/>
       </owl:onProperty>
      </owl:Restriction>
      <owl:Restriction>
       <owl:someValuesFrom>
        <owl:Class rdf:about="#LessThanOrEqual2"/>
       </owl:someValuesFrom>
       <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasPersonAge"/>
       </owl:onProperty>
      </owl:Restriction>
     </owl:intersectionOf>
    </owl:Class>
   </owl:equivalentClass>
   <rdfs:subClassOf rdf:resource="#Children"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4B.owl#Baby
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="ChildrenSchoolName">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#ChildrenSchoolName
   &lt;/p>
   &lt;p style="margin-top: 0">
    equivalent to local4B.owl#BabyKindergartenName
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="#Person">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#Person
   &lt;/p>
   &lt;p style="margin-top: 0">
    equivalent to local4B.owl#Person
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
  </owl:Class>
  <owl:Class rdf:ID="AdultDegree">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#AdultDegree
   &lt;/p>
   &lt;p style="margin-top: 0">
    equivalent to local4B.owl#AdultDegree
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  </owl:Class>
  <owl:Class rdf:ID="PersonAge">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
```

```
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4A.owl#PersonAge
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local4B.owl#PersonAge
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Adult">
 <rdfs:subClassOf rdf:resource="#Person"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local4A.owl#Adult
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local4B.owl#Adult
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#AdultAgeSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual20">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf rdf:resource="#AdultAgeSemantic"/>
</owl:Class>
<owl:Class rdf:about="#LessThanOrEqual2">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#BabyAgeSemantic"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#BabyAgeSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasChildrenGrade">
 <rdfs:range rdf:resource="#ChildrenGrade"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Children"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSemantic">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:range rdf:resource="#AdultAgeSemantic"/>
 <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSemantic">
 <rdfs:range rdf:resource="#ChildrenAgeSemantic"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:domain rdf:resource="#Children"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
 <rdfs:range rdf:resource="#PersonName"/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasPersonAge">
 <rdfs:range rdf:resource="#PersonAge"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabySemantic">
 <rdfs:domain rdf:resource="#Baby"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:range rdf:resource="#BabyAgeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultPosition">
 <rdfs:domain rdf:resource="#Adult"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#AdultPosition"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultDegree">
 <rdfs:domain rdf:resource="#Adult"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#AdultDegree"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabyBirthWeight">
```

```
  <rdfs:domain rdf:resource="#Baby"/>
  <rdfs:range rdf:resource="#BabyBirthWeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasAdultSalary">
  <rdfs:range rdf:resource="#AdultSalary"/>
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasChildrenSchoolName">
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#ChildrenSchoolName"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultPositionType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#AdultPosition"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultDegreeType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#AdultDegree"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasChildrenGradeType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#ChildrenGrade"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonNameType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasBabyBirthWeightType">
  <rdfs:domain rdf:resource="#BabyBirthWeight"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonAgeType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#PersonAge"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultSalaryType">
  <rdfs:domain rdf:resource="#AdultSalary"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasChildrenSchoolNameType">
  <rdfs:domain rdf:resource="#ChildrenSchoolName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

ข.4 ตัวอย่างวิวที่กำกับลงบนฐานข้อมูล

1. วิวที่กำกับลงบนแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

```
CREATE OR REPLACE VIEW Person(name, age)
    AS
    SELECT a.name, a.age
    FROM PersonTab a
    UNION
    SELECT b.person.name, b.person.age
    FROM ChildrenTab b
    UNION
    SELECT c.person.name, c.person.age
    FROM AdultTab c;

CREATE OR REPLACE VIEW Children(name, age, grade, school_name)
```

```
    AS
    SELECT a.person.name, a.person.age, a.grade, a.school_name
    FROM ChildrenTab a;

CREATE OR REPLACE VIEW Baby(name, age, grade, school_name, birth_weight)
    AS
    SELECT a.person.name, a.person.age, a.grade, a.school_name, '-'
    FROM ChildrenTab a
    WHERE a.person.age <=2;

CREATE OR REPLACE VIEW Adult(name, age, position, salary, degree)
    AS
    SELECT a.person.name, a.person.age, a.position, a.salary, a.degree
    FROM AdultTab a;
```

2. วิวที่กำกับลงบนแบบจำลองข้อมูลเชิงวัตถุชุดที่ 2

```
CREATE OR REPLACE VIEW Person(name, age)
    AS
    SELECT a.name, a.age
    FROM PersonTab a
    UNION
    SELECT b.person.name, b.person.age
    FROM BabyTab b
    UNION
    SELECT c.person.name, c.person.age
    FROM AdultTab c;

CREATE OR REPLACE VIEW Children(name, age, grade, school_name)
    AS
    SELECT a.person.name, a.person.age, a.grade, a.kindergarten_name
    FROM BabyTab a;

CREATE OR REPLACE VIEW Baby(name, age, grade, school_name, birth_weight)
    AS
SELECT a.person.name, a.person.age, a.grade, a.kindergarten_name, a.birth_weight
    FROM BabyTab a;

CREATE OR REPLACE VIEW Adult(name, age, position, salary, degree)
    AS
    SELECT a.person.name, a.person.age, a.position, a.salary, a.degree
    FROM AdultTab a;
```

ภาคผนวก ค
ผลงานที่ตีพิมพ์จากการวิจัย

An Integration of Data Sources with UML Class Models Based on Ontological Analysis

Manachaya Jamadhvaja
Department of Computer Engineering
Chulalongkorn University
Phyathai Road, Pathumwan, Bangkok 10330 Thailand
+66 2 218 6991
manachaya.j@student.chula.ac.th

Twittie Senivongse
Department of Computer Engineering
Chulalongkorn University
Phyathai Road, Pathumwan, Bangkok 10330 Thailand
+66 2 218 6996
twittie.s@chula.ac.th

ABSTRACT
Data integration is an effective method to interoperate data that reside in different sources for the purpose of providing users with a single point of access to those data. Due to data heterogeneity, data correctness and consistency are significant for integration. Richer semantics of data is a major factor in resolving conflicts among heterogeneous data sources. As UML class model represents only schema-based semantics of data (e.g. classes, attributes, and class structures), alternative methods such as ontology is useful for representing additional semantics (e.g. data values, data units, and synonym and hypernym lists). This paper proposes a method for integrating two data sources with UML class models by using an analysis of their ontologies. In our framework, ontology will be applied to describe semantics of data in each source. Then the ontologies are analysed and compared to determine their similarities and differences. The result of the comparison is used to devise an integrated ontology that will enable querying on the integrated information.

Categories and Subject Descriptors
H.2.5 **[Heterogeneous Databases]**

General Terms
Algorithms.

Keywords
Data integration, heterogeneous data sources, UML class model, semantic integration, ontology integration.

1. INTRODUCTION
Data integration has been an information systems issue concerning the techniques to combine heterogeneous data from different sources, and to allow users to transparently access all data from multiple sources via a single view [14]. The difficulty with data integration is data heterogeneity (i.e. schema heterogeneity, semantic heterogeneity) which is comparable to the problem of federated database [22]. Several research work proposes frameworks comprising three components—global schema, source schema, and mapping between global and source schemas—for resolving the problem [4]. In such frameworks, both global and source schemas can be represented in any forms of knowledge representation which is able to reveal semantics of global and source data. And the mapping specifies the relationship between two types of the schemas.

The data integration process depends on the form of knowledge representation; therefore, good representation will sort out the heterogeneity and facilitate conflict solving for the source schemas. Some examples of schema conflicts are naming conflicts (the same data represented by different names), and structure conflicts (e.g. same data represented by different data types), while some examples of semantic or data level conflicts are data value conflict (e.g. the different ranges of 'age' values), and data unit conflict (e.g. the different units of 'age') [20]. Referring to UML class model, it reveals adequate semantics for resolving only the schema conflicts. Alternative knowledge representation that has the ability to reveal additional semantics is required. Otherwise, the data integrator has to manually solve the data level conflicts.

Ontology, as an explicit specification of a conceptualisation, has the ability to represent any knowledge in the form of a set of concepts (or terms) and describable relationships between them [12]. It is an adaptive knowledge representation for defining shared vocabularies to reveal both schema-based semantics and additional semantics. In our research, a data integration process consists of a form of knowledge representation that has the ability to represent rich semantics for sorting out data heterogeneity, an integration algorithm, a data source schema which is associated with a data source, and a technique to transform the data source schema into the knowledge representation.

This paper demonstrates an application of the integration process to two data sources with UML class models. First, we select ontology as a knowledge representation. Second, the ontology integration algorithm is adapted from the methodology for integrating object-oriented data models in [24]. This algorithm compares and integrates two object-oriented data models by using some heuristics from an experiment over ten different pairs of data models. The algorithm gives satisfactory results with regards to the results of the integration conducted manually by database integrators, and also the resulting integrated data models show

IHIS'05, November 4, 2005, Bremen, Germany.

characteristics of good object-oriented design according to some object-oriented measurement. We therefore rely on this algorithm to implement our ontology integration. In our approach, data source schemas in UML class models will be transformed into ontology-based schemas according to DSTC's Ontology Definition MetaModel which is proposed to the Object Management Group (OMG) [8]. The ontology-based schemas will be enriched with additional data semantics.

