

รายการอ้างอิง

- [1] Bukhres, O. A. , and Elmagarmid, A. K. Object-Oriented Multidatabase Systems (A Solution for Advanced Applications). New Jersey: Prentice Hall International, 1996.
- [2] Sheth, A. P. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys Vol.22 No.3 (September 1990).
- [3] Reddy, M. P., Prasad, B. E., Reddy, P. G., and Gupta, A. A Methodology for Integration of Heterogeneous. Databases. IEEE Transactions on Knowledge and Data Engineering Vol.6 No.6 (December 1994).
- [4] Yu, C., Sun, W., Dao, S., and Keirse, D. Determining Relationships Among Attributes for Interoperability of Multi-Database Systems. Proceedings of the 1st Int. Workshop Interoperability in Mulidatabase System (1991): 251-257.
- [5] Johannesson, P. Linguistic Instruments and Qualitative Reasoning for Schema Integration. Third International Conference on Information and Knowledge Management. Ed. N. Adam. Gaithersburg, Maryland. IEEE Press (1994).
- [6] V. Ramesh and Sudha Ram. A Methodology for Interschema Relationship Identification in Heterogeneous Databases. Proceedings of the 28th Annual Hawaii International Conference on System Sciences, 1995.
- [7] Larab, O., Benharkat, A. N., and Wagner, R. R. Description Logics and Correspondence Refinement Process in Databases Interoperability's Service. Proceedings of the 8th Intl. Workshop on Database and Expert Systems Applications (1997): 700-705.
- [8] Schmitt, I., and Sake, G. Merging Inheritance Hierarchies for Database Integration. Proceedings of the 3rd IFCIS Intl. Conference on Cooperative Information Systems (1998): 322-331.
- [9] Garcia-Solaco, M., Saitor, F., and Castellanos, M. A Structure Based Schema Integration Methodology. Proceedings of the 11th Intl. IEEE Conference on Data Engineering (1995): 505-512.

- [10] Dillon, T. S., and Tan, P. L. Object-oriented Conceptual Modeling. Prentice Hall International, 1993.
- [11] Sull, W., and Kashyap, R. L. A Self-Organizing Knowledge Representation Scheme for Extensible Heterogeneous Information Environment. IEEE Transactions on Knowledge and Data Engineering Vol.4 No.2 (April 1992).
- [12] Song, W. W., Desai, B. C., and Eaglestone, B. OFEM:An Approach towards Detection and Reduction of Objectives Inconsistencies. Proceedings of IDEAS'97 International Symposium on Database Engineering and Application (1997): 250-259.
- [13] Norman, R. J. Object-oriented systems analysis and design. New Jersey: Prentice Hall International, 1996.
- [14] Oracle8 documentation [CD-ROM]. Oracle Cooperation.
- [15] Kim, W. (ed.). Modern database systems. The object model, interperability, and beyond. ACM Press: Addison-Wesley, 1995.
- [16] Henderson-Sellers, B. Object-oriented metrics, measures of complexity. New Jersey: Prentice Hall PTR, 1996.
- [17] Sawatrakkiet, S., and Senivongse, T. Integrating OODB Schemas Using Heuristics and Richer Semantics. Proceedings of the National Computer Science and Engineering Conference: NCSEC (1999).
- [18] Fleming, C. C., and Halle, B. V. Handbook of relational database design. Addison-Wesley, 1989.
- [19] Navathe, B. C. Conceptual Database Design: An Entity-Relationship Approach. California: Benjamin/Cummings, 1992.



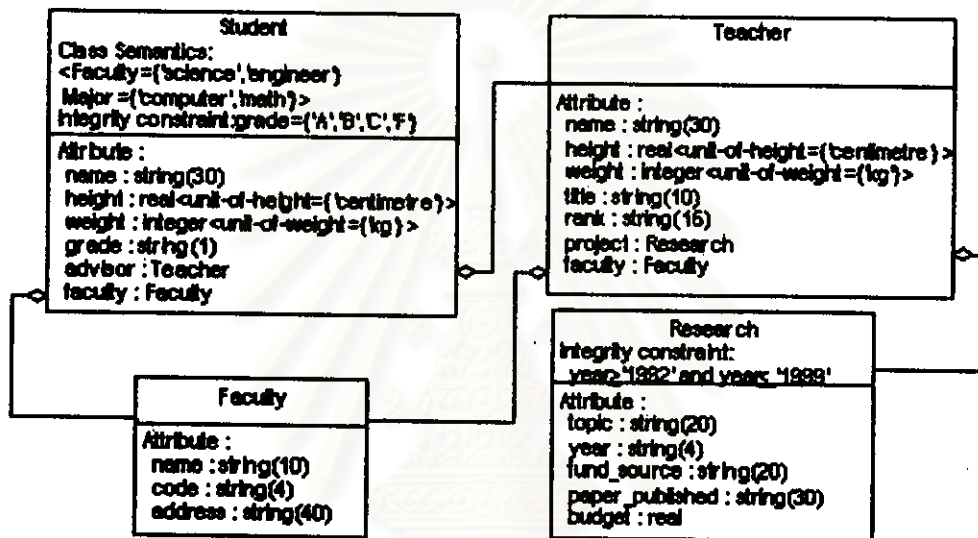
ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

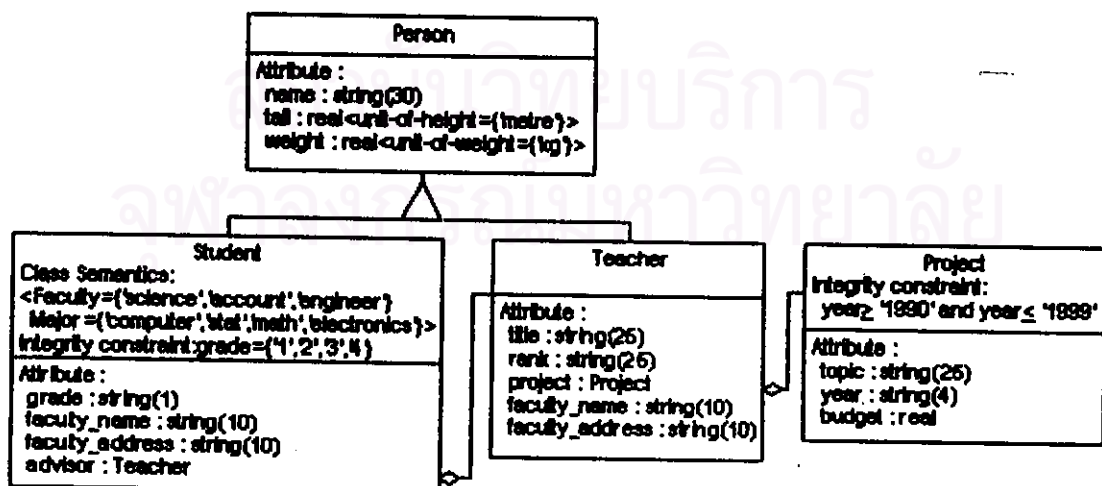
ภาคผนวก ก
ตัวอย่างที่ใช้ในการทดสอบ 10 คู่

ภาคผนวกนี้แสดงตัวอย่างของแบบจำลองข้อมูลที่นำมาทดสอบ โดยบางตัวอย่างเป็นเพียงบางส่วนของแบบจำลองข้อมูลทั้งหมดที่ออกแบบไว้สำหรับระบบหนึ่ง และมีบางตัวอย่างที่เป็นแบบจำลองข้อมูลทั้งหมดของระบบ

ตัวอย่างคู่ที่ 1

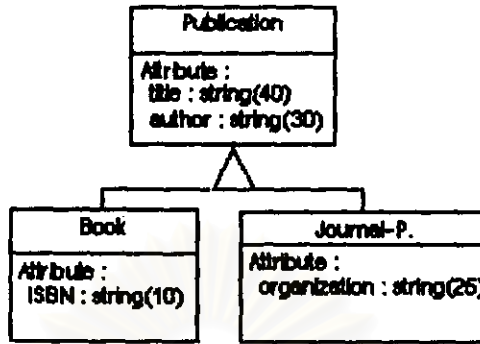


(Schema 1.1 คัดแปลงจาก [17])

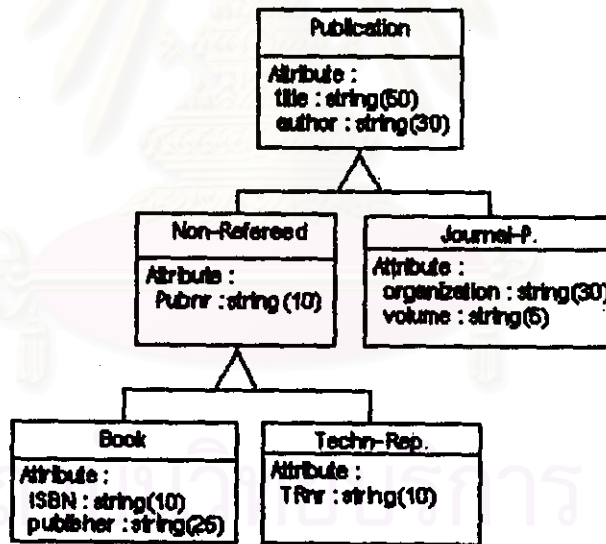


(Schema 1.2 คัดแปลงจาก [17])

ตัวอย่างที่ 2



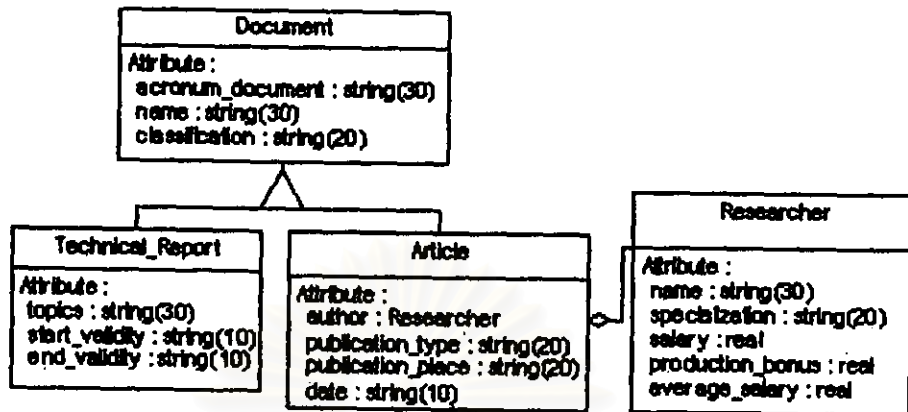
(Schema2.1 คัดแปลงจาก [8])



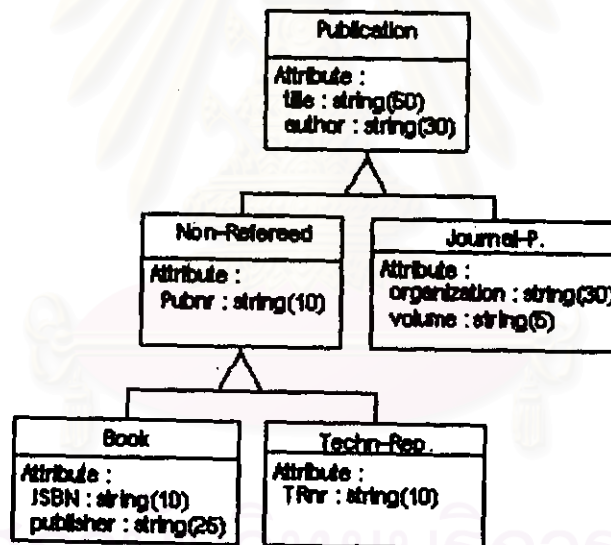
(Schema2.2 คัดแปลงจาก [8])

สํานักงานวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างที่ 3



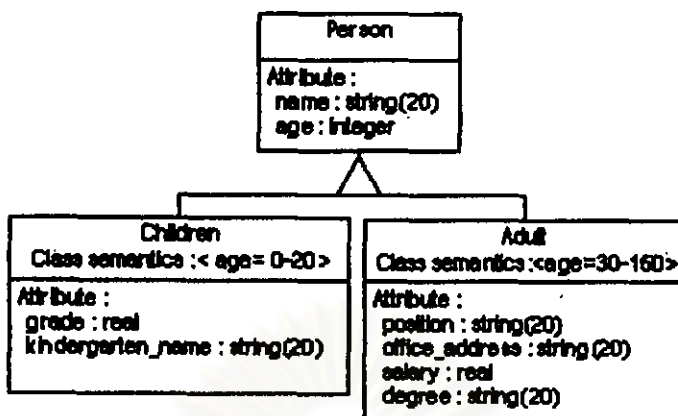
(Schema3.1)



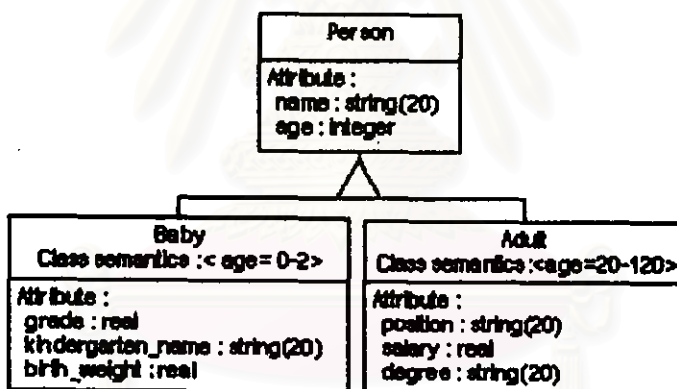
(Schema3.2 สืบเนื่องจาก [8])

จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างที่ 4



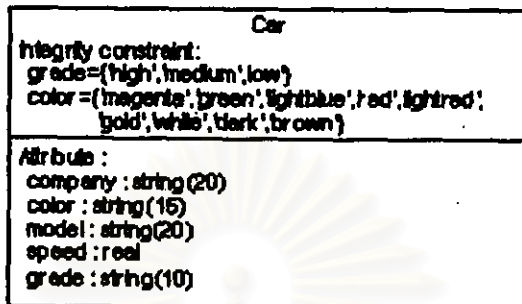
(Schema4.1)



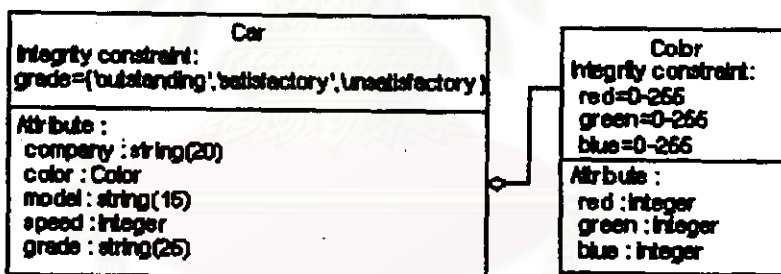
(Schema4.2)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างที่ 5

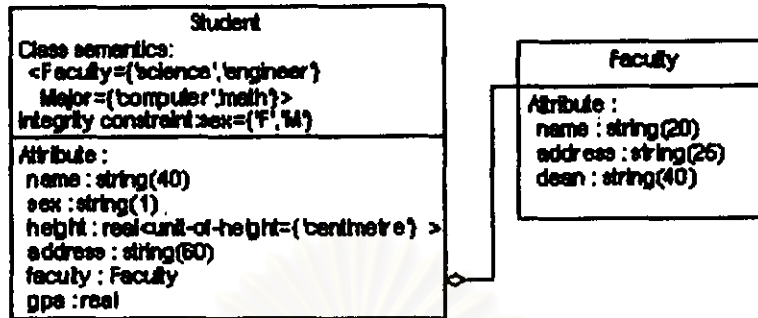


(Scheme5.1 คัดแปลงจาก [1])

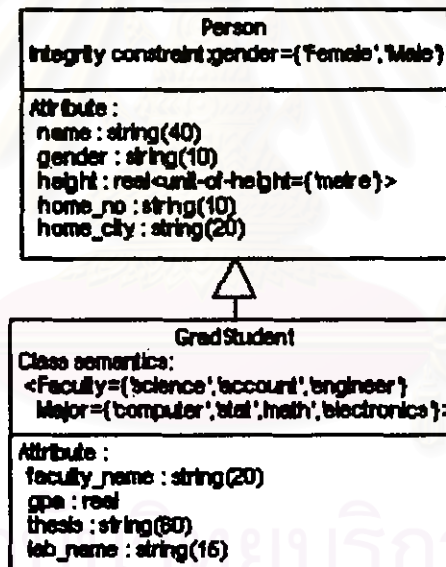


(Scheme5.2 คัดแปลงจาก [1])

ตัวอย่างที่ 6

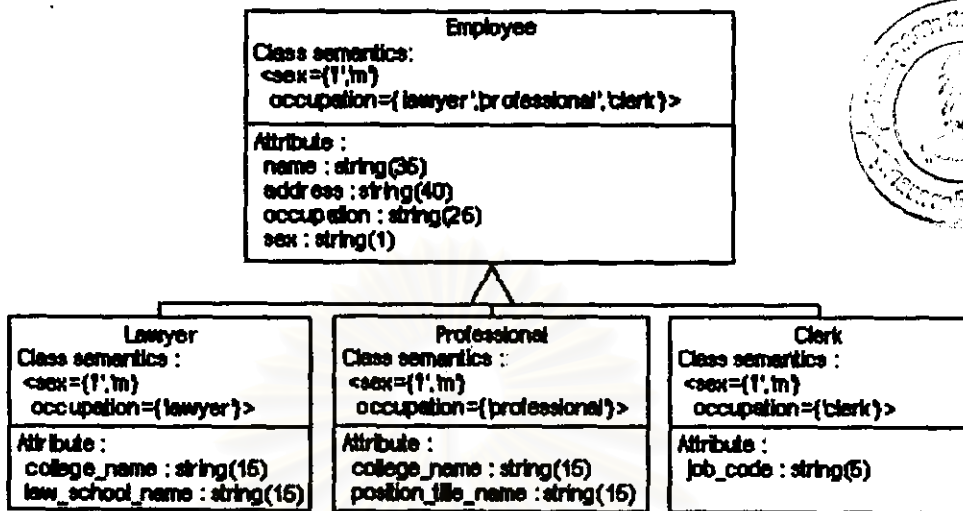


(Schema6.1)

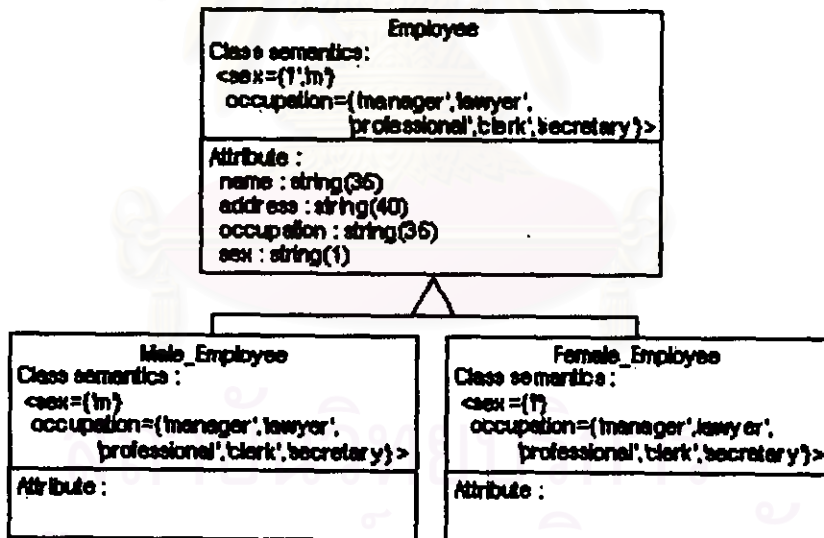


(Schema6.2)

ตัวอย่างที่ 7

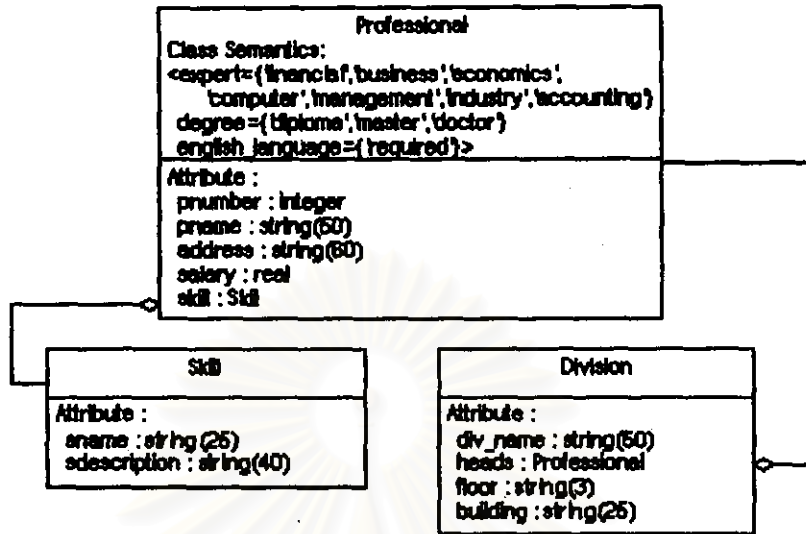


(Schema7.1 สืบเนื่องจาก [18])

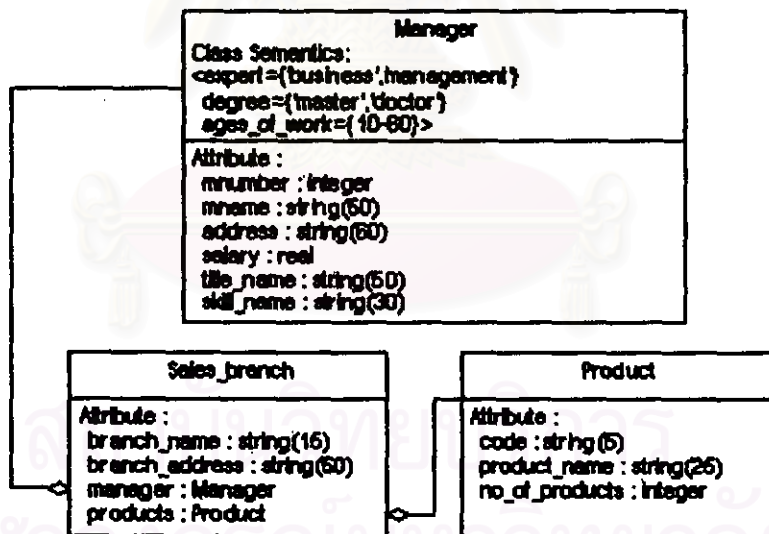


(Schema7.2 สืบเนื่องจาก [18])

ตัวอย่างคู่ที่ 8

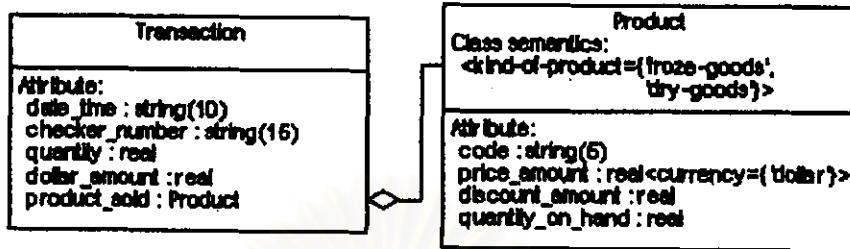


(Schema8.1)

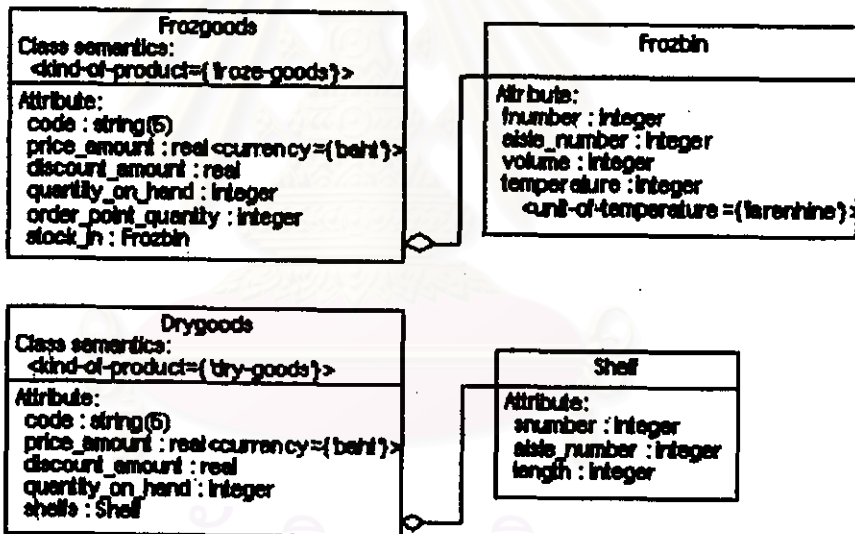


(Scheme8.2)

ตัวอย่างที่ 9



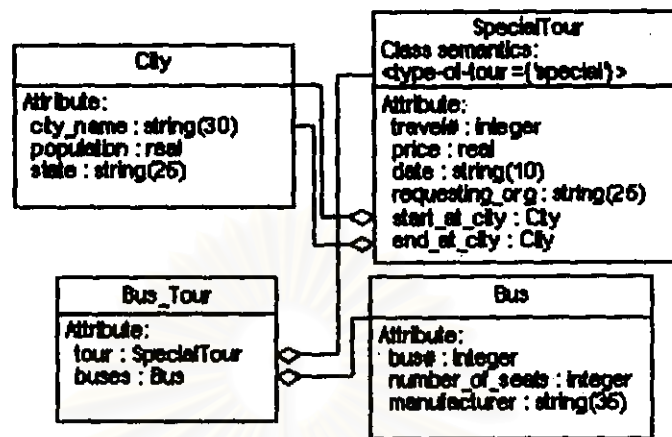
(Schema 9.1 ดัดแปลงจาก [18])



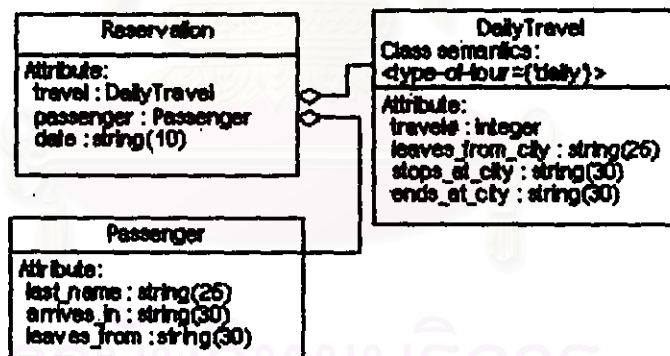
(Schema 9.2 ดัดแปลงจาก [18])

สถาบันวิจัยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างที่ 10



(Schema10.1 ดัดแปลงจาก [21])



(Schema10.2 ดัดแปลงจาก [21])

คำที่มีความหมายเหมือนกันหรือคำที่เป็นส่วนประกอบกันในบางคู่ของตัวอย่าง

Schema1.1 และ Schema1.2

Schema1.1.Student.faculty.name = Schema1.2.Student.faculty_name

Schema1.1.Student.faculty.address = Schema1.2.Student.faculty_address

Schema1.1.Teacher.faculty.name = Schema1.2.Teacher.faculty_name

Schema1.1.Teacher.faculty.address = Schema1.2.Teacher.faculty_address

Schema1.1.Student.height = Schema1.2.Student.tall

Schema1.1.Teacher.height = Schema1.2.Teacher.tall

Schema3.1 และ Schema3.2

Schema3.1.Technical_Report = Schema3.2.Techn-Rep

Schema3.1.Technical_Report.topics = Schema3.2.Techn-Rep.title

Schema3.1.Document = Schema3.2.Publication

Schema3.1.Document.name = Schema3.2.Publication.title

Schema3.1.Article = Schema3.2.Journal-P

Schema6.1 และ Schema6.2

Schema6.1.Student.sex = Schema6.2.GradStudent.gender

Schema6.1.Student.faculty.name = Schema6.2.GradStudent.faculty_name

Schema6.1.Student.address = Schema6.2.GradStudent.home_no + Schema6.2.GradStudent.home_city

Schema8.1 และ Schema8.2

Schema8.1.Professional.pname = Schema8.2.Manager.mname

Schema8.1.Professional.pnumber = Schema8.2.Manager.mnumber

Schema8.1.Professional.skill.skill_name = Schema8.2.Manager.skill_name

Schema10.1 และ Schema10.2

Schema10.1.SpecialTour.start_at_city.city_name = Schema10.2.DailyTravel.leaves_from_city

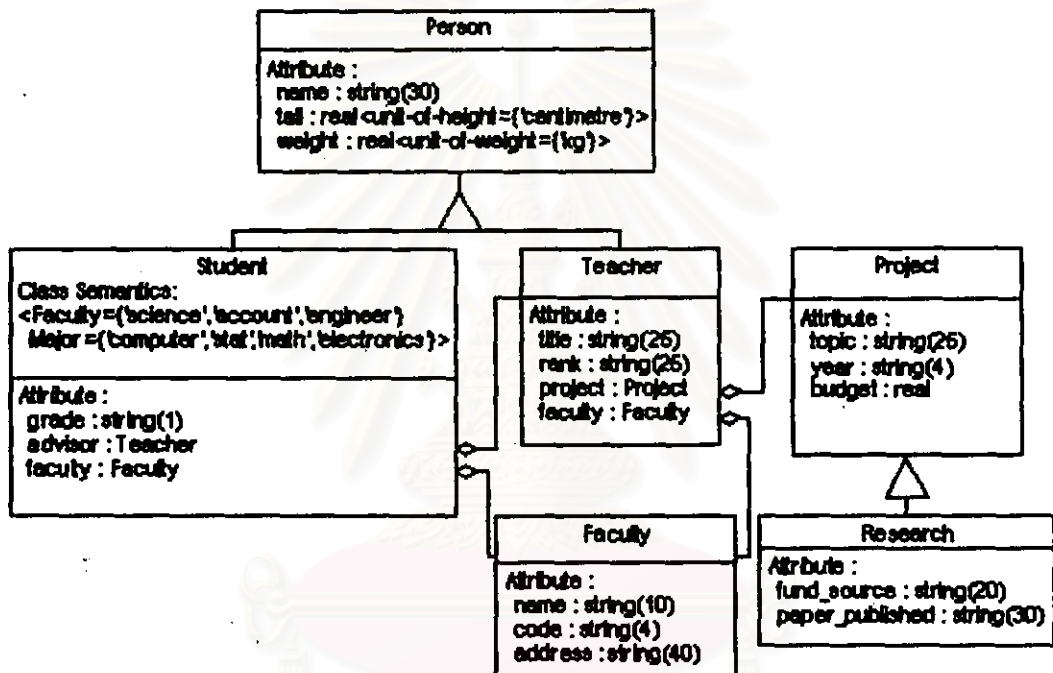
Schema10.1.SpecialTour.end_at_city.city_name = Schema10.2.DailyTravel.ends_at_city

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข
แบบจำลองข้อมูลรวมที่ได้จากการรวมโดยผู้เชี่ยวชาญเชิงวัตถุ

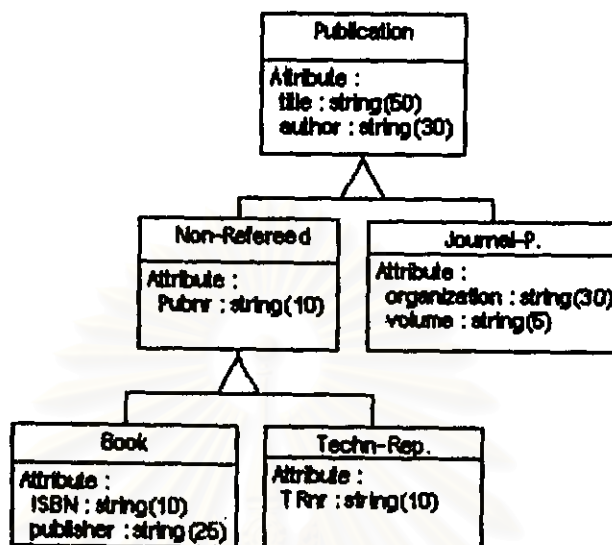
ในส่วนนี้จะแสดงแบบจำลองข้อมูลรวมที่ได้จากการรวมตัวอย่างแบบจำลองข้อมูล 10 คู่
ในภาคผนวก ก โดยอาศัยผู้เชี่ยวชาญเชิงวัตถุ