While several research work deals with the use of ontologies for integrating heterogeneous data (e.g. SIMS [1], OBSERVER [17], COIN [11], and MECOTA [25]), a number of researches specifically propose ontology integration techniques, for example OBSERVER [17], FCA-MERGE [23], PROMPT [19] and Chimaera [16]. However, these techniques tend to integrate unstructured or semi-structured data such as Web pages. By nature of such data, it is unavoidable that the proposed algorithms are complex (e.g. natural language processing is employed). On the contrary, our integration method, which is adapted from [24], focuses on an integration of two UML class models, so it is assumed that their ontologies are more structured and the algorithm proposed here should be less complex.

In Section 2, we briefly explain the use of ontology for data inference and integration. Section 3 reviews related work and discusses a comparison to our research. Our data integration framework in Section 4 is described with an example of the integration of the UML class models of two simple population databases. We demonstrate the use of the integration framework in Section 5. Finally, Section 6 concludes the paper.

2. ONTOLOGICAL INFERENCE AND INTEGRATION

Ontology provides means for sharing knowledge among information systems by defining a common vocabulary in which shared knowledge is represented [12]. As a result, ontology is a shared and common understanding that reaches across people and systems [10]. By defining UML class models as ontologies, shared vocabularies facilitate the comparison and analysis between them. Ontology will act as a mediator to handle the heterogeneity and distribution across all data sources in the integration process, and provide and interface between users and data sources in the query process.

Currently, ontology has been applied to many web applications. A number of organisations set standards to the use of ontology. For example, W3C has recommended Web Ontology Language (OWL) for publishing and sharing ontologies [2]. OWL DL (Description Logic) is an OWL sublanguage that was designed to provide a language subset that has desirable computational properties for reasoning systems. The reasoner places a number of constraints on supporting the inference capabilities. Inference engines for ontologies can reason about an ontology's schema definition [10]. Such reasoners help build ontologies and use them for advanced information access and navigation. For example, if 'baby' is subclass of 'children' in an ontology, we can automatically derive the fact that 'baby' is 'children'. Besides, if we put a rule 'children whose age are less than or equal to 2 is baby', a new fact will be inferred for some 'children' who meets this condition, such that those 'children' are also 'baby'.

OWL is developed as a vocabulary extension to RDF (Resource Description Framework) [15], so two of the representations of OWL are RDF graph and RDF triple [3], which we will use in this paper. Also Jena [7] and RDQL [21] are used as inference engine and query language for RDF triple.

In summary, the representation of the schemas and other semantic knowledge in the form of ontology give us advantages by the power of ontology integration and inference.

3. RELATED WORK

3.1 Ontology-Based Data Integration

Several researches propose approaches to apply ontologies to integrate heterogeneous data. The approaches can be classified into three groups [26] (Figure 1):

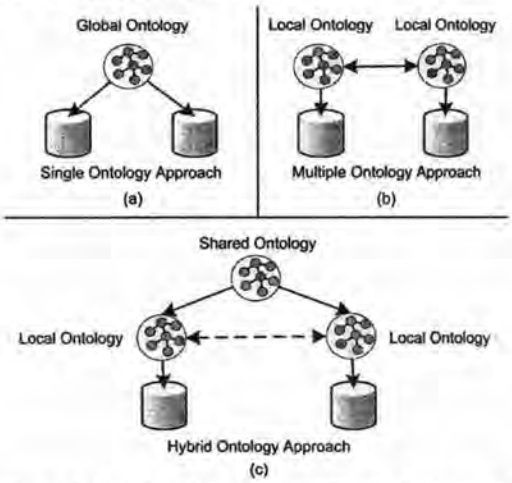


Figure 1. Ontology-based approaches for information integration [26].

1. Single Ontology Approach: One global ontology is created to describe data in all sources (Figure 1(a)), and query on the integrated data is via the global ontology. This approach looks straightforward but needs an expert who knows the meaning of all data sources to define the global ontology (e.g. SIMS [1]).
2. Multiple Ontology Approach: Data from each source is conveniently described by its own local ontology but a mapping between the local ontologies has to be established (Figure 1(b)). Query on the integrated data is via a local ontology and the mapping is used to compose local queries on other local ontologies. This approach looks more adaptive but the user has to select a suitable ontology that provides the required data as the mediator (e.g. OBSERVER [17]).
3. Hybrid Ontology Approach: This combination of the first two approaches describes data in each source with a local ontology and a shared ontology is built for sharing vocabularies among local ontologies (Figure 1(c)). This approach takes the advantages of the two approaches: ease of defining ontologies locally and of querying via the shared ontology. But mapping shared ontology onto local ontologies each time there is a query is an overhead (e.g. COIN [11] and MECOTA [25]).

Our research will combine the hybrid approach with the single ontology approach. Following the hybrid approach, data in each source is described by a local ontology and an integrated ontology is constructed for merging two local ontologies. The integrated ontology will then be associated directly to the data sources and query on the integrated data will be through the integrated ontology only, hence the single approach. Section 4 will go into more details on our approach.

3.2 Transformation of UML Class Model into Ontology

Ontology is a standard knowledge representation language which allows users to define common vocabularies as shared terms for semantically communicating and exchanging knowledge. By using ontological concepts, knowledge will be described as classes, properties and restrictions on the properties [13, 18]. As ontology language is flexible and easy to use, it has become popular in many fields of research including knowledge engineering and semantic web. Not only representing general knowledge which is unstructured or semi-structured data, ontology also represents structured data such as UML class model. Many research work deals with the transformation from UML class model into ontology. In [5], mappings between UML, DL, ER, and Ontology are proposed. Java2OWL [6] converts Java classes and instances to OWL using cardinality constraints and XML Schema datatypes.

Since the OMG has announced an RFP (Request For Proposal) for an Ontology Definition MetaModel, its requirements are to represent the semantics of ontologies, depict the ontology by using UML and so on [9]. Four organizations, including DSTC, submitted their proposal. In our research, we adopt the DSTC's proposal [8] to define the transformation of UML class model into ontology.

4. OUR FRAMEWORK

As discussed in Section 3.1, our data integration approach is a combination of the single ontology and the hybrid approaches, and we simplify the ontology integration methodology by focusing on how to integrate ontologies which describe structured information such as two UML class models. Our framework is divided into four main steps as follows (Figure 2):

1. Define: The data owner of each source, using an upper ontology for UML class models (see Section 4.1), defines their data in terms of their local ontology.
2. Add semantics: The data integrator takes part in adding some semantics such as synonym and hypernym lists to local ontologies.
3. Compare: Compare is an automatic step to compare and analyse the degree of relationships between two local ontologies. This step results in five types of relationships, i.e. equivalence, superclass, subclass, sibling and disjoint.
4. Construct & Inference: By using the output from previous step, an integrated ontology is constructed. Rules can be added to generate new facts to the resulting integrated ontology, and after that, an inference engine is employed to derive more facts about the data models. This helps refine the integrated ontology. The integrated ontology is then ready to act as the single point of access to the data sources.

Our approach gains from the expressiveness of ontology which can describe both the semantics already present in the UML class models (i.e. classes, attributes, types of attributes, inheritance, aggregation, sibling) and other additional semantics (i.e. other class and attribute semantics, synonym, and hypernym). For simplicity and due to limited space, the proposed approach will be explained through an example of the integration of two simple population data models in Figures 3(a) and 3(b). Additional semantics are also added to the example, i.e. the semantic lists for classes are attached to both models together with a synonym list in Figure 3(c). The integration process will follow the four steps mentioned above.

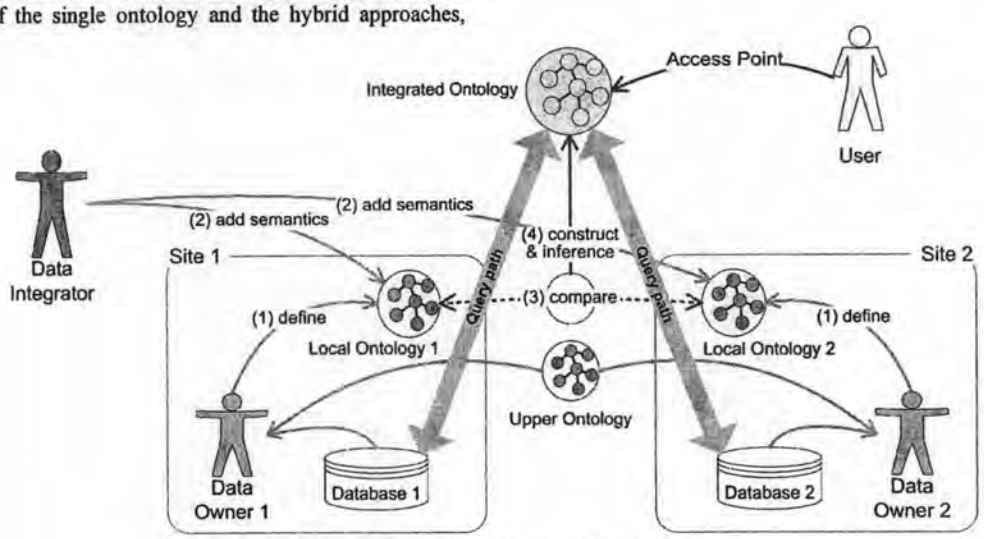


Figure 2. Our data integration framework.

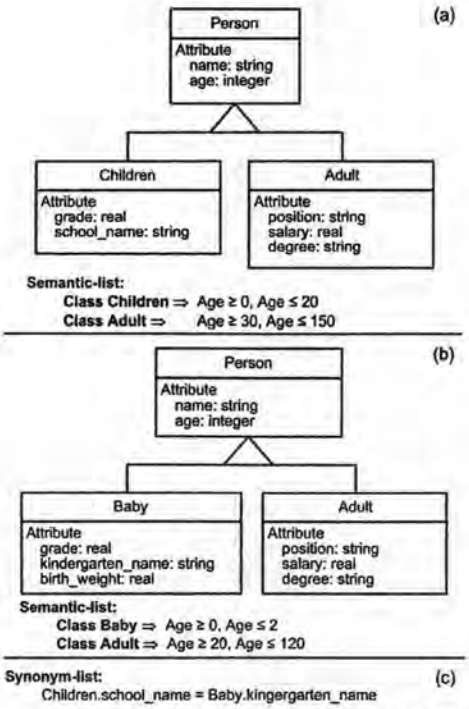


Figure 3. Example of UML class models and additional semantics.

4.1 Defining Local Ontologies

Upper ontology (the upper part of Figure 4) is a meta-ontology from which the local ontologies of the two data models will be derived (OWL ontology here is represented in RDF graph [15]). It explains what semantics should be described for the data models. Both data owners and integrator need to define or add any kinds of semantic data that conform to the given upper ontology.

From the upper ontology, the semantics of the UML class model is classified into two types: schema-based semantics and additional semantics. Schema-based semantics defines the semantics that is already present in the UML class model. In Figure 3, schema-based semantics refers to the UML class diagrams (the upper part of Figures 3(a) and 3(b)). We use two ontology classes, i.e. "Class" and "ClassAttribute", to represent object-oriented class and attribute respectively. The ontological object property "hasClassAttribute" is used to describe the property of a class having an attribute, and the ontological datatype property "hasClassAttributeType" is used to describe the data type of an attribute. Additional semantics defines other semantics that a data integrator may add to local ontologies for the purpose of integration; for example, in Figure 3, additional semantics is represented as Semantic-list (in the lower part of Figures 3(a) and 3(b)) and Synonym list (in Figure 3(c)). According to the upper ontology, the data owners/ integrator can add the additional semantics to both classes and attributes of UML class models in two ways:

1. By using "hasClassSemantic" or "hasClass AttributeSemantic" property for class semantics and attribute semantics, the semantics will be defined as "SemanticName" and "Semantic Value". For example, in Figure 5, we define semantics of Children as "ChildrenAgeSemantic" with value "≥ 0" and "≤ 20".
2. By using "owl:equivalentClass" and "owl:unionOf" property for both class and attribute, class semantics and attribute semantics will be defined for a synonym or hypernym relationship between a term in a local ontology and another term in the other local ontology. For example, in Figure 5, we define a synonym of "ChildrenSchoolName" as "owl:equivalentClass BabyKindergartenName" to show that they have the same meaning.

The semantics of the two models in Figure 3 will be represented as local ontology 1 (the middle part) and local ontology 2 (the lower part) of Figure 4 respectively. They are derived from the upper ontology in the upper part.

4.2 Comparing Local Ontologies

Ontology comparison is based on class comparison in order to find the relationships between two UML classes, which are now represented as subclasses of "Class" in the upper ontology. Class comparison (Section 4.2.3) is a bottom-up process that results from semantic comparison (Section 4.2.1) and ClassAttribute comparison (Section 4.2.2). First, each pair of the classes from the two local ontologies is selected. Second, class semantic comparison and ClassAttribute comparison are performed respectively. Finally, the relationship that this pair of classes has with each other is returned by class comparison.

4.2.1 Semantic Comparison

Semantic comparison is based on some heuristics that is derived from an experiment on the integration of ten pairs of OO data models and from a data integration expert [24]. It gives a numeric value that tells the degree of relationships between two terms that are compared. Let T1 and T2 be the terms in local ontology 1 and local ontology 2 respectively whose semantics are to be compared. α is the number of SemanticName of T1, β is the number of SemanticName of T2, and $\alpha \leq \beta$.

Each pair of "SemanticName and SemanticValue" will be compared and the heuristic value will be returned according to Table 1. The heuristic scores that reflect the result from the comparison of each semantic pair will be gathered in a set $R = [r_1, r_2, \ldots, r_\alpha]$, and $Ms = \sum r_i$ where $r_i \in R$ and $i = 1, \ldots, \alpha$ is computed. Table 2 shows the kind of relationships that T1 has with T2.

Table 1. Results of comparison of each semantics of T1 with that of T2.

Case	Semantic name	Semantic values	r
1	Same	Same	1
2	Same	Subset	0.8
3	Same	Overlap	0.8
4	Same	Disjoint	0.5
5	Different	Any	0

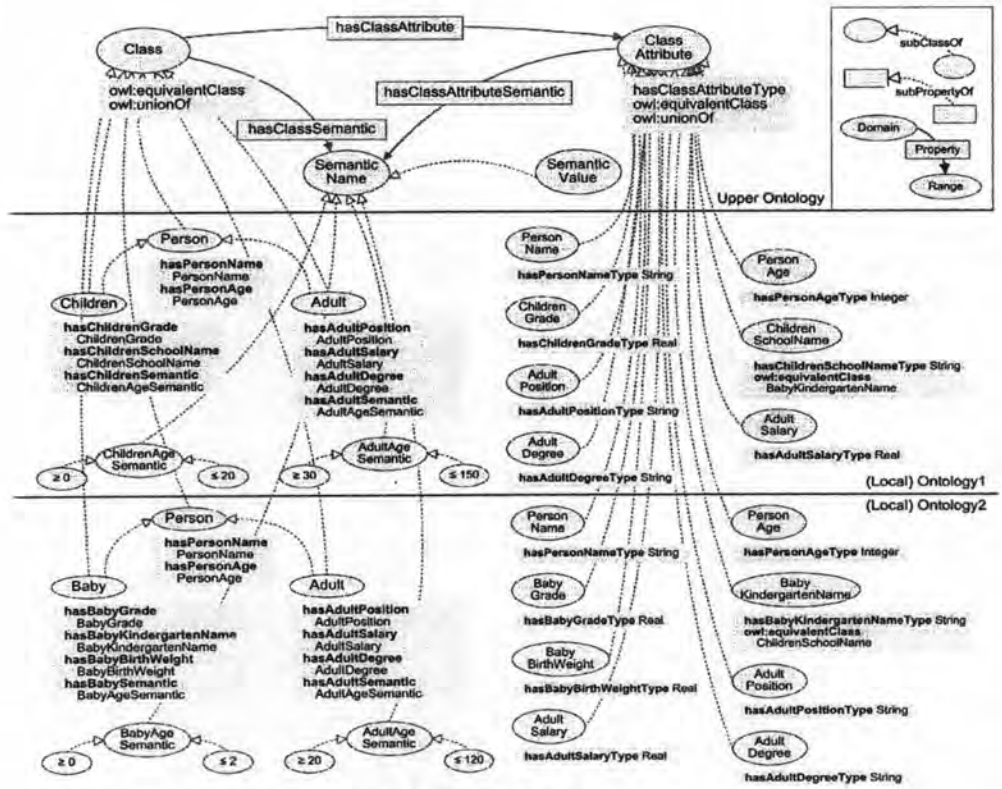


Figure 4. Upper and local ontologies.

Table 2. Relationships of terms with semantic consideration.

Case	Ms	Result	Relationship	s
1	α ($\alpha = \beta$)	Equivalence	Equivalence	1
2	α ($\alpha < \beta$)	Inclusion	Superclass*	1
3	$\geq \frac{3}{4}\alpha$ ($\alpha = \beta$)	Equivalence	Equivalence	1
4	$\geq \frac{3}{4}\alpha$ ($\alpha < \beta$)	Inclusion	Superclass*	1
5	$\geq \frac{1}{2}\alpha$	Tight-intersection	Sibling	0.1
6	$< \frac{1}{2}\alpha$	Loose-intersection	Disjoint	0
7	0	Disjoint	Disjoint	0

*T1 is Superclass of T2 = T2 is Subclass of T1

4.2.2 ClassAttribute Comparison

ClassAttribute comparison is based on the comparison of ClassAttribute and the comparison of the set of ClassAttributes.