แบบจำลองข้อมูลรวมจากการรวมตัวอย่างคู่ที่ 1

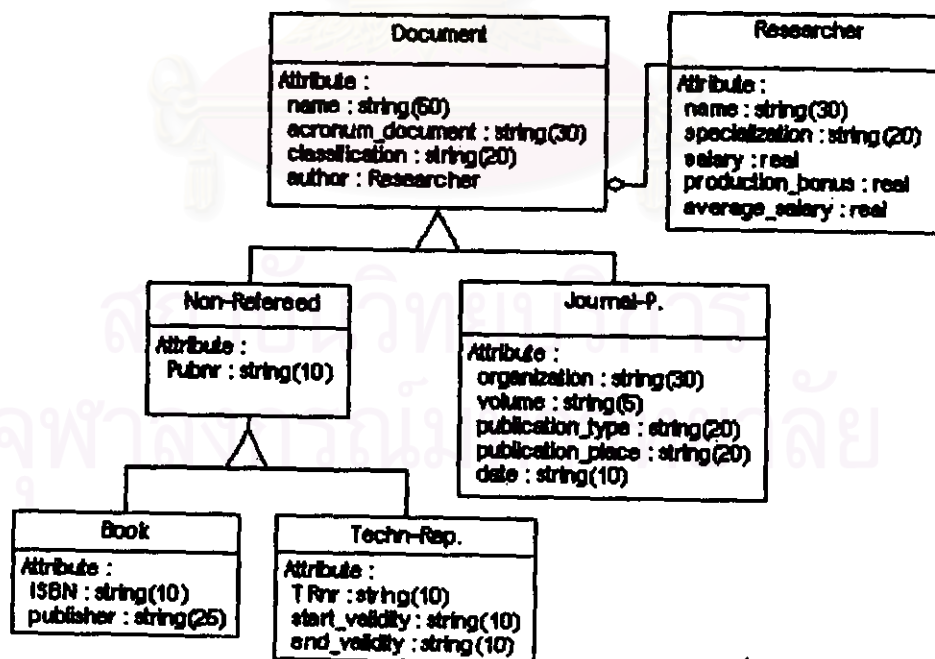


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

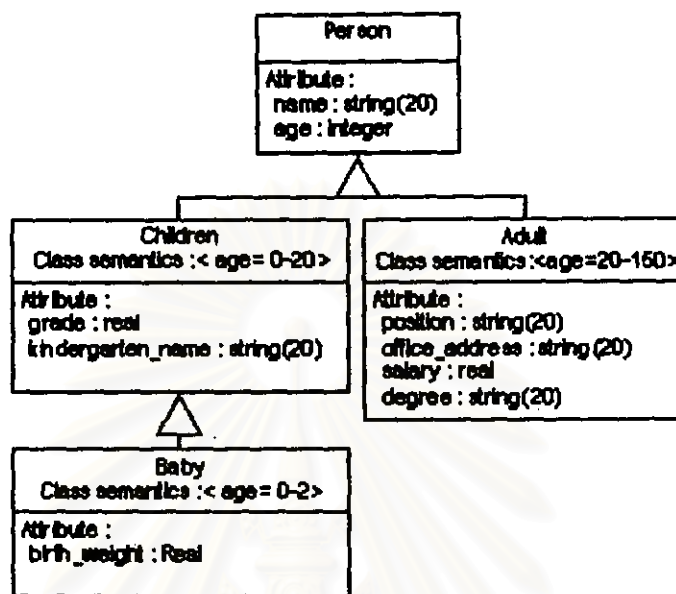
แบบจำลองข้อมูลรวมจากตารางตัวอย่างที่ 2



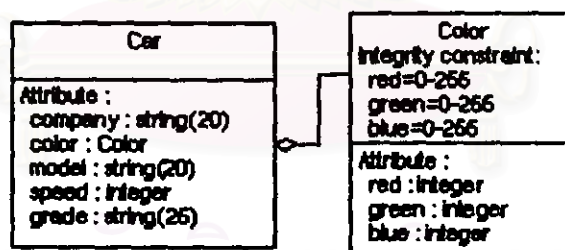
แบบจำลองข้อมูลรวมจากตารางตัวอย่างที่ 3



แบบจำลองข้อมูลรวมจากความสัมพันธ์ตัวอย่างที่ 4

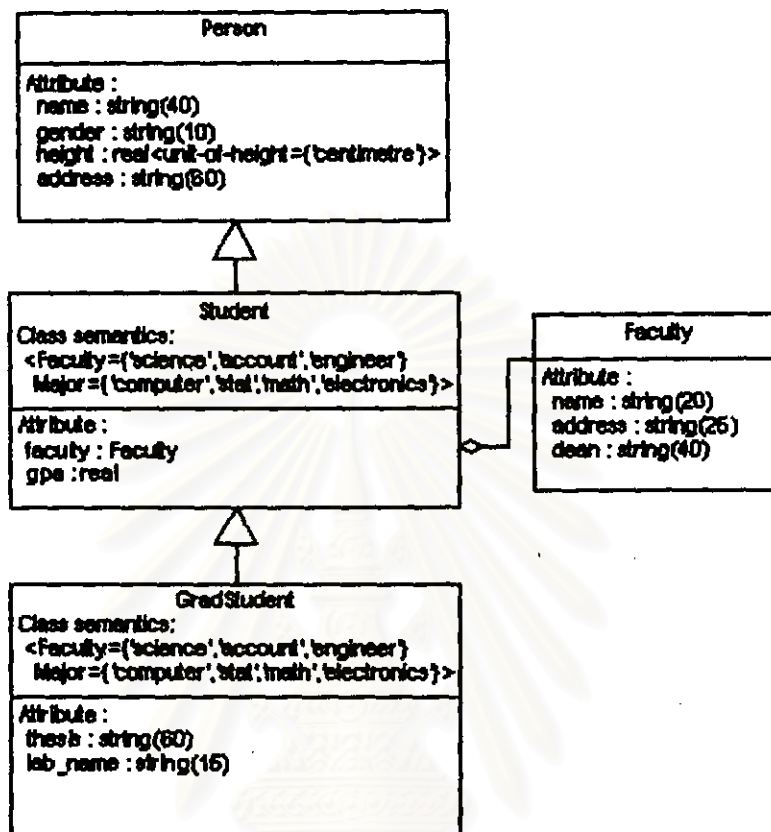


แบบจำลองข้อมูลรวมจากความสัมพันธ์ตัวอย่างที่ 5

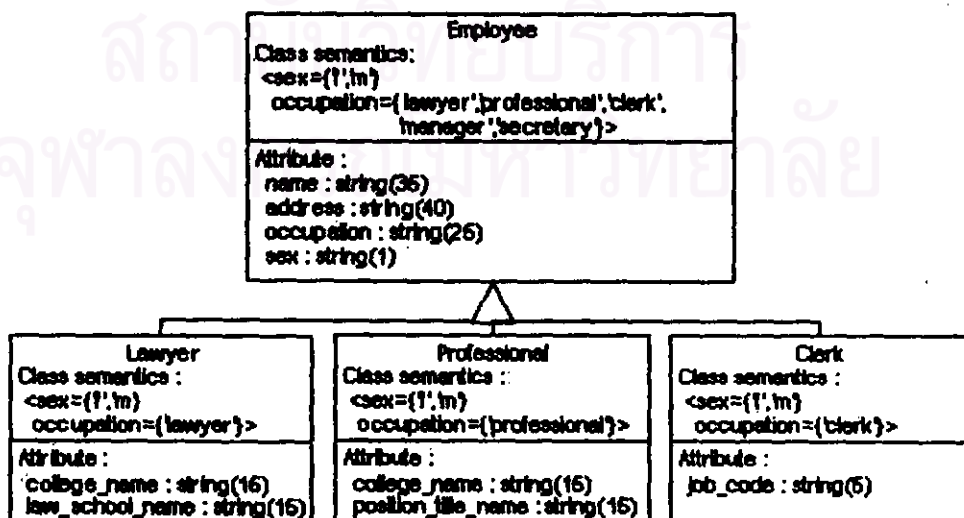


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

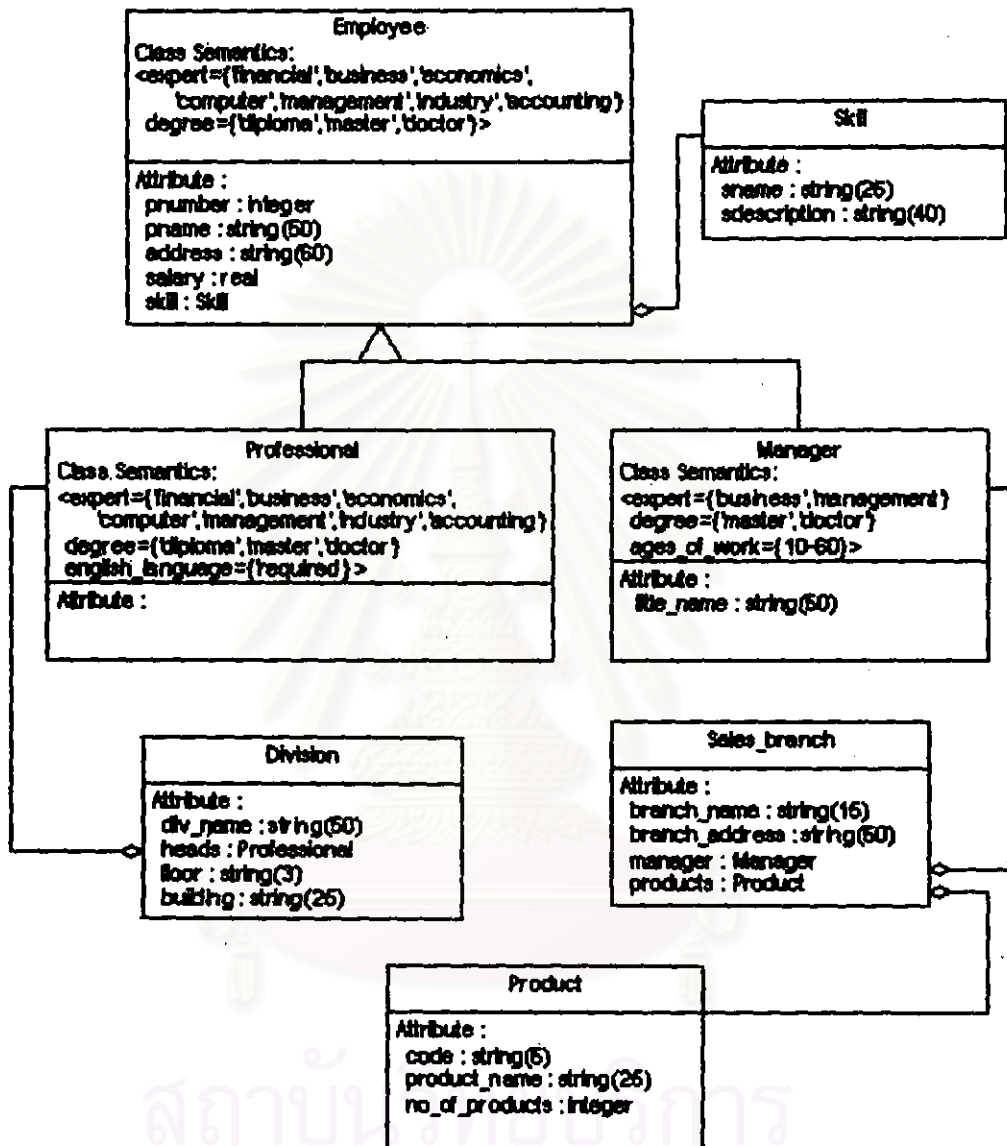
แบบจำลองข้อมูลรวมจากตารางตัวอย่างที่ 6



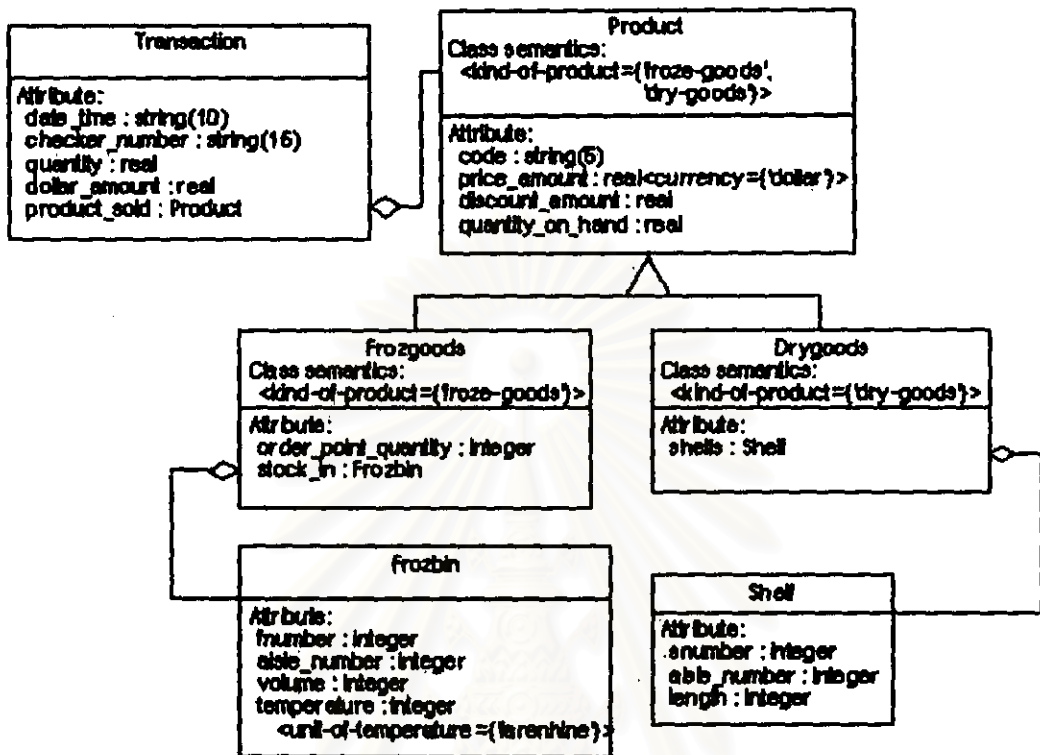
แบบจำลองข้อมูลรวมจากตารางตัวอย่างที่ 7



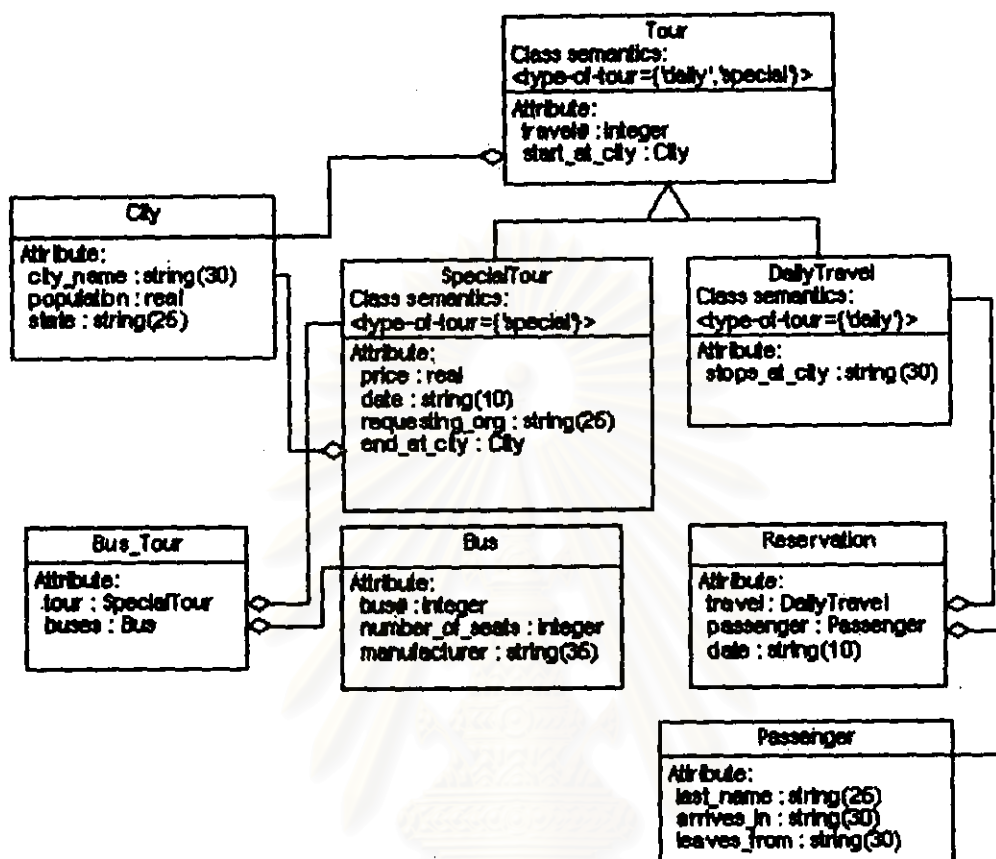
แบบจำลองข้อมูลรวมจากความสัมพันธ์อย่างคู่ที่ 8



แบบจำลองข้อมูลรวมจากความสัมพันธ์อย่างคู่ที่ 9



แบบจำลองข้อมูลรวมจากภาพรวมตัวอยู่ที่ 10



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

แบบจำลองข้อมูลรวมที่ได้จากการรวมตัวอย่าง 10 คู่จากต้นแบบ

แบบจำลองข้อมูลรวมจากการรวมตัวอย่างคู่ที่ 1

[Class] : Faculty (Source - Schema:schema1.1/Class:Faculty)

<Class Attribute>

[Attribute] : name

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute : name Class : Faculty (Path : Faculty) Schema : schema1.1

DataType : String Size : 10 Set : false

Attribute : faculty_name Class : Teacher Schema : schema1.2

DataType : String Size : 10 Set : false

Attribute : faculty_name Class : Student Schema : schema1.2

DataType : String Size : 10 Set : false

[Attribute] : code

DataType : string Size : 4 Set : false

<Map Attribute From> :

Attribute : code Class : Faculty (Path : Faculty) Schema : schema1.1

DataType : String Size : 4 Set : false

[Attribute] : address

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute : address Class : Faculty (Path : Faculty) Schema : schema1.1

DataType : String Size : 20 Set : false

Attribute : faculty_address Class : Teacher Schema : schema1.2

DataType : String Size : 10 Set : false

Attribute : faculty_address Class : Student Schema : schema1.2

DataType : String Size : 10 Set : false

[Class] : Research (Source - Schema:schema1.1/Class:Research)

Parent : Project

<Class Attribute>

[Attribute] : fund_source

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute : fund_source Class : Research (Path : Research) Schema : schema1.1

DataType : String Size : 20 Set : false

[Attribute] : paper_published

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute : paper_published Class : Research (Path : Research) Schema : schema1.1

DataType : String Size : 30 Set : false

[Class] : Teacher (Source - Schema:schema1.1/Class:Teacher , Schema:schema1.2/Class:Teacher)

Parent : Person

<Class Attribute>

[Attribute] : title

DataType : string Size : 25 Set : false

<Map Attribute From> :

Attribute : title Class : Teacher (Path : Teacher) Schema : schema1.1

DataType : String Size : 10 Set : false

Attribute : title Class : Teacher (Path : Teacher, Teacher-Person) Schema : schema1.2

DataType : String Size : 25 Set : false

[Attribute] : rank

DataType : string Size : 25 Set : false

<Map Attribute From> :

Attribute : rank Class : Teacher (Path : Teacher) Schema : schema1.1

DataType : String Size : 15 Set : false

Attribute : rank Class : Teacher (Path : Teacher, Teacher-Person) Schema : schema1.2

DataType : String Size : 25 Set : false

[Attribute] : project

DataType : Project Set : false

<Map Attribute From> :

Attribute : project Class : Teacher (Path : Teacher) Schema : schema1.1

DataType : Research Set : false

Attribute : project Class : Teacher (Path : Teacher, Teacher-Person) Schema : schema1.2

DataType : Project Set : false

[Attribute] : faculty

DataType : Faculty Set : false

<Map Attribute From> :

Attribute : faculty Class : Teacher (Path : Teacher) Schema : schema1.1

DataType : Faculty Set : false

[Class] : Student (Source - Schema:schema1.1/Class:Student , Schema:schema1.2/Class:Student)

Parent : Person

<Class Semantic>

Semantic : faculty Value : account,engineer,science

Map function : false

Semantic : major Value : computer,electronics,math,stat

Map function : false

<Class Attribute>

[Attribute] : grade

DataType : string Size : 1 Set : false

<Attribute Constraint> Possible Value:1,2,3,4,A,B,C,F

MapValueFunction : Required

<Map Attribute From> :

Attribute : grade Class : Student (Path : Student) Schema : schema1.1

DataType : String Size : 1 Set : false

Possible Value : A,B,C,F

Attribute :grade Class :Student (Path :Student, Student-Person) Schema :schema1.2