1. ClassAttribute comparison is a process to compare each pair of ClassAttribute in a class. Each pair of ClassAttribute will be compared and the heuristic value will be returned according to Table 3.
2. Set of ClassAttributes comparison is a process to compare a set of ClassAttributes of a class to another one. Set of ClassAttributes comparison is also based on heuristics scoring from [24]. The heuristics value tells the degree of relationships between two classes that are compared. Let C1 and C2 be the classes in local ontology 1 and local ontology 2 respectively whose ClassAttributes are to be compared. θ_{C1} is the number of ClassAttributes of C1, θ_{C2} is the number of ClassAttributes of C2. Let $\alpha = \theta_{C1}$, $\beta = \theta_{C2}$, and $\alpha \leq \beta$. If a ClassAttribute in C1 has "owl:unionOf" property with other group of ClassAttributes in C2, then $\alpha = \alpha - 1$ + the number of ClassAttributes in the group. And if a ClassAttribute in C2 has "owl:unionOf" property with other group of ClassAttributes in C1, $\beta = \beta - 1$ + the number of ClassAttributes in the group.

Table 3. Results of comparison of each ClassAttribute A with ClassAttribute B.

Case	Comparison condition	s	Relationship
1	Name of A is sameAs B and their additional semantics are defined	(see Table 2)	(see Table 2)
2	Name of A is differentFrom B but both are equivalentClass, and their additional semantics are defined	(see Table 2)	(see Table 2)
3	Name of A is sameAs B but no additional semantics is defined	1	Equivalence
4	Name of A is differentFrom B but both are equivalentClass, and no additional semantics is defined	1	Synonym
5	A is unionOf B or B is unionOf A	1	Hypernym
6	A is AllDifferent from B	0	Disjoint

Each pair of ClassAttributes in C1 and C2 will be compared using the method in the ClassAttribute comparison process. The result from the comparison of each ClassAttribute pair will be gathered in a set $S = [s_1, s_2, \ldots, s_\alpha]$, and $Ms = \sum s_i$ where $s_i \in S$ and $i = 1, \ldots, \alpha$ is computed. Table 4 shows the kind of relationships that C1 has with C2.

Table 4. Relationships of classes C1 and C2, with consideration on the set of ClassAttribute.

Case	Ms	Result	Relationship
1	$\alpha(\alpha = \beta)$	Equivalence	Equivalence
2	$\alpha(\alpha < \beta)$	Inclusion	Superclass*
3	$\geq \frac{3}{4}\alpha(\alpha = \beta)$	Equivalence	Equivalence
4	$\geq \frac{3}{4}\alpha(\alpha < \beta)$	Inclusion	Superclass*
5	$\geq \frac{1}{2}\alpha$	Tight-intersection	Sibling
6	$< \frac{1}{2}\alpha$	Loose-intersection	Disjoint
7	0	Disjoint	Disjoint

*C1 is Superclass of C2 = C2 is Subclass of C1

4.2.3 Class Comparison

The relationship that a class C1 has with a class C2 is based on the relationship between their class semantics and the relationship between all of their ClassAttributes as a whole. From the method in section 4.2.1 and 4.2.2, we can obtain Relationship of ClassSemantic which indicates the kind of relationships between all of their class semantics and Relationship of the set of ClassAttributes which indicates the kind of relationships between a particular pair of their attributes. In this process, we can obtain the relationship that C1 has with C2 from Table 5.

Table 5. Summary of class relationships.

Case	Result of ClassSemantic comparison (Section 4.2.1)	Result of set of ClassAttributes comparison (Section 4.2.2)	Relationship
1	- Equivalence - Superclass - Subclass - Somenone*	Equivalence	Equivalence
2	- Equivalence - Superclass - Subclass - Somenone*	Superclass	Superclass
3	- Equivalence - Superclass - Subclass - Somenone*	Subclass	Subclass
4	- Equivalence - Superclass - Subclass - Somenone*	Sibling	Sibling
5	- Disjoint	Sibling	Sibling

*Either or both of the two classes do not define class semantics

Figure 5 shows the comparison of classes 'Children' and 'Baby' from Figure 4. The comparison follows the steps in section 4.2.1, 4.2.2 and 4.2.3. The result is that 'Children' has ClassSemantics and the set of ClassAttribute that are superclass of those of 'Baby', and therefore 'Children' is superclass of 'Baby'. Other pairs of classes in the two ontologies will be compared in a similar manner.

4.3 Constructing Integrated Ontology

Ontology comparison results in the relationship between each pair of the classes that are compared being identified (i.e. equivalence, superclass, subclass, or sibling). From the result, we can construct a new integrated ontology based on the resulting relationships. Table 6 shows the result of the comparison between pairs of the classes in our example.

Table 6. Result of the comparison between local ontology 1 and local ontology 2.

	Ontology1.Person	Ontology1.Children	Ontology1.Adult
Ontology2.Person	Equivalence	Superclass	Superclass
Ontology2.Baby	Subclass	Subclass	Sibling
Ontology2.Adult	Subclass	Sibling	Equivalence

The resulting relationship between each pair of the compared classes determines the order of class integration. Before integrating a class in one ontology with any of the classes in the other ontology, the relationship that the class has with each class in the other ontology must be determined. We suggest that the equivalence relationship has the highest priority, and so the two classes with this kind of relationship should be integrated first. This can be followed by the integration of the classes with the lower-priority relationships, i.e. superclass, subclass, and sibling respectively. Figure 6 presents the integrated ontology of our population example which depicts 'Person' as the root class with 'Children' and 'Adult' as its subclasses whereas 'Baby' also inherits from 'Children'.

From integrated ontology, adding some inference rules from data integrator's knowledge will increase the ability and correctness of the integrated ontology when processing the query. For example, with the semantics of 'Children' and 'Baby', the integrator can add the fact that 'Children' whose age is less than or equal to 2 is also 'Baby'.

In the case that the data integrator is not satisfied with the integrated ontology, it is recommended to return to the 'Add semantic' step to add more meanings to the local ontologies so as to make them capture richer semantics. The 'Compare' and 'Construct' steps will then be repeated.

4.4 Applying Integrated Ontology to Data Sources

Since the integrated ontology is created and inferred as shown in Figure 6, all resulting classes in the integrated ontology will be mapped to the local ontologies. By using this mapping, the integrated ontology will be annotated to the data sources, e.g. as a view over the DBMS which manages the data sources. A query through the integrated ontology will be passed and translated as queries on the data sources. Example of query will be demonstrated in the next section.

5. QUERY EXAMPLE

We present two query examples for demonstrating paths and translations of query which is submitted to our framework. Query is in terms of RDQL (RDL Data Query Language) [21], which is suitable for extracting information from RDF triples. After the query is submitted to the integrated ontology, the integrated ontology passes RDQL to the DBMS of all data sources. Then a view, which has been created previously for supporting annotation of the integrated ontology, takes part to translate RDQL to other query language based on the DBMS language.

Semantic comparison	
Ontology2.Baby ⊂ Ontology1.Children ⇒ Ontology1.Children.Age has Ontology2.Baby.Age as a subset ∴ R = [0.8], Ms = 0.8, α = 1, β = 1 Result = Equivalence **Relationship = Equivalence**	- Case 2 in Table 1 - Case 3 in Table 2
ClassAttribute comparison	
Ontology1.Children.PersonName is sameAs Ontology2.Baby.PersonName, but they have no additional semantics. ⇒ return Equivalent s = 1	- Case 3 in Table 3
Ontology1.Children.PersonAge is sameAs Ontology2.Baby.PersonAge, and semantics of Ontology2.Baby.PersonAge ⊂ semantics of Ontology1.Children.PersonAge ⇒ return Superclass s = 1	- Case 1 in Table 3 - Case 3 in Table 2 (also from semantic comparison)
Ontology1.Children.ChildrenGrade is sameAs Ontology2.Baby.BabyGrade, but they have no additional semantics. ⇒ return Equivalent s = 1	- Case 3 in Table 3
Ontology1.Children.ChildrenSchoolName is differentFrom Ontology2.Baby.BabyKindergartenName, but Ontology1.Children.ChildrenSchoolName is equivalentClass to Ontology2.Baby.BabyKindergartenName, and they have no additional semantics. ⇒ return Synonym s = 1	- Case 4 in Table 3
∴ $Ms_{SetOfClassAttribute}$ = 1+1+1+1 = 4, α = 4, β = 5 Result = Inclusion **Relationship = Superclass**	- Case 2 in Table 4
Class comparison	
∴ From Table 5, Relationship of ClassSemantic + Relationship of set of ClassAttribute **= Equivalence + Superclass = Superclass**	- Case 2 in Table 5

Figure 5. Comparison between class 'Children' in local ontology 1 and class 'Baby' in local ontology 2.

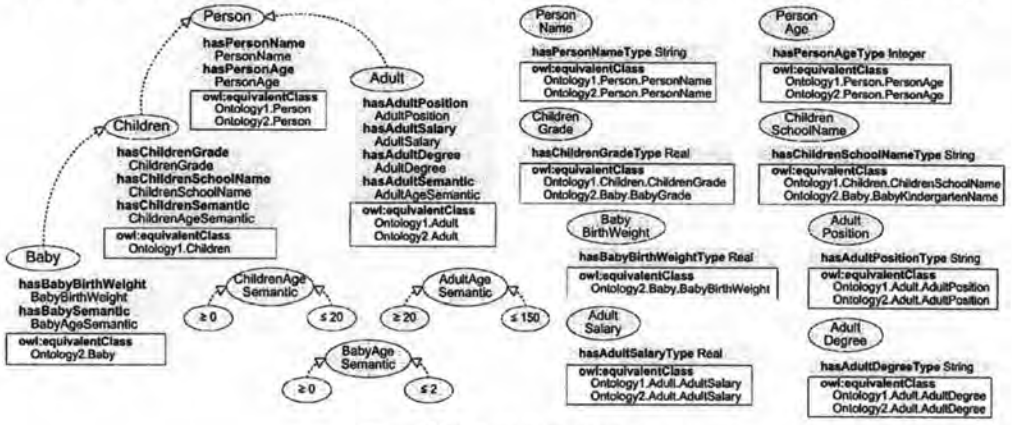


Figure 6. Integrated ontology.

Example 1.

```
SELECT ?AdultPosition, ?AdultSalary
WHERE  (?AdultPosition info:position
        ?AdultSalary)
AND ?AdultSalary <= 20000
USING info FOR <http://example.org/peopleInfo#>
```

From integrated ontology in Figure 6, the query in Example 1 wants to find the positions with a salary less than or equal to 20000. This query will be passed to the DBMS of both data sources. At data sources 1 and 2 which are described by local ontologies 1 and 2 respectively, this query will be translated to such an SQL query as shown below. The DBMS will manage to send the new SQL query and return the result back to the integrated ontology.

```
SELECT Adult.position, Adult.salary
FROM Adult
WHERE Adult.salary <= 20000
```

Example 2.

```
SELECT ?ChildrenName, ?ChildrenAge
WHERE  (?Children info:age ?ChildrenAge)
AND ?ChildrenAge > 1
USING info FOR <http://example.org/peopleInfo#>
```

From integrated ontology in Figure 6, the query in Example 2 wants to find the names and ages of the children who are older than one-year old. This query will be passed to the DBMS of both data sources. At source 1, this query will be translated to such an SQL query as shown below.

```
SELECT Children.name, Children.age
FROM Children
WHERE Children.age > 1
```

At source 2, this query will be translated to an SQL query as shown below.

```
SELECT Baby.name, Baby.age
FROM Baby
WHERE Children.age > 1
```

6. CONCLUSION

The use of ontology for integrating structured data such as UML class models gains advantage from the expressiveness of ontology, and achieves good result for integration. A query through the integrated ontology will be passed and translated as

queries onto the data sources. Users can also benefit from describing semantics by ontology because query can then be semantics-based by the inference power of ontology. For example, a query for the data of all children will result in the data of children from data source 1 and of baby from data source 2 to be returned.

This paper reports the progress of the prototype of the ontology integration framework. We are also in the process of refining the ontology integration approach. We expect more kinds of semantics to be included in the local ontologies, such as integrity constraints on attributes. Further evaluation of the approach is also a future work. As mentioned before, the integration algorithm has been evaluated by data integration experts and object-oriented measurement scheme. We expect to try it with data models that are used in real applications and compare the performance against other research works.

7. ACKNOWLEDGMENTS

This work is part of "A Design and Development of Enterprise System with Distributed System Technology" research supported by the Thai Government-Industry Linkage Research Grant (Fiscal Year 2004).

8. REFERENCES

[1] Arens, Y., Hsu, C.-N., et al. Query Processing in the SIMS Information Mediator. in *Readings in agents*, Morgan Kaufmann Publishers Inc., 1998, 82 - 90.

[2] Bechhofer, S., Harmelen, F.v., et al. OWL Web Ontology Language Reference. Dean, M., et al. eds. *W3C Recommendation*, W3C, 2004.

[3] Beckett, D. RDF/XML Syntax Specification (Revised). McBride, B. ed. *W3C Recommendation*, W3C, 2004.

[4] Cali, A., Reasoning in Data Integration Systems: Why LAV and GAV Are Siblings. in *The 14th International Symposium on Methodologies for Intelligent Systems (ISMIS 2003)*, (2003), 562-571.

[5] Colomb, R.M., Gerber, A., et al., Issues in Mapping Metamodels in the Ontology Development Metamodel Using QVT. in *The 1st International Workshop on the Model-Driven Semantic Web (MSDW 2004)*, (Monterey, California, USA, 2004).

[6] Dean, M. Java2OWL, DAML.org, 2003.

[7] Dickinson, I. The Jena 2 Ontology API, Jena Semantic Web Framework, 2005.

[8] DSTC. Ontology Definition MetaModel - DSTC Initial Submission, Object Management Group, 2003, 70.

[9] DSTC, Gentleware, et al. Ontology Definition MetaModel *Preliminary Revised Submission to OMG RFP ad/2003-03-40 Volume 1*, 2004, 169.

[10] Fensel, D., van Harmelen, F., et al. OIL: an ontology infrastructure for the Semantic Web. *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, *16* (2). 38-45.

[11] Goh, C.H., Bressan, S.p., et al. Context Interchange: New Features and Formalisms for the Intelligent Integration of Information. *ACM Transactions on Information Systems (TOIS)*, *17* (3). 270-293.

[12] Gruber, T.R. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition (1042-8143)*, *5* (2). 199 - 220.

[13] Horridge, M., Knublauch, H., et al. *A Practical Guide To Building OWL Ontologies Using The Protege-OWL Plugin and CO-ODE Tools*. The University Of Manchester, 2004.

[14] Lenzerini, M., Data Integration: A Theoretical Perspective. in *The 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (Madison, Wisconsin, 2002), ACM Press New York, NY, USA, 233 - 246.

[15] McBride, B. An Introduction to RDF and the Jena RDF API, http://jena.sourceforge.net/tutorial/RDF_API/index.html, 2005.

[16] McGuinness, D.L., Fikes, R., et al., An Environment for Merging and Testing Large Ontologies. in *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, (Breckenridge, Colorado, 2000).

[17] Mena, E., Illarramendi, A., et al. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. *Distributed and Parallel Databases -- An International Journal*, *8* (2). 223-271.

[18] Noy, N.F. and McGuinness, D.L. Ontology Development 101: A Guide to Creating Your First Ontology *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, 2001.

[19] Noy, N.F. and Musen, M.A., PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. in *The Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, (2000), AAAI Press / The MIT Press, 450 - 455.

[20] Ram, S. and Park, J. Semantic conflict resolution ontology (SCROL): an ontology for detecting and resolving data and schema-level semantic conflicts. *Knowledge and Data Engineering, IEEE Transactions on*, *16* (2). 189-202.

[21] Seaborne, A. RDQL - A Query Language for RDF *W3C Member Submission 9*, W3C, 2004.

[22] Sheth, A.P. and Larson, J.A. Fedetated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, *22* (3). 183-236.

[23] Stumme, G. and Maedche, A., FCA-MERGE: Bottom-Up Merging of Ontologies. in *The Seventeenth International Joint Conference on Artificial Intelligence*, (Seattle, Washington, USA, 2001).

[24] Swasrukkiet, S. A Methodology for Integration of Object-Oriented Data Models Using Heuristics and Analysis of Relationships between Class. *Department of Computer Engineering*, Chulalongkorn University, Bangkok, 1999, 162.

[25] Wache, H., Scholz, T., et al., An integration method for the specification of rule-oriented mediators. in *Database Applications in Non-Traditional Environments, 1999. (DANTE '99) Proceedings. 1999 International Symposium on*, (1999), 109-112.

[26] Wache, H., Vogele, T., et al., Ontology-Based Integration of Information — A Survey of Existing Approaches. in *The Seventeenth International Joint Conference on Artificial Intelligence*, (Seattle, Washington, USA, 2001).

An Approach to Constructing an Integrated Ontology for Integrating Object-Oriented Data Models

Manachaya Jamadhvaja* , Twittie Senivongse**

ABSTRACT

Semantically determining similarities and differences between data from multiple data sources plays an important role in the process of data integration. The similarities and differences identify the degree of relationships and are used for constructing the integrated data model. In structured data such as object-oriented database, their data models reveal only structure-based semantics without other additional semantics. This paper proposes an approach to integrate two object-oriented data models based on the integration of ontologies that describe their semantics. The ontology for an object model describes semantics based on the model structure and additional semantics, i.e. class and attribute semantics, and synonym and hypernym lists. The ontologies are analysed to devise an integrated data model that will be used to facilitate query on the integrated data.