DataType:String Size :1 Set :false

Possible Value : 1,2,3,4

[Attribute] : advisor

DataType :Teacher Set :false

<Map Attribute From> :

Attribute :advisor Class :Student (Path :Student) Schema :schema1.1

DataType:Teacher Set :false

Attribute :advisor Class :Student (Path :Student, Student-Person) Schema :schema1.2

DataType:Teacher Set :false

[Attribute] : faculty

DataType :Faculty Set :false

<Map Attribute From> :

Attribute :faculty Class :Student (Path :Student) Schema :schema1.1

DataType:Faculty Set :false

[Class] : Project (Source - Schema:schema1.2/Class:Project)

<Class Attribute>

[Attribute] : topic

DataType :string Size :25 Set :false

<Map Attribute From> :

Attribute :topic Class :Research Schema :schema1.1

DataType:String Size :20 Set :false

Attribute :topic Class :Project (Path :Project) Schema :schema1.2

DataType:String Size :25 Set :false

[Attribute] : year

DataType :string Size : 4 Set : false

<Attribute Constraint> Possible Value:1990-1999

<Map Attribute From> :

Attribute :year Class :Research Schema :schema1.1

DataType:String Size :4 Set :false

Possible Value : 1992-1999

Attribute :year Class :Project (Path :Project) Schema :schema1.2

DataType:String Size :4 Set :false

Possible Value : 1990-1999

[Attribute] : budget

DataType :double Size :0 Set :false

<Map Attribute From> :

Attribute :budget Class :Research Schema :schema1.1

DataType:Real Size :0 Set :false

Attribute :budget Class :Project (Path :Project) Schema :schema1.2

DataType:Real Size :0 Set :false

[Class] : Person (Source - Schema:schema1.2/Class:Person)

<Class Attribute>

[Attribute] : name

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute :name Class :Person (Path :Person) Schema :schema1.2

DataType :String Size :30 Set : false

Attribute :name Class :Teacher Schema :schema1.1

DataType :String Size :30 Set : false

Attribute :name Class :Student Schema :schema1.1

DataType :String Size :30 Set : false

[Attribute] : tall

DataType :double Size : 0 Set : false

<Attribute Semantic> :

Semantic : unit-of-height Value : centimetre,metre

Map function : true

For Semantic :

- unit-of-height = centimetre(Schema:schema1.1 Class:Teacher Attribute:height)

unit-of-height = metre(Schema:schema1.2 Class:Person Attribute:tall)

- unit-of-height = centimetre(Schema:schema1.1 Class:Student Attribute:height)

unit-of-height = metre(Schema:schema1.2 Class:Person Attribute:tall)

<Map Attribute From> :

Attribute :tall Class :Person (Path :Person) Schema :schema1.2

DataType :Real Size :0 Set : false

Attribute :height Class :Teacher Schema :schema1.1

DataType :Real Size :0 Set : false

Attribute :height Class :Student Schema :schema1.1

DataType :Real Size :0 Set : false

[Attribute] : weight

DataType :double Size : 0 Set : false

<Attribute Semantic> :

Semantic : unit-of-weight Value : kg

Map function : false

<Map Attribute From> :

Attribute :weight Class :Person (Path :Person) Schema :schema1.2

DataType :Real Size :0 Set : false

Attribute :weight Class :Teacher Schema :schema1.1

DataType :integer Size :0 Set : false

MapDataTypeFunction : Required

Attribute :weight Class :Student Schema :schema1.1

DataType :Integer Size :0 Set : false

MapDataTypeFunction : Required

แบบจำลองข้อมูลรวมจากภาพรวมตัวอย่างที่ 2

[Class] : Publication (Source - Schema:schema2.1/Class:Publication , Schema:schema2.2/Class:Publication)

<Class Attribute>

[Attribute] : title

DataType : string Size : 50 Set : false

<Map Attribute From> :

Attribute :title Class :Publication (Path :Publication) Schema :schema2.1

DataType :String Size :40 Set : false

Attribute :title Class :Publication (Path :Publication) Schema :schema2.2

DataType :String Size :50 Set : false

[Attribute] : author

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute :author Class :Publication (Path :Publication) Schema :schema2.1

DataType :String Size :30 Set : false

Attribute :author Class :Publication (Path :Publication) Schema :schema2.2

DataType :String Size :30 Set : false

[Class] : Book (Source - Schema:schema2.1/Class:Book , Schema:schema2.2/Class:Book)

Parent : Non-Referenced

<Class Attribute>

[Attribute] : ISBN

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute :ISBN Class :Book (Path :Book, Book-Publication) Schema :schema2.1

DataType :String Size :10 Set : false

Attribute :ISBN Class :Book (Path :Book, Book-Non-Referenced) Schema :schema2.2

DataType :String Size :10 Set : false

[Attribute] : publisher

DataType : string Size : 25 Set : false

<Map Attribute From> :

Attribute :publisher Class :Book (Path :Book, Book-Non-Referenced) Schema :schema2.2

DataType :String Size :25 Set : false

[Class] : Journal-P (Source - Schema:schema2.1/Class:Journal-P , Schema:schema2.2/Class:Journal-P)

Parent : Publication

<Class Attribute>

[Attribute] : organization

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute :organization Class :Journal-P (Path :Journal-P, Journal-P-Publication) Schema :schema2.1

DataType :String Size :25 Set : false

Attribute :organization Class :Journal-P (Path :Journal-P, Journal-P-Publication) Schema :schema2.2

DataType :String Size :30 Set : false

[Attribute] : volume

DataType : string Size : 5 Set : false

<Map Attribute From> :

Attribute : volume Class : Journal-P (Path : Journal-P, Journal-P-Publication) Schema : schema2.2

DataType : String Size : 6 Set : false

[Class] : Non-Refereed (Source - Schema:schema2.2/Class:Non-Refereed)

Parent : Publication

<Class Attribute>

[Attribute] : Pubnr

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute : Pubnr Class : Non-Refereed (Path : Non-Refereed, Non-Refereed-Publication) Schema : schema2.2

DataType : String Size : 10 Set : false

[Class] : Techn-Rep (Source - Schema:schema2.2/Class:Techn-Rep)

Parent : Non-Refereed

<Class Attribute>

[Attribute] : TRnr

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute : TRnr Class : Techn-Rep (Path : Techn-Rep, Techn-Rep-Non-Refereed) Schema : schema2.2

DataType : String Size : 10 Set : false

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แบบจำลองข้อมูลรวมจากการรวมตัวอย่างที่ 3

[Class] : Document (Source - Schema:schema3.1/Class:Document , Schema:schema3.2/Class:Publication)

<Class Attribute>

[Attribute] : name

DataType : string Size : 60 Set : false

<Map Attribute From> :

Attribute : name Class : Document (Path : Document) Schema : schema3.1

DataType : String Size : 30 Set : false

Attribute : title Class : Publication (Path : Publication) Schema : schema3.2

DataType : String Size : 50 Set : false

Attribute : topics Class : Technical_Report Schema : schema3.1

DataType : String Size : 30 Set : false

[Attribute] : acronym_document

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute : acronym_document Class : Document (Path : Document) Schema : schema3.1

DataType : String Size : 30 Set : false

[Attribute] : classification

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute : classification Class : Document (Path : Document) Schema : schema3.1

DataType : String Size : 20 Set : false

[Attribute] : author

DataType : Researcher Set : false

<Map Attribute From> :

Attribute : author Class : Publication (Path : Publication) Schema : schema3.2

DataType : String Size : 30 Set : false

MapDataTypeFunction : Required

Attribute : author Class : Article Schema : schema3.1

DataType : Researcher Set : false

[Class] : Technical_Report (Source - Schema:schema3.1/Class:Technical_Report , Schema:schema3.2/Class:Techn-Rep)

Parent : Document, Non-Refered

<Class Attribute>

[Attribute] : start_validity

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute : start_validity Class : Technical_Report Schema : schema3.1

DataType : String Size : 10 Set : false

[Attribute] : end_validity

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute : end_validity Class : Technical_Report Schema : schema3.1

DataType : String Size : 10 Set : false

[Attribute] : TRnr

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute :TRnr Class :Techn-Rep Schema :schema3.2

DataType :String Size :10 Set : false

[Class] : Article (Source - Schema:schema3.1/Class:Article , Schema:schema3.2/Class:Journal-P)

Parent : Document

<Class Attribute>

[Attribute] : publication_type

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute :publication_type Class :Article (Path :Article, Article-Documnt) Schema :schema3.1

DataType :String Size :20 Set : false

[Attribute] : publication_place

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute :publication_place Class :Article (Path :Article, Article-Documnt) Schema :schema3.1

DataType :String Size :20 Set : false

[Attribute] : date

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute :date Class :Article (Path :Article, Article-Documnt) Schema :schema3.1

DataType :String Size :10 Set : false

[Attribute] : organization

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute :organization Class :Journal-P (Path :Journal-P, Journal-P-Publication) Schema :schema3.2

DataType :String Size :30 Set : false

[Attribute] : volume

DataType : string Size : 5 Set : false

<Map Attribute From> :

Attribute :volume Class :Journal-P (Path :Journal-P, Journal-P-Publication) Schema :schema3.2

DataType :String Size :5 Set : false

[Class] : Researcher (Source - Schema:schema3.1/Class:Researcher)

<Class Attribute>

[Attribute] : name

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute :name Class :Researcher (Path :Researcher) Schema :schema3.1

DataType :String Size :30 Set : false

[Attribute] : specialization

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute :specialization Class :Researcher (Path :Researcher) Schema :schema3.1

```

    DataType:String Size:20 Set:false
  [Attribute]: salary
    DataType:double Size:0 Set:false
  <Map Attribute From>:
    Attribute:salary Class:Researcher (Path:Researcher) Schema:schema3.1
    DataType:Real Size:0 Set:false
  [Attribute]: production_bonus
    DataType:double Size:0 Set:false
  <Map Attribute From>:
    Attribute:production_bonus Class:Researcher (Path:Researcher) Schema:schema3.1
    DataType:Real Size:0 Set:false
  [Attribute]: average_salary
    DataType:double Size:0 Set:false
  <Map Attribute From>:
    Attribute:average_salary Class:Researcher (Path:Researcher) Schema:schema3.1
    DataType:Real Size:0 Set:false

[Class] : Non-Refered (Source - Schema:schema3.2/Class:Non-Refered)
Parent: Document
<Class Attribute>
  [Attribute]: Pubnr
    DataType:string Size:10 Set:false
  <Map Attribute From>:
    Attribute:Pubnr Class:Non-Refered (Path:Non-Refered, Non-Refered-Publication) Schema:schema3.2
    DataType:String Size:10 Set:false

[Class] : Book (Source - Schema:schema3.2/Class:Book)
Parent: Non-Refered
<Class Attribute>
  [Attribute]: ISBN
    DataType:string Size:10 Set:false
  <Map Attribute From>:
    Attribute:ISBN Class:Book (Path:Book, Book-Non-Refered) Schema:schema3.2
    DataType:String Size:10 Set:false
  [Attribute]: publisher
    DataType:string Size:26 Set:false
  <Map Attribute From>:
    Attribute:publisher Class:Book (Path:Book, Book-Non-Refered) Schema:schema3.2
    DataType:String Size:26 Set:false

```

แบบจำลองข้อมูลรวมจากการรวมตัวครั้งที่ 4

[Class] : Person (Source - Schema:schema4.1/Class:Person , Schema:schema4.2/Class:Person)

<Class Attribute>

[Attribute] : name

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute :name Class:Person (Path:Person) Schema:schema4.1

DataType:String Size:20 Set:false

Attribute :name Class:Person (Path:Person) Schema:schema4.2

DataType:String Size:20 Set:false

[Attribute] : age

DataType:integer Size:0 Set:false

<Map Attribute From> :

Attribute:age Class:Person (Path:Person) Schema:schema4.1

DataType:integer Size:0 Set:false

Attribute:age Class:Person (Path:Person) Schema:schema4.2

DataType:integer Size:0 Set:false

[Class] : Children (Source - Schema:schema4.1/Class:Children)

Parent : Person

<Class Semantic>

Semantic : age Value : 0-20

Map function : false

<Class Attribute>

[Attribute] : grade

DataType : double Size : 0 Set : false

<Map Attribute From> :

Attribute :grade Class:Children (Path:Children, Children-Person) Schema:schema4.1

DataType:Real Size:0 Set:false

Attribute:grade Class:Baby Schema:schema4.2

DataType:Real Size:0 Set:false

[Attribute] : kindergarten_name

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute :kindergarten_name Class:Children (Path:Children, Children-Person) Schema:schema4.1

DataType:String Size:20 Set:false

Attribute :kindergarten_name Class:Baby Schema:schema4.2

DataType:String Size:20 Set:false

[Class] : Adult (Source - Schema:schema4.1/Class:Adult , Schema:schema4.2/Class:Adult)

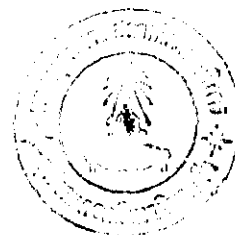
Parent : Person

<Class Semantic>

Semantic : age Value : 20-150

Map function : false

<Class Attribute>



[Attribute] : position

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute : position Class : Adult (Path : Adult, Adult-Person) Schema : schema4.1

DataType : String Size : 20 Set : false

Attribute : position Class : Adult (Path : Adult, Adult-Person) Schema : schema4.2

DataType : String Size : 20 Set : false

[Attribute] : salary

DataType : double Size : 0 Set : false

<Map Attribute From> :