KEYWORD : Ontology Integration, Data Model Integration, Ontology, Object-Oriented Data Model.

1. INTRODUCTION

The growth of information technology affects the use of data in information systems. In organisations, using a single data source is not much enough to run the business. Hence, data model integration is required for combining heterogeneous data from different data sources and for providing users with a single point of access to all data sources [9]. The data models that are integrated tend to be in the same context but are designed by different owners (e.g. the data models represent the population databases of different departments in a district). This will result in several kinds of semantic heterogeneity, for example, name conflict (the same information represented by different names or different information represented by the same name), structure conflict (e.g. same information represented by different data types), and scaling conflict (e.g. an attribute 'salary' with different currencies in different data models) [14]. To integrate two data models, a database integrator must understand well the meaning of both data models and can identify the similarities and differences in the semantics that underlies the data models.

For structured data such as object-oriented databases, structure-based semantics are represented as shown in UML class diagram in Figure 1. UML class diagram describes the semantics of an object-oriented data model in terms of classes and attributes that form the model as well as the relationships between them, but it does not reveal any other meaning inside the data.

The use of ontologies for describing the meaning inside the data is a practical method to solve the problem of semantic heterogeneity. Ontology, an explicit specification of a conceptualization [13], represents knowledge in a domain of interest in the form of a set of concepts (or terms) and describable relationships between them. W3C has recommended Web Ontology Language (OWL) for publishing and sharing ontologies [11]. OWL is developed as a vocabulary extension to RDF (Resource Description Framework), so one of the representations of OWL is RDF graph [1]. By using ontological concepts to represent semantics in the data models, shared vocabularies are defined to illustrate both structure-based semantics and other additional semantics in the models. As a result, ontologies will handle the heterogeneity and distribution across all data sources, and be the mediator between users and data sources. Ontologies will facilitate flexible semantics-based query on the data sources. Several research work deals with the use of ontologies for integrating heterogeneous data (e.g. SIMS [15], OBSERVER [5], COIN [2] and MECOTA [7]).

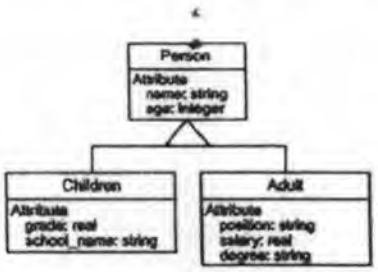


Fig. 1. Example of object-oriented data model

This paper proposes an approach to integrate two object-oriented data models whose semantics are described by ontologies. Since there is a mapping between object model and ontology, we follow such a suggestion from DSTC's Ontology Definition MetaModel [4] to develop an ontology integration approach by adapting from the methodology for integrating object-oriented data models in [12]. All classes from the two data models are analysed in order to determine the degree of relationships between them before the integration can be applied. Several kinds of relationships between ontological concepts within the models are considered, i.e. inheritance, aggregation, sibling, equivalence, synonym, and hypernym. The degree of similarities and differences between the ontologies are identified by heuristics-based scoring.

In Section 2, we review related work and discuss a comparison to our research. Our ontology integration method in Section 3 is described with an example of the integration of the object-oriented data models of two population databases. Finally, Section 4 discusses the use of this approach and concludes the paper.

2. RELATED WORK

2.1 Ontology-based information integration

Several research work suggests ways to use ontologies for heterogeneous data integration. In summary, all research can be categorised into three approaches [8]:

1. Single Ontology Approach: One global ontology is used to describe all data sources (Figure 2(a)), and query on the integrated data is via the global ontology. This approach

*Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University
**Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University

looks straightforward but needs an expert who knows the meaning of all data sources to define the global ontology (e.g. SIMS [15])

2. Multiple Ontology Approach: Each data source is conveniently described by its own local ontology but a mapping between the local ontologies has to be established (Figure 2(b)). Query on the integrated data is via a local ontology and the mapping is used to compose local queries on other local ontologies. This approach looks more adaptive but mapping local ontologies each time there is a query is an overhead (e.g. OBSERVER [5]).

3. Hybrid Ontology Approach: This combination of the first two approaches describes each data source with a local ontology and a shared ontology is built to for sharing vocabularies among local ontologies (Figure 2(c)). This approach takes the advantages of the two approaches: ease of defining ontologies locally and of querying via the shared ontology (e.g. COIN [2] and MECOTA [7]). Our research follows this hybrid approach.

2.2 Ontology integration

Most of research in ontology integration presents techniques for integrating ontologies which describe knowledge about unstructured or semi-structured data such as Web pages. By nature of such data, it is unavoidable that the proposed algorithms are complex (e.g. natural language processing is employed). For example, OBSERVER [1] presents a method that performs the translation from a local ontology into other local ontologies for cross-ontology query; no integrated ontology is actually created. On the contrary, FCA-MERGE [6], PROMPT [10], and Chimaera [3] propose ideas for constructing integrated ontology. FCA-MERGE [6] applies mathematical techniques to derive concepts of ontology, and produces suggestions for transforming into an integrated ontology. PROMPT [10] and Chimaera [3] are interactive tools that can perform some integration tasks automatically and guide the ontology integrator in performing other tasks.

Our research focuses on integrating two object-oriented data models, so it is assumed that the semantics of these models are more structured and the algorithm proposed here should be less complex.

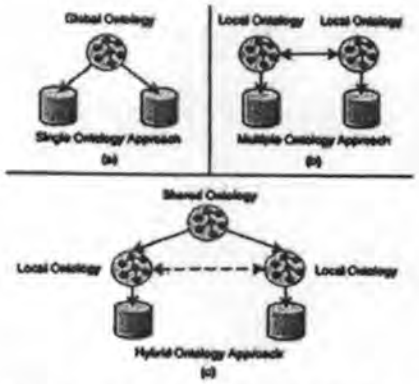


Fig. 2. Ontology-based approaches for information integration [8]

3. INTEGRATING ONTOLOGIES

For simplicity and due to limited space, the proposed approach to integrating ontologies for object-oriented data models will be explained through an example of the integration of two simple population data models in Figures 1 and 3. First, the semantics of these two models will be represented as two local ontologies based on the guideline in [4] (Section 3.1). Then they are compared in Section 3.2 and integrated in Section 3.3 using the methodology adapted from [12].

Our approach gains semantic richness of ontology and describe both the semantics that is already present in the data models (i.e. class, attribute, inheritance, aggregation, sibling) and the other additional semantics (i.e. other class and attribute semantics, equivalence, synonym, hypernym).

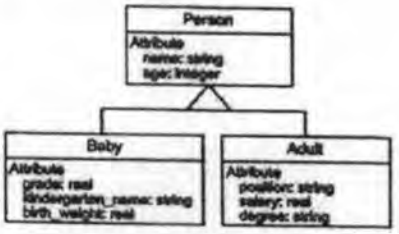


Fig. 3. Another example of object-oriented data model

3.1 Using upper ontology to derive local ontologies

Upper ontology (Figure 4) is a meta-ontology from which the local ontologies of the two object-oriented data models will be derived (OWL ontology here is represented in RDF graph). It explains what semantics should be described for the data models.

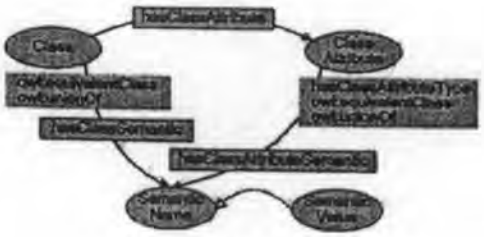


Fig. 4. Upper ontology

From the upper ontology, the semantics of the object-oriented data model are classified into two types: structure-based semantics and additional semantics. Structure-based semantics defines the semantics that is already present in the object-oriented data models. We use two ontology classes, i.e. "Class" and "ClassAttribute", to represent object-oriented class and attribute respectively. The ontology object property "hasClassAttribute" is used to describe the property of a class having an attribute, and the ontology datatype property "hasClassAttributeType" is used to describe data type of an attribute. Additional semantics defines other semantics that a data model integrator may add to the ontology for the purpose

of integration. According to the upper ontology, the additional semantics can be added to both OO classes and attributes in two ways:

1. By using "hasClassSemantic" or "hasClass Attribute Semantic" property for class semantics and attribute semantics, the semantics will be defined as "SemanticName" and "Semantic Value". For example, in Figure 5, we define semantics of Children as "ChildrenAgeSemantic" with value "≥ 0" and "≤ 20".

2. By using "owl:equivalentClass" and "owl:unionOf" property for both class and attribute, class semantics and attribute semantics will be defined as what term in a local ontology is a synonym or hypernym of what term in the other local ontology. For example, in Figure 5, we define a synonym of "ChildrenSchoolName" as "owl:equivalentClass BabyKindergartenName" to show that their meanings are the same.

The semantics of the two models in Figures 1 and 3 will be represented as local ontology 1 (upper side) and 2 (lower side) in Figure 5 respectively. They are derived from the upper ontology in the middle.

3.2 Ontology comparison

Ontology comparison is based on class comparison to find out the relationships between object-oriented classes, which are represented by the ontology class inherited from "Class" in the upper ontology. Class comparison (Section 3.2.3) is a bottom-up process, resulting from semantic comparison (Section 3.2.1) and ClassAttribute comparison (Section 3.2.2). Firstly, each pair of the classes from the two local ontologies is picked one by one. Secondly, class semantic comparison and ClassAttribute comparison are performed respectively. Finally, the relationship that this pair of classes has with each other is returned by class comparison. The process of ontology comparison is depicted in Figure 6.

3.2.1 Semantic comparison

Semantic comparison is based on some heuristics that is derived from an experiment on the integration of ten pairs of OO data models and from a data integration expert [11]. It gives a numeric value that tells the degree of relationship between two terms that are compared. Let T1 and T2 be the terms in local ontology 1 and 2 respectively whose semantics are to be compared. α is the number of SemanticName of T1, β is the number of SemanticName of T2, and $\alpha \leq \beta$.

Each pair of "SemanticName and SemanticValue" will be compared and the heuristic value will be returned according to Table 1. The result of each semantic pair will be gathered in a set $R = [r_1, r_2, \ldots, r_\alpha]$, and $Ms = \Sigma r_i$ where $r_i \in R$ and $i = 1, \ldots, \alpha$ is computed. Table 2 shows the kind of relationship that T1 has with T2.

3.2.2 ClassAttribute comparison

ClassAttribute comparison is based on the comparison of ClassAttribute and the comparison of the set of ClassAttribute.

1. ClassAttribute comparison is a process to compare each pair of ClassAttribute in a class. Each pair of Class Attribute will be compared and the heuristic value will be returned according to Figure 7.

Table 1. Result of comparison of each semantics of T1 with that of T2

Same	Same	1
Same	Subset	0.8
Same	Overlap	0.8
Same	Disjoint	0.5
Different	Any	0

Table 2. Relationships of Terms with Semantic Consideration

α ($\alpha = \beta$)	Equivalence	Equivalence	1
α ($\alpha < \beta$)	Inclusion	Superclass*	1
$\geq \frac{1}{4}\alpha$ ($\alpha = \beta$)	Equivalence	Equivalence	1
$\geq \frac{1}{4}\alpha$ ($\alpha < \beta$)	Inclusion	Superclass*	1
$\geq \frac{1}{2}\alpha$	Tight-intersection	Sibling	0.1
$< \frac{1}{2}\alpha$	Lose-intersection	Disjoint	0
0	Disjoint	Disjoint	0

*T1 is Superclass of T2 = T2 is Subclass of T1

2. Set of ClassAttribute comparison is a process to compare a set of ClassAttribute of a class to another one. Set of ClassAttribute comparison is also based on heuristics scoring from [11]. The heuristics value tells the degree of relationship between two classes that are compared. Let C1 and C2 be the classes in local ontology 1 and 2 respectively whose ClassAttributes are to be compared. θ_{C1} is the number of ClassAttribute of C1, θ_{C2} is the number of ClassAttribute of C2. Let $\alpha = \theta_{C1}, \beta = \theta_{C2}$, and $\alpha \leq \beta$. If a ClassAttribute in C1 has "owl:unionOf" property with other group of ClassAttribute in C2, then $\alpha = \alpha - 1 +$ number of ClassAttribute in the group. And if a ClassAttribute in C2 has "owl:unionOf" property with other group of ClassAttribute in C1, $\beta = \beta - 1 +$ number of ClassAttribute in the group.

Each pair of ClassAttribute in C1 and C2 will be compared using the method in the ClassAttribute comparison process. The result of each ClassAttribute pair will be gathered in a set $S = [s_1, s_2, \ldots, s_\alpha]$, and $Ms = \Sigma s_i$ where $s_i \in S$ and $i = 1, \ldots, \alpha$ is computed. Table 3 shows the kind of relationship that C1 has with C2.

Table. 3 Relationships of classes C1 and C2, with consideration on the set of ClassAttribute

α ($\alpha = \beta$)	Equivalence	Equivalence
α ($\alpha < \beta$)	Inclusion	Superclass*
$\geq \frac{1}{4}\alpha$ ($\alpha = \beta$)	Equivalence	Equivalence
$\geq \frac{1}{4}\alpha$ ($\alpha < \beta$)	Inclusion	Superclass*
$\geq \frac{1}{2}\alpha$	Tight-intersection	Sibling
$< \frac{1}{2}\alpha$	Lose-intersection	Disjoint
0	Disjoint	Disjoint

*C1 is Superclass of C2 = C2 is Subclass of C1

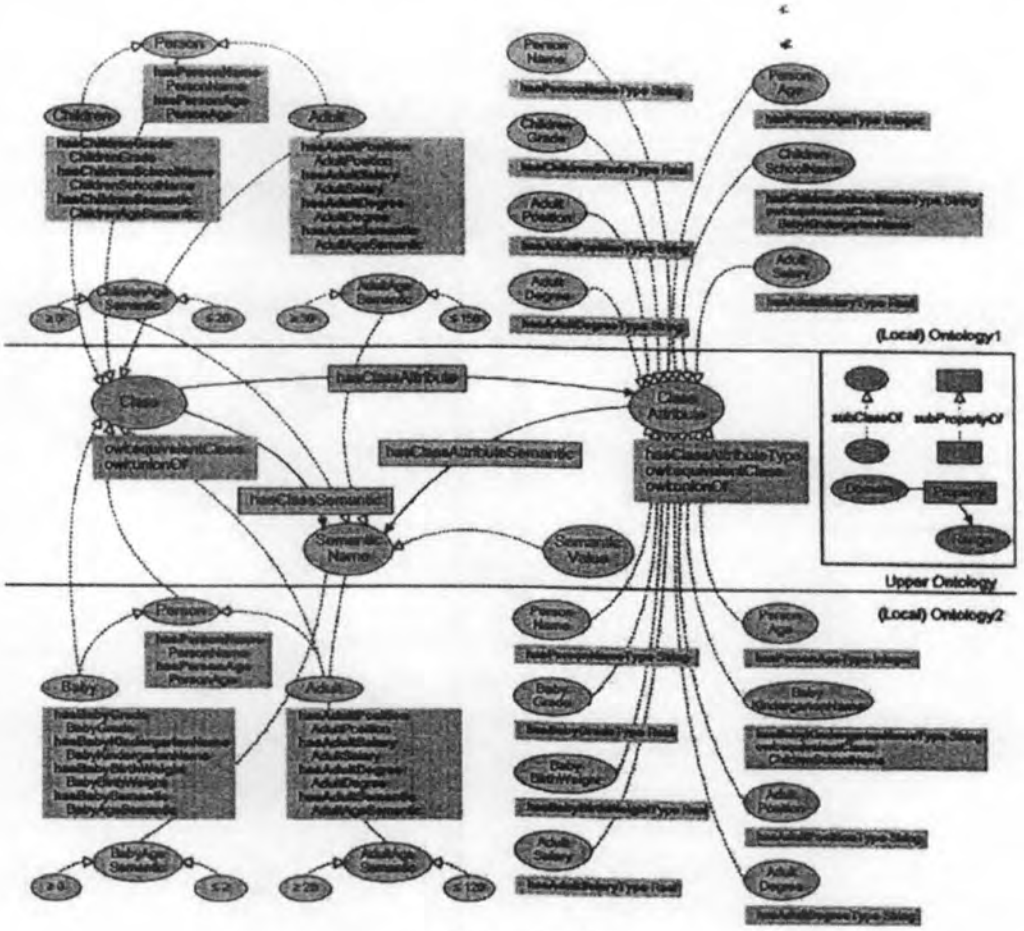


Fig. 5. Local ontologies

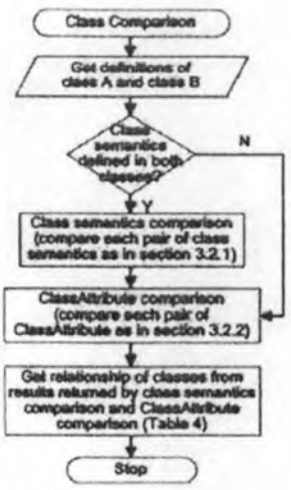


Fig. 6. Class comparison

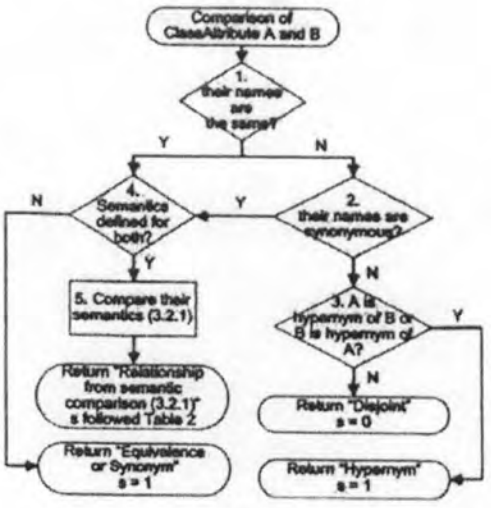


Fig. 7. ClassAttribute comparison

3.2.3 Class comparison

The relationship that a class C1 has with a class C2 is based on the relationship between their class semantics and the relationship between all of their ClassAttributes as a whole. From the method in section 3.2.1 and 3.2.2, we can obtain Relationship of ClassSemantic which indicates the kind of relationship between all of their class semantics and Relationship of the set of ClassAttribute which indicates the kind of relationship between a particular pair of their attributes. In this process, we can obtain the relationship that C1 has with C2 from Table 4.