Attribute : salary Class : Adult (Path : Adult, Adult-Person) Schema : schema4.1

DataType : Real Size : 0 Set : false

Attribute : salary Class : Adult (Path : Adult, Adult-Person) Schema : schema4.2

DataType : Real Size : 0 Set : false

[Attribute] : degree

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute : degree Class : Adult (Path : Adult, Adult-Person) Schema : schema4.1

DataType : String Size : 20 Set : false

Attribute : degree Class : Adult (Path : Adult, Adult-Person) Schema : schema4.2

DataType : String Size : 20 Set : false

[Attribute] : office_address

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute : office_address Class : Adult (Path : Adult, Adult-Person) Schema : schema4.1

DataType : String Size : 20 Set : false

[Class] : Baby (Source - Schema:schema4.2/Class:Baby)

Parent : Children

<Class Semantic>

Semantic : age Value : 0-2

Map function : false

<Class Attribute>

[Attribute] : birth_weight

DataType : double Size : 0 Set : false

<Map Attribute From> :

Attribute : birth_weight Class : Baby (Path : Baby, Baby-Person) Schema : schema4.2

DataType : Real Size : 0 Set : false

แบบจำลองข้อมูลรวมจากการรวมตัวอย่างครั้งที่ 5

[Class] : Car (Source - Schema:schema5.1/Class:Car , Schema:schema5.2/Class:Car)

<Class Attribute>

[Attribute] : model

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute :model Class :Car (Path :Car) Schema :schema5.1

DataType :String Size :20 Set : false

Attribute :model Class :Car (Path :Car) Schema :schema5.2

DataType :String Size :15 Set : false

[Attribute] : company

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute :company Class :Car (Path :Car) Schema :schema5.1

DataType :String Size :20 Set : false

Attribute :company Class :Car (Path :Car) Schema :schema5.2

DataType :String Size :20 Set : false

[Attribute] : color

DataType : Color Set : false

<Map Attribute From> :

Attribute :color Class :Car (Path :Car) Schema :schema5.1

DataType :String Size :15 Set : false

MapDataTypeFunction : Required

Possible Value : magenta,green,lightblue,red,lightred,gold,white,dark,brown

Attribute :color Class :Car (Path :Car) Schema :schema5.2

DataType :Color Set : false

[Attribute] : speed

DataType : double Size : 0 Set : false

<Map Attribute From> :

Attribute :speed Class :Car (Path :Car) Schema :schema5.1

DataType :Real Size :0 Set : false

Attribute :speed Class :Car (Path :Car) Schema :schema5.2

DataType :Integer Size :0 Set : false

MapDataTypeFunction : Required

[Attribute] : grade

DataType : string Size : 25 Set : false

<Attribute Constraint> Possible Value:high,low,medium,outstanding,satisfactory,unsatisfactory

MapValueFunction : Required

<Map Attribute From> :

Attribute :grade Class :Car (Path :Car) Schema :schema5.1

DataType :String Size :10 Set : false

Possible Value : high,medium,low

Attribute :grade Class :Car (Path :Car) Schema :schema5.2

DataType :String Size :25 Set : false

Possible Value : outstanding,satisfactory,unsatisfactory

[Class] : Color (Source - Schema:schema5.2/Class:Color)

<Class Attribute>

[Attribute] : red

DataType : integer Size : 0 Set : false

<Attribute Constraint> Possible Value:0-255

<Map Attribute From> :

Attribute :red Class :Color (Path :Color) Schema :schema5.2

DataType :integer Size :0 Set : false

Possible Value : 0-255

[Attribute] : green

DataType : Integer Size : 0 Set : false

<Attribute Constraint> Possible Value:0-255

<Map Attribute From> :

Attribute :green Class :Color (Path :Color) Schema :schema5.2

DataType :integer Size :0 Set : false

Possible Value : 0-255

[Attribute] : blue

DataType :integer Size :0 Set : false

<Attribute Constraint> Possible Value:0-255

<Map Attribute From> :

Attribute :blue Class :Color (Path :Color) Schema :schema5.2

DataType :integer Size :0 Set : false

Possible Value : 0-255



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แบบจำลองข้อมูลรวมจากการรวมตัวอย่างที่ 6

[Class] : Student (Source - Schema:schema6.1/Class:Student)

Parent : Person

<Class Semantic>

Semantic : Faculty Value : account,engineer,science

Map function : false

Semantic : Major Value : computer,electronics,math,stat

Map function : false

<Class Attribute>

[Attribute] : gpa

DataType : double Size : 0 Set : false

<Map Attribute From> :

Attribute :gpa Class :Student (Path :Student) Schema :schema6.1

DataType :Real Size :0 Set : false

Attribute :gpa Class :GradStudent Schema :schema6.2

DataType :Real Size :0 Set : false

[Attribute] : faculty

DataType : Faculty Set : false

<Map Attribute From> :

Attribute :faculty Class :Student (Path :Student) Schema :schema6.1

DataType :Faculty Set : false

[Class] : Faculty (Source - Schema:schema6.1/Class:Faculty)

<Class Attribute>

[Attribute] : name

DataType : string Size : 20 Set : false

<Map Attribute From> :

Attribute :name Class :Faculty (Path :Faculty) Schema :schema6.1

DataType :String Size :20 Set : false

Attribute :faculty_name Class :GradStudent Schema :schema6.2

DataType :String Size :20 Set : false

[Attribute] : dean

DataType : string Size : 40 Set : false

<Map Attribute From> :

Attribute :dean Class :Faculty (Path :Faculty) Schema :schema6.1

DataType :String Size :40 Set : false

[Attribute] : address

DataType : string Size : 25 Set : false

<Map Attribute From> :

Attribute :address Class :Faculty (Path :Faculty) Schema :schema6.1

DataType :String Size :25 Set : false

[Class] : Person (Source - Schema:schema6.2/Class:Person)

<Class Attribute>

[Attribute] : name

DataType : string Size : 40 Set : false

<Map Attribute From> :

Attribute :name Class :Student Schema :schema6.1

DataType :String Size :40 Set : false

Attribute :name Class :Person (Path :Person) Schema :schema6.2

DataType :String Size :40 Set : false

[Attribute] : sex

DataType : string Size : 10 Set : false

<Attribute Constraint> Possible Value:F,Female,M,Male

MapValueFunction : Required

<Map Attribute From> :

Attribute :sex Class :Student Schema :schema6.1

DataType :String Size :1 Set : false

Possible Value : F,M

Attribute :gender Class :Person (Path :Person) Schema :schema6.2

DataType :String Size :10 Set : false

Possible Value : Female, Male

[Attribute] : height

DataType : double Size : 0 Set : false

<Attribute Semantic> :

Semantic : unit-of-height Value : centimetre,metre

Map function : true

For Semantic :

- unit-of-height = centimetre(Schema:schema6.1 Class:Student Attribute:height)

unit-of-height = metre(Schema:schema6.2 Class:Person Attribute:height)

<Map Attribute From> :

Attribute :height Class :Student Schema :schema6.1

DataType :Real Size :0 Set : false

Attribute :height Class :Person (Path :Person) Schema :schema6.2

DataType :Real Size :0 Set : false

[Attribute] : address

DataType : string Size : 60 Set : false

<Map Attribute From> :

Attribute :address Class :Student Schema :schema6.1

DataType :String Size :60 Set : false

Compound Attributes :

home_no Class :Person Schema :schema6.2

DataType :String Size :10 Set : false

home_city Class :Person Schema :schema6.2

DataType :String Size :20 Set : false

[Class] : GradStudent (Source - Schema:schema6.2/Class:GradStudent)

Parent : Student

<Class Semantic>

Semantic : Faculty Value : science,account,engineer

Map function : false

Semantic : Major Value : computer,stat,math,electronics

Map function : false

<Class Attribute>

[Attribute] : thesis

DataType : string Size : 60 Set : false

<Map Attribute From> :

Attribute :thesis Class :GradStudent (Path :GradStudent, GradStudent-Person) Schema :schema6.2

DataType :String Size :60 Set : false

[Attribute] : lab_name

DataType : string Size : 15 Set : false

<Map Attribute From> :

Attribute :lab_name Class :GradStudent (Path :GradStudent, GradStudent-Person) Schema :schema6.2

DataType :String Size :15 Set : false



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แบบจำลองข้อมูลรวมจากการรวมตัวครั้งที่ 7

[Class] : Employee_Male_Employee_Female_Employee

(Source - Schema:schema7.1/Class:Employee , Schema:schema7.2/Class:Employee ,

Schema:schema7.2/Class:Male_Employee , Schema:schema7.2/Class:Female_Employee)

<Class Semantic>

Semantic : sex Value : f,m

Map function : false

Semantic : occupation Value : clerk,lawyer,manager,professional,secretary

Map function : false

<Class Attribute>

[Attribute] : name

DataType : string Size : 35 Set : false

<Map Attribute From> :

Attribute :name Class :Employee (Path :Employee) Schema :schema7.1

DataType :String Size :35 Set :false

Attribute :name Class :Employee Schema :schema7.2

DataType :String Size :35 Set :false

[Attribute] : address

DataType : string Size : 40 Set : false

<Map Attribute From> :

Attribute :address Class :Employee (Path :Employee) Schema :schema7.1

DataType :String Size :40 Set :false

Attribute :address Class :Employee (Path :Employee, Female_Employee, Female_Employee-Employee,

Male_Employee, Male_Employee-Employee) Schema :schema7.2

DataType :String Size :40 Set :false

[Attribute] : occupation

DataType : string Size : 35 Set : false

<Map Attribute From> :

Attribute :occupation Class :Employee (Path :Employee) Schema :schema7.1

DataType :String Size :26 Set :false

Attribute :occupation Class :Employee Schema :schema7.2

DataType :String Size :35 Set :false

[Attribute] : sex

DataType : string Size : 1 Set : false

<Map Attribute From> :

Attribute :sex Class :Employee (Path :Employee) Schema :schema7.1

DataType :String Size :1 Set :false

Attribute :sex Class :Employee Schema :schema7.2

DataType :String Size :1 Set :false

[Class] : Lawyer (Source - Schema:schema7.1/Class:Lawyer)

Parent : Employee_Male_Employee_Female_Employee

<Class Semantic>

Semantic : sex Value : f,m

Map function : false

Semantic : occupation Value : lawyer

Map function : false

<Class Attribute>

[Attribute] : college_name

DataType : string Size : 15 Set : false

<Map Attribute From> :

Attribute : college_name Class : Lawyer (Path : Lawyer, Lawyer-Employee) Schema : schema7.1

DataType : String Size : 15 Set : false

[Attribute] : law_school_name

DataType : string Size : 15 Set : false

<Map Attribute From> :

Attribute : law_school_name Class : Lawyer (Path : Lawyer, Lawyer-Employee) Schema : schema7.1

DataType : String Size : 15 Set : false

[Class] : Professional (Source - Schema:schema7.1/Class:Professional)

Parent : Employee_Male_Employee_Female_Employee

<Class Semantic>

Semantic : occupation Value : professional

Map function : false

Semantic : sex Value : f,m

Map function : false

<Class Attribute>

[Attribute] : college_name

DataType : string Size : 15 Set : false

<Map Attribute From> :

Attribute : college_name Class : Professional (Path : Professional, Professional-Employee) Schema : schema7.1

DataType : String Size : 15 Set : false

[Attribute] : position_title_name

DataType : string Size : 15 Set : false

<Map Attribute From> :

Attribute : position_title_name Class : Professional (Path : Professional, Professional-Employee) Schema : schema7.1

DataType : String Size : 15 Set : false

[Class] : Clerk (Source - Schema:schema7.1/Class:Clerk)

Parent : Employee_Male_Employee_Female_Employee

<Class Semantic>

Semantic : occupation Value : clerk

Map function : false

Semantic : sex Value : f,m

Map function : false

<Class Attribute>

[Attribute] : job_code

DataType : string Size : 5 Set : false

<Map Attribute From> :

Attribute : job_code Class : Clerk (Path : Clerk, Clerk-Employee) Schema : schema7.1

DataType : String Size : 5 Set : false

แบบจำลองข้อมูลรวมจากการรวมตัวอย่างที่ 8

[Class] : Professional (Source - Schema:schema8.1/Class:Professional)

Parent : Professional_Manager

<Class Semantic>

Semantic : expert Value : financial,business,economics,computer,management,industry,accounting

Map function : false

Semantic : degree Value : diploma,master,doctor

Map function : false

Semantic : english_language Value : required

Map function : false

[Class] : Skill (Source - Schema:schema8.1/Class:Skill)

<Class Attribute>

[Attribute] : skill_name

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute : skill_name Class : Skill (Path : Skill) Schema : schema8.1

DataType : String Size : 25 Set : false

Attribute : skill_name Class : Manager Schema : schema8.2

DataType : String Size : 30 Set : false

[Attribute] : skill_description

DataType : string Size : 40 Set : false

<Map Attribute From> :

Attribute : skill_description Class : Skill (Path : Skill) Schema : schema8.1

DataType : String Size : 40 Set : false

[Class] : Division (Source - Schema:schema8.1/Class:Division)

<Class Attribute>

[Attribute] : div_name

DataType : string Size : 50 Set : false

<Map Attribute From> :

Attribute : div_name Class : Division (Path : Division) Schema : schema8.1

DataType : String Size : 50 Set : false

[Attribute] : heads

DataType : Professional Set : false

<Attribute Related In Aggregated Class> : Professional

<Map Attribute From> :

Attribute : heads Class : Division (Path : Division) Schema : schema8.1

DataType : Professional Set : false

[Attribute] : floor

DataType : string Size : 3 Set : false

<Map Attribute From> :

Attribute : floor Class : Division (Path : Division) Schema : schema8.1

DataType : String Size : 3 Set : false

[Attribute] : building

DataType : string Size : 25 Set : false

<Map Attribute From> :

Attribute : building Class : Division (Path : Division) Schema : schema8.1

DataType : String Size : 25 Set : false

[Class] : Manager (Source - Schema:schema8.2/Class:Manager)

Parent : Professional_Manager

<Class Semantic>

Semantic : expert Value : business,management

Map function : false

Semantic : degree Value : master,doctor

Map function : false

Semantic : ages_of_work Value : 10-60

Map function : false

<Class Attribute>

[Attribute] : title_name

DataType : string Size : 50 Set : false

<Map Attribute From> :

Attribute : title_name Class : Manager (Path : Manager) Schema : schema8.2

DataType : String Size : 50 Set : false

[Class] : Sales_Branch (Source - Schema:schema8.2/Class:Sales_Branch)

<Class Attribute>

[Attribute] : branch_name

DataType : string Size : 15 Set : false

<Map Attribute From> :

Attribute : branch_name Class : Sales_Branch (Path : Sales_Branch) Schema : schema8.2

DataType : String Size : 15 Set : false

[Attribute] : manager

DataType : Manager Set : false

<Attribute Related In Aggregated Class> : Manager

Attribute : title_name

Map Attribute From :

Attribute : title_name Class : Manager Schema : schema8.2

<Map Attribute From> :

Attribute : manager Class : Sales_Branch (Path : Sales_Branch) Schema : schema8.2

DataType : Manager Set : false

[Attribute] : branch_address

DataType : string Size : 50 Set : false

<Map Attribute From> :

Attribute : branch_address Class : Sales_Branch (Path : Sales_Branch) Schema : schema8.2

DataType : String Size : 50 Set : false

[Attribute] : products

DataType : Product Set : false

<Map Attribute From> :

Attribute :products Class :Sales_Branch (Path :Sales_Branch) Schema :schema8.2
 DataType :Product Set :false

[Class] : Product (Source - Schema:schema8.2/Class:Product)

<Class Attribute>

[Attribute] : code

DataType :string Size :5 Set :false

<Map Attribute From> :

Attribute :code Class :Product (Path :Product) Schema :schema8.2

DataType :String Size :5 Set :false

[Attribute] : product_name

DataType :string Size :25 Set :false

<Map Attribute From> :

Attribute :product_name Class :Product (Path :Product) Schema :schema8.2

DataType :String Size :25 Set :false

[Attribute] : no_of_products

DataType :integer Size :0 Set :false

<Map Attribute From> :

Attribute :no_of_products Class :Product (Path :Product) Schema :schema8.2

DataType :Integer Size :0 Set :false

[Class] : Professional_Manager

<Class Semantic>

Semantic :expert Value :accounting,business,computer,economics,financial,industry,management

Map function :false

Semantic :degree Value :diploma,doctor,master

Map function :false

<Class Attribute>

[Attribute] : pnumber

DataType :integer Size :0 Set :false

<Map Attribute From> :

Attribute :pnumber Class :Professional Schema :schema8.1

DataType :Integer Size :0 Set :false

Attribute :mnumber Class :Manager Schema :schema8.2

DataType :Integer Size :0 Set :false

[Attribute] : pname

DataType :string Size :50 Set :false

<Map Attribute From> :

Attribute :pname Class :Professional Schema :schema8.1

DataType :String Size :50 Set :false

Attribute :mname Class :Manager Schema :schema8.2

DataType :String Size :50 Set :false

[Attribute] : address

DataType :string Size :60 Set :false

<Map Attribute From> :

Attribute :address Class :Professional Schema :schema8.1

DataType :String Size :60 Set : false

Attribute :address Class :Manager Schema :schema8.2

DataType :String Size :60 Set : false

[Attribute] : salary

DataType :double Size :0 Set : false

<Map Attribute From> :

Attribute :salary Class :Professional Schema :schema8.1

DataType :Real Size :0 Set : false

Attribute :salary Class :Manager Schema :schema8.2

DataType :Real Size :0 Set : false

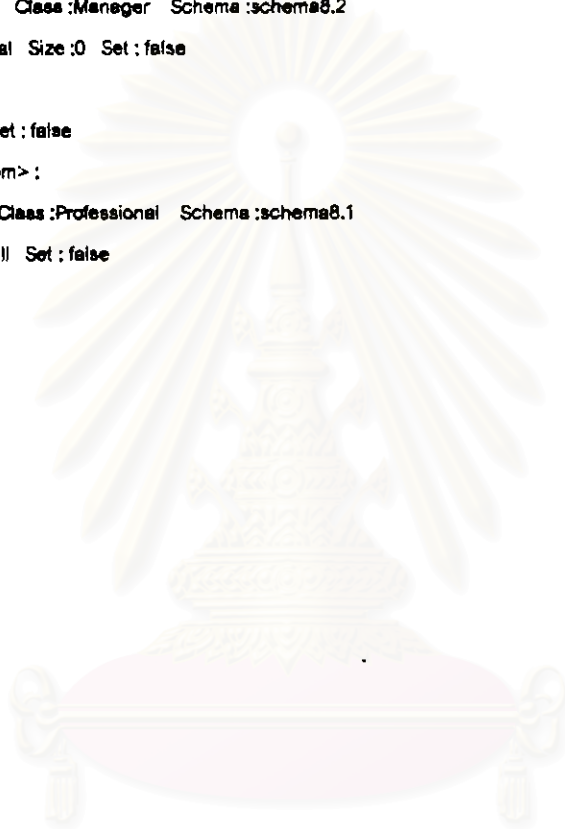
[Attribute] : skill

DataType :Skill Set : false

<Map Attribute From> :

Attribute :skill Class :Professional Schema :schema8.1

DataType :Skill Set : false



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แบบจำลองข้อมูลรวมจากการรวมตัวอย่างที่ 9

[Class] : Product (Source - Schema:schema9.1/Class:Product)

<Class Semantic>

Semantic : kind-of-product Value : dry-goods,froze-goods

Map function : false

<Class Attribute>

[Attribute] : code

DataType : string Size : 5 Set : false

<Map Attribute From> :

Attribute :code Class :Product (Path :Product) Schema :schema9.1

DataType :String Size :5 Set : false

Attribute :code Class :Frozgoods Schema :schema9.2

DataType :String Size :5 Set : false

Attribute :code Class :Drygoods Schema :schema9.2

DataType :String Size :5 Set : false

[Attribute] : price_amount

DataType : double Size : 0 Set : false

<Attribute Semantic> :

Semantic : currency Value : baht,dollar

Map function : true

For Semantic :

- currency = dollar(Schema:schema9.1 Class:Product Attribute:price_amount)

currency = baht(Schema:schema9.2 Class:Drygoods Attribute:price_amount)

<Map Attribute From> :

Attribute :price_amount Class :Product (Path :Product) Schema :schema9.1

DataType :Real Size :0 Set : false

Attribute :price_amount Class :Frozgoods Schema :schema9.2

DataType :Real Size :0 Set : false

Attribute :price_amount Class :Drygoods Schema :schema9.2

DataType :Real Size :0 Set : false

[Attribute] : discount_amount

DataType : double Size : 0 Set : false

<Map Attribute From> :

Attribute :discount_amount Class :Product (Path :Product) Schema :schema9.1

DataType :Real Size :0 Set : false

Attribute :discount_amount Class :Frozgoods Schema :schema9.2

DataType :Real Size :0 Set : false

Attribute :discount_amount Class :Drygoods Schema :schema9.2

DataType :Real Size :0 Set : false

[Attribute] : quantity_on_hand

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute :quantity_on_hand Class :Product (Path :Product) Schema :schema9.1

DataType :integer Size :0 Set : false

Attribute :quantity_on_hand Class :Frozgoods Schema :schema9.2

DataType:Integer Size:0 Set:false
 Attribute:quantity_on_hand Class:Drygoods Schema:schema9.2
 DataType:Integer Size:0 Set:false

[Class] : Transaction (Source - Schema:schema9.1/Class:Transaction)

<Class Attribute>

[Attribute]: date_time

DataType:string Size:10 Set:false

<Map Attribute From> :

Attribute:date_time Class:Transaction (Path:Transaction) Schema:schema9.1

DataType:String Size:10 Set:false

[Attribute]: checker_number

DataType:string Size:15 Set:false

<Map Attribute From> :

Attribute:checker_number Class:Transaction (Path:Transaction) Schema:schema9.1

DataType:String Size:15 Set:false

[Attribute]: quantity

DataType:integer Size:0 Set:false

<Map Attribute From> :

Attribute:quantity Class:Transaction (Path:Transaction) Schema:schema9.1

DataType:integer Size:0 Set:false

[Attribute]: dollar_amount

DataType:double Size:0 Set:false

<Map Attribute From> :

Attribute:dollar_amount Class:Transaction (Path:Transaction) Schema:schema9.1

DataType:Real Size:0 Set:false

[Attribute]: product_sold

DataType:Product Set:false

<Map Attribute From> :

Attribute:product_sold Class:Transaction (Path:Transaction) Schema:schema9.1

DataType:Product Set:false

[Class] : Frozbin (Source - Schema:schema9.2/Class:Frozbin)

<Class Attribute>

[Attribute]: fnumber

DataType:integer Size:0 Set:false

<Map Attribute From> :

Attribute:fnumber Class:Frozbin (Path:Frozbin) Schema:schema9.2

DataType:integer Size:0 Set:false

[Attribute]: aisle_number

DataType:integer Size:0 Set:false

<Map Attribute From> :

Attribute:aisle_number Class:Frozbin (Path:Frozbin) Schema:schema9.2

DataType:integer Size:0 Set:false

[Attribute]: volume

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute : volume Class : Frozbin (Path : Frozbin) Schema : schema9.2

DataType : integer Size : 0 Set : false

[Attribute] : temperature

DataType : integer Size : 0 Set : false

<Attribute Semantic> :

Semantic : unit-of-temperature Value : fahrenheit

Map function : false

<Map Attribute From> :

Attribute : temperature Class : Frozbin (Path : Frozbin) Schema : schema9.2

DataType : integer Size : 0 Set : false

[Class] : Frozgoods (Source - Schema:schema9.2/Class:Frozgoods)

Parent : Product

<Class Semantic>

Semantic : kind-of-product Value : froze-goods

Map function : false

<Class Attribute>

[Attribute] : order_point_quantity

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute : order_point_quantity Class : Frozgoods (Path : Frozgoods) Schema : schema9.2

DataType : integer Size : 0 Set : false

[Attribute] : stock_in

DataType : Frozbin Set : false

<Map Attribute From> :

Attribute : stock_in Class : Frozgoods (Path : Frozgoods) Schema : schema9.2

DataType : Frozbin Set : false

[Class] : Shelf (Source - Schema:schema9.2/Class:Shelf)

<Class Attribute>

[Attribute] : snumber

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute : snumber Class : Shelf (Path : Shelf) Schema : schema9.2

DataType : integer Size : 0 Set : false

[Attribute] : aisle_number

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute : aisle_number Class : Shelf (Path : Shelf) Schema : schema9.2

DataType : integer Size : 0 Set : false

[Attribute] : length

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute :length Class :Shelf (Path :Shelf) Schema :schema9.2
 DataType :Integer Size :0 Set : false

[Class] : Drygoods (Source - Schema:schema9.2/Class:Drygoods)

Parent : Product

<Class Semantic>

Semantic : kind-of-product Value : dry-goods

Map function : false

<Class Attribute>

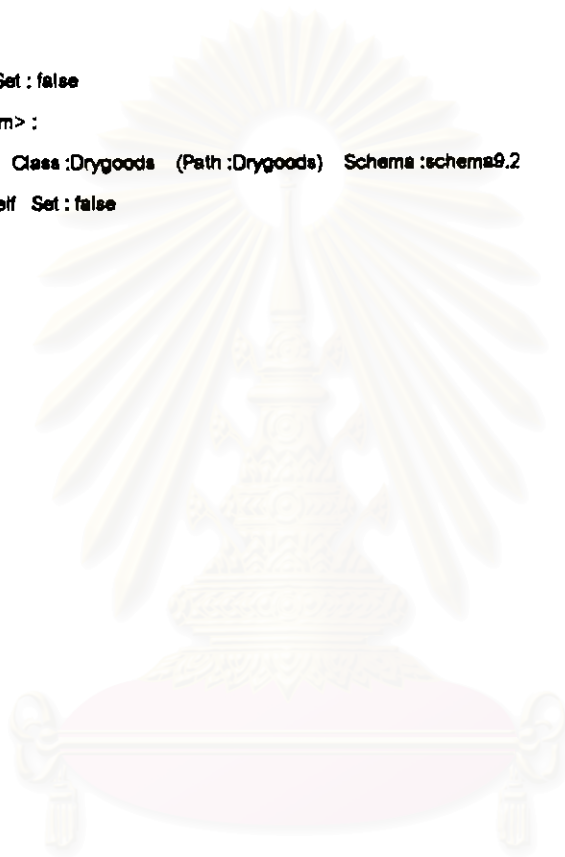
[Attribute] : shelves

DataType : Shelf Set : false

<Map Attribute From> :

Attribute :shelves Class :Drygoods (Path :Drygoods) Schema :schema9.2

DataType :Shelf Set : false



สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย

แบบจำลองข้อมูลรวมจากกรณีตัวอย่างที่ 10

[Class] : City (Source - Schema:schema10.1/Class:City)

<Class Attribute>

[Attribute] : city_name

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute : city_name Class : City (Path : City) Schema : schema10.1

DataType : String Size : 30 Set : false

Attribute : leaves_from_city Class : DailyTravel Schema : schema10.2

DataType : String Size : 25 Set : false

Attribute : ends_at_city Class : DailyTravel Schema : schema10.2

DataType : String Size : 30 Set : false

[Attribute] : population

DataType : double Size : 0 Set : false

<Map Attribute From> :

Attribute : population Class : City (Path : City) Schema : schema10.1

DataType : Real Size : 0 Set : false

[Attribute] : state

DataType : string Size : 25 Set : false

<Map Attribute From> :

Attribute : state Class : City (Path : City) Schema : schema10.1

DataType : String Size : 25 Set : false

[Class] : SpecialTour (Source - Schema:schema10.1/Class:SpecialTour)

Parent : SpecialTour_DailyTravel

<Class Semantic>

Semantic : type-of-tour Value : special

Map function : false

<Class Attribute>

[Attribute] : price

DataType : double Size : 0 Set : false

<Map Attribute From> :

Attribute : price Class : SpecialTour (Path : SpecialTour) Schema : schema10.1

DataType : Real Size : 0 Set : false

[Attribute] : date

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute : date Class : SpecialTour (Path : SpecialTour) Schema : schema10.1

DataType : String Size : 10 Set : false

[Attribute] : requesting_org

DataType : string Size : 25 Set : false

<Map Attribute From> :

Attribute : requesting_org Class : SpecialTour (Path : SpecialTour) Schema : schema10.1

DataType : String Size : 25 Set : false

[Attribute] : end_at_city

DataType : City Set : false

<Map Attribute From> :

Attribute : end_at_city Class : SpecialTour (Path : SpecialTour) Schema : schema10.1

DataType : City Set : false

[Class] : Bus (Source - Schema:schema10.1/Class:Bus)

<Class Attribute>

[Attribute] : bus#

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute : bus# Class : Bus (Path : Bus) Schema : schema10.1

DataType : Integer Size : 0 Set : false

[Attribute] : number_of_seats

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute : number_of_seats Class : Bus (Path : Bus) Schema : schema10.1

DataType : Integer Size : 0 Set : false

[Attribute] : manufacturer

DataType : string Size : 35 Set : false

<Map Attribute From> :

Attribute : manufacturer Class : Bus (Path : Bus) Schema : schema10.1

DataType : String Size : 35 Set : false

[Class] : Bus_Tour (Source - Schema:schema10.1/Class:Bus_Tour)

<Class Attribute>

[Attribute] : tour

DataType : SpecialTour Set : false

<Map Attribute From> :

Attribute : tour Class : Bus_Tour (Path : Bus_Tour) Schema : schema10.1

DataType : SpecialTour Set : false

[Attribute] : buses

DataType : Bus Set : false

<Map Attribute From> :

Attribute : buses Class : Bus_Tour (Path : Bus_Tour) Schema : schema10.1

DataType : Bus Set : false

[Class] : DailyTravel (Source - Schema:schema10.2/Class:DailyTravel)

Parent : SpecialTour_DailyTravel

<Class Semantic>

Semantic : type-of-tour Value : daily

Map function : false

<Class Attribute>

[Attribute] : stops_at_city

DataType : string Size : 30 Set : true

<Map Attribute From> :

Attribute :stops_at_city Class :DailyTravel (Path :DailyTravel) Schema :schema10.2
 DataType :String Size :30 Set : true

[Class] : Reservation (Source - Schema:schema10.2/Class:Reservation)

<Class Attribute>

[Attribute] : travel

DataType : DailyTravel Set : false

<Attribute Related In Aggregated Class> : DailyTravel

Attribute : stops_at_city

Map Attribute From :

Attribute:stops_at_city Class:DailyTravel Schema:schema10.2

<Map Attribute From> :

Attribute :travel Class :Reservation (Path :Reservation) Schema :schema10.2

DataType :DailyTravel Set : false

[Attribute] : passenger

DataType : Passenger Set : false

<Map Attribute From> :

Attribute :passenger Class :Reservation (Path :Reservation) Schema :schema10.2

DataType :Passenger Set : false

[Attribute] : date

DataType : string Size : 10 Set : false

<Map Attribute From> :

Attribute :date Class :Reservation (Path :Reservation) Schema :schema10.2

DataType :String Size :10 Set : false

[Class] : Passenger (Source - Schema:schema10.2/Class:Passenger)

<Class Attribute>

[Attribute] : last_name

DataType : string Size : 25 Set : false

<Map Attribute From> :

Attribute :last_name Class :Passenger (Path :Passenger) Schema :schema10.2

DataType :String Size :25 Set : false

[Attribute] : arrives_in

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute :arrives_in Class :Passenger (Path :Passenger) Schema :schema10.2

DataType :String Size :30 Set : false

[Attribute] : leaves_from

DataType : string Size : 30 Set : false

<Map Attribute From> :

Attribute :leaves_from Class :Passenger (Path :Passenger) Schema :schema10.2

DataType :String Size :30 Set : false

[Class] : SpecialTour_DailyTravel

<Class Semantic>

Semantic : type-of-tour Value : daily,special

Map function : true

For Semantic :

- type-of-tour = special(Schema:schema10.1 Class:SpecialTour)

type-of-tour = daily(Schema:schema10.2 Class:DailyTravel)

<Class Attribute>

[Attribute] : travel#

DataType : integer Size : 0 Set : false

<Map Attribute From> :

Attribute :travel# Class :SpecialTour Schema :schema10.1

DataType :Integer Size:0 Set : false

Attribute :travel# Class :DailyTravel Schema :schema10.2

DataType :Integer Size:0 Set : false

[Attribute] : start_at_city

DataType : City Set : false

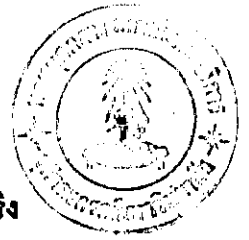
<Map Attribute From> :

Attribute :start_at_city Class :SpecialTour Schema :schema10.1

DataType :City Set : false



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ง

ตัวอย่างการนำแบบจำลองข้อมูลมาสร้างวิบบนฐานข้อมูลจริง

แบบจำลองข้อมูลรวมที่ได้จะนำไปสร้างเป็นวิบบนฐานข้อมูลออร์เคิล 8 เพื่อใช้ในการเข้าถึงข้อมูลจากสกีมาดั้งเดิม 2 ชุด โดยตัวอย่างที่จะนำมาสร้างเป็นวิบบนฐานข้อมูลนี้ ได้แก่แบบจำลองข้อมูลรวมจากการรวมตัวอย่างคู่ที่ 1 และ 5 โดยจะถือว่าแต่ละตัวอย่างที่จะนำมารวมนี้ได้มีการสร้างเป็นสกีมาบนฐานข้อมูลบนออร์เคิลเช่นกัน และมีข้อมูลในแต่ละสกีมาด้วย

การสร้างวิบบนฐานข้อมูลตามแบบจำลองข้อมูลรวมที่ได้จากการรวมตัวอย่างคู่ที่ 1

```
SQL> create type t_faculty as object
```

```
( NAME          VARCHAR2(10),
  CODE          CHAR(4),
  ADDRESS       VARCHAR2(20),
  SCHEMA        VARCHAR2(15)
);
```

```
SQL> create type t_person as object
```

```
( NAME          VARCHAR2(30),
  TALL          NUMBER,
  WEIGHT        NUMBER,
  SCHEMA        VARCHAR2(15)
);
```

```
SQL> create type t_project as object
```

```
( TOPIC         VARCHAR2(26),
  YEAR          CHAR(4),
  BUDGET        NUMBER,
  SCHEMA        VARCHAR2(15)
);
```

```
SQL> create type t_research as object
```

```
( FUND_SOURCE   VARCHAR2(20),
  PAPER_PUBLISHED VARCHAR2(30),
  PROJECT       REF T_PROJECT,
  SCHEMA        VARCHAR2(15),
  TOPIC         VARCHAR2(26),
  YEAR          CHAR(4)
);
```

SQL> create type t_teacher as object

```
( PERSON          REF T_PERSON,
  TITLE           VARCHAR2(25),
  RANK            VARCHAR2(25),
  PROJECT         REF T_PROJECT,
  SCHEMA          VARCHAR2(15),
  NAME            VARCHAR2(30)
);
```

SQL> create type t_student as object

```
( PERSON          REF T_PERSON,
  GRADE           CHAR(1),
  ADVISOR         REF T_TEACHER,
  FACULTY         REF T_FACULTY,
  SCHEMA          VARCHAR2(15),
  NAME            VARCHAR2(30)
);
```

SQL> create or replace view faculty of t_faculty with object oid(schema,code,name)

```
as select a.name,a.code,a.address,'local1_1'schema from local1_1.faculty a
union
select b.faculty_name,'---',b.faculty_address,'local1_2'schema from local1_2.student b;
```

SQL> create or replace view person of t_person with object oid(schema,name)

```
as select a.name,a.height,a.weight,'local1_1' schema from local1_1.student a
union
select a.name,a.height,a.weight,'local1_1' schema from local1_1.teacher a
union
select a.name,a.tail*100,a.weight,'local1_2' schema from local1_2.person a;
```

SQL> create or replace view project of t_project with object oid(schema,topic,year)

```
as select topic,year,budget,'local1_2' schema from local1_2.project
union
select topic,year,budget,'local1_1' schema from local1_1.research;
```

SQL> create or replace view research of t_research with object oid(schema,topic,year)

```
as select a.fund_source,a.paper_published,ref(b)project,
'local1_1'schema,a.topic,a.year from local1_1.research a,project b
where b.schema = 'local1_1' and a.topic = b.topic and a.year = b.year;
```

SQL> create function map_grade (gr in char)

```

return varchar
is
  g varchar(1);
begin
  if(gr='1') then
    g:='D';
  elsif (gr='2') then
    g:='C';
  elsif (gr='3') then
    g:='B';
  elsif (gr='4') then
    g:='A';
  elsif (gr='A') then
    g:='A';
  elsif (gr='B') then
    g:='B';
  elsif (gr='C') then
    g:='C';
  elsif (gr='D') then
    g:='D';
  end if;
  return g;
end;
/

```

SQL> create or replace view student of t_student with object oid(schema,name)

```

as select ref(a)person,map_grade(b.grade)grade,ref(d)advisor,ref(c)faculty,
'local1_1'schema,a.name
from person a,local1_1.student b,faculty c,teacher d
where a.schema = 'local1_1' and a.name = b.name and
      c.schema = 'local1_1' and c.code = b.faculty.code and
      d.schema = 'local1_1' and b.advisor.name = d.person.name
union
select ref(a)person,map_grade(b.grade)grade,ref(d)advisor,ref(c)faculty,
'local1_2'schema,a.name
from person a,local1_2.student b,faculty c,teacher d
where a.schema = 'local1_2' and a.name = b.person_id.name and
      c.schema = 'local1_2' and c.name = b.faculty_name and
      d.schema = 'local1_2' and b.advisor.person_id.name = d.person.name;

```

```

SQL> create or replace view teacher of t_teacher with object oid(schema,name)
as select ref(a)person,b.title,b.rank,ref(c)project,'local1_1' schema,a.name
from person a,local1_1.teacher b,project c
where a.schema = 'local1_1' and a.name = b.name and
c.schema = 'local1_1' and c.topic = b.project.topic and
c.year = b.project.year
union
select ref(a)person,b.title,b.rank,ref(c)project,'local1_2' schema,a.name
from person a,local1_2.teacher b,project c
where a.schema = 'local1_2' and a.name = b.person_id.name and
c.schema = 'local1_2' and c.topic = b.project.topic and
c.year = b.project.year;

```

การสร้างวิวตามแบบจำลองข้อมูลรวมที่ได้จากการรวมตัวอย่างคู่ที่ 5

```

SQL> create type t_color as object

```

```

( RED          INTEGER,
  GREEN        INTEGER,
  BLUE         INTEGER
);

```

```

SQL> create type t_car as object

```

```

( COMPANY      VARCHAR2(20),
  MODEL        VARCHAR2(15),
  SPEED        NUMBER,
  COLOR        REF T_COLOR,
  GRADE        VARCHAR2(25),
  SCHEMA       VARCHAR2(15)
);

```

```

SQL> create or replace view color of t_color with object oid(red,green,blue)

```

```

as select red,green,blue from local5_2.color;

```

```

SQL> create function map_color_name

```

```

(r integer,g integer,b integer)

```

```

return varchar

```

```

is

```

```

col t_color;

```

```

col_name varchar2(25);

```

```

begin

```

```

if(r=0 and g=0 and b=0)then

```

```

col_name := 'black';

```

```

elseif(r=255 and g=255 and b=255)then
    col_name := 'white';
elseif(r=200 and g=250 and b=260)then
    col_name := 'brown';
end if;
return col_name;
end;

```

SQL> create function map_grade_car

```

    (gr varchar)
    return varchar
is
    gname varchar2(25);
begin
    if(gr='High')then
        gname := 'outstanding';
    elseif(gr='Medium')then
        gname := 'satisfactory';
    elseif(gr='Low')then
        gname := 'unsatisfactory';
    else gname := gr;
    end if;
    return gname;
end;

```

SQL