Figure 8 shows the comparison of classes 'Children' and 'Baby' from Figure 5. The comparison follows the steps in section 3.2.1, 3.2.2 and 3.2.3. The result is that 'Children' has ClassSemantics and the set of ClassAttribute that are superclass of those of 'Baby', and therefore 'Children' is superclass of 'Baby'. Other pairs of classes in the two ontologies will be compared in a similar manner.

3.3 Ontology integration

Ontology comparison results in the relationship between each pair of the classes that are compared being identified (i.e. equivalence, superclass, subclass, or sibling). From the result, we can construct a new integrated ontology based on the resulting relationships. Table 5 shows the result of the comparison between pairs of the classes in our example. The class relationships in this table lead to the creation of the integrated ontology. Figure 9 presents the final integrated ontology of our population example which depicts 'Person' as the root class with 'Children' and 'Adult' as its subclass whereas 'Baby' also inherits from 'Children'.

Table 4: Summary of class relationships

• Equivalence • Superclass • Subclass • Someone*	Equivalence	Equivalence
• Equivalence • Superclass • Subclass • Someone*	Superclass	Superclass
• Equivalence • Superclass • Subclass • Someone*	Subclass	Subclass
• Equivalence • Superclass • Subclass • Someone*	Sibling	Sibling
• Disjoint	Sibling	Sibling

*Either or none of the two classes have ClassSemantics

Ontology1.Children.Age has Ontology2.Baby.Age as a subset
∴ R = [0.8], Ms = 0.8, α = 1, β = 1 Result = Equivalence **Relationship = Equivalence**

Ontology1.Children.PersonName has no additional semantics but has the same name as Ontology2.Baby.PersonName
⇒ return Equivalent s = 1
Ontology1.Children.PersonAge and Ontology2.Baby.PersonAge have class semantics that are subset and also have the same name
⇒ return Superclass s = 1
Ontology1.Children.ChildrenGrade has no additional semantics but has the same name as Ontology2.Baby.BabyGrade
⇒ return Equivalent s = 1
Ontology1.Children.ChildrenSchoolName and Ontology2.Baby.BabyKindergartenName have no additional semantics but they are synonymous
⇒ return Synonym s = 1
∴ $Ms_{SetOfClassAttribute}$ = 1+1+1+1 = 4, α = 4, β = 5 Result = Inclusion **Relationship = Superclass**

∴ From Table 4, Relationship of ClassSemantic + Relationship of set of ClassAttribute
= Equivalence + Superclass = Superclass

Fig. 8. Comparison between class Children in local ontology 1 and class Baby in local ontology 2

Table. 5 Result of the comparison between local ontology 1 and local ontology 2

Ontology2.Person	Equivalence	Superclass	Superclass
Ontology2.Baby	Subclass	Subclass	Sibling
Ontology2.Adult	Subclass	Sibling	Equivalence

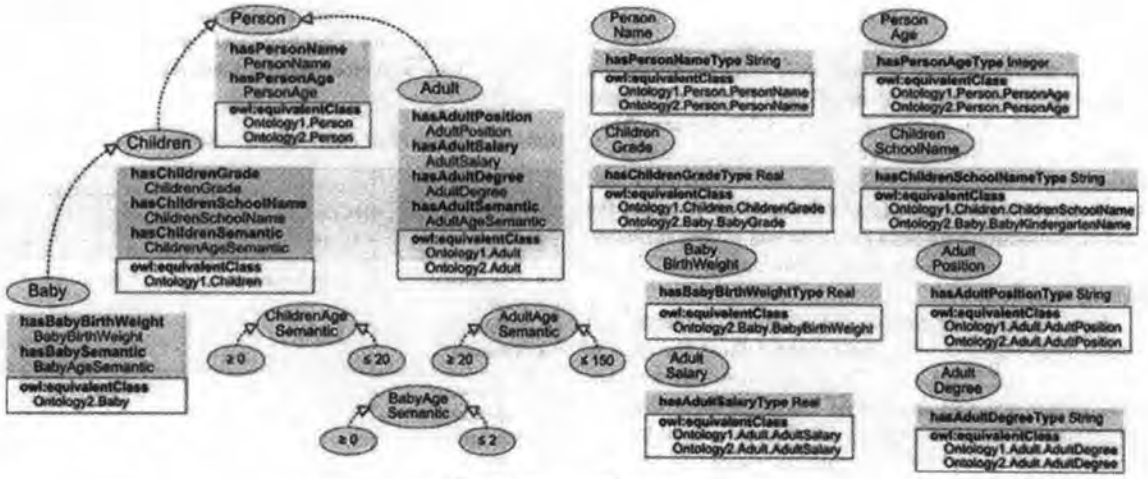


Fig. 9. Integrated ontology

4. CONCLUSION

From ontology integration, all resulting classes in the integrated ontology will be mapped to the local ontologies. A query through the integrated ontology will be translated as queries on the local ontologies of the underlying databases. Users can also benefit from describing semantics by ontology because query can then be semantics-based by the inference power of ontology. For example, a query for the data of all children will result in the data of children from data model 1 and of baby from data model 2 to be returned.

We are in the process of refining the ontology integration approach. We expect more kinds of semantics to be included in the local ontologies, such as integrity constraints on attributes. The prototype of the ontology integration and query framework is also in progress.

5. REFERENCE

[1] B. McBride, "An Introduction to RDF and the Jena RDF API," vol. 2005: http://jena.sourceforge.net/tutorial/RDF_API/index.html, 2005.

[2] C. H. Goh, et al, "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information," ACM Transactions on Information Systems (TOIS), vol. 17, pp. 270-293, 1999.

[3] D. L. McGuinness, et al, "An Environment for Merging and Testing Large Ontologies," Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, 2000.

[4] DSTC, "Ontology Definition MetaModel – DSTC Initial Submission," submit to Object Management Group, 2003.

[5] E. Mena, et al, "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies," Distributed and Parallel Databases, vol. 8, pp. 223-271, 2000.

[6] G. Stumme and A. Maedche, "FCA-MERGE: Bottom-Up Merging of Ontologies," Proceedings of 17th International Joint Conference on Artificial Intelligence, Seattle, Washington, USA, 2001.

[7] H. Wache, et al, "An integration method for the specification of rule-oriented mediators," Proceedings of 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE '99), 1999.

[8] H. Wache, et al, "Ontology-Based Integration of Information — A Survey of Existing Approaches," Proceedings of 17th International Joint Conference on Artificial Intelligence, Seattle, Washington, USA, 2001.

[9] M. Lenzerini, "Data Integration: A Theoretical Perspective," Proceedings of 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Madison, Wisconsin, 2002.

[10] N. F. Noy and M. A. Musen, "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment," Proceedings of 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, 2000.

[11] S. Bechhofer, et al, "OWL Web Ontology Language Reference," W3C Recommendation, vol. 2005, M. Dean and G Schreiber, Eds., February 10, 2004.

[12] S. Swasrukkiet, "A Methodology for Integration of Object-Oriented Data Models Using Heuristics and Analysis of Relationships between Class," Master Thesis, Department of Computer Engineering, Chulalongkorn University, Bangkok, 1999, pp. 162.

[13] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition (1042-8143), vol. 5, pp. 199 - 220, 1993.

[14] W. Sull and R.L. Kashyap, "A Self-Organizing Knowledge Representation Scheme for Extensible Heterogeneous Information Environment", IEEE Transactions on Knowledge and Data Engineering, vol. 4 no. 2, pp. 185-191, 1992.

[15] Y. Arens, et al, "Query Processing in the SIMS Information Mediator," in Readings in agents: Morgan Kaufmann Publishers Inc., pp. 82 - 90, 1998.

ประวัติผู้เขียนวิทยานิพนธ์

นางสาว มนัชยา ชมธวัช เกิดเมื่อวันที่ 26 มิถุนายน 2522 ที่จังหวัดตรัง สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2542 แล้วเข้าทำงานที่ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค) ในตำแหน่งวิศวกร 1 เป็นเวลา 3 ปี และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี 2546 เคยได้รับทุนอุดหนุนวิทยานิพนธ์สำหรับนิสิต ประจำภาคการศึกษาต้น ปีการศึกษา 2549 จากบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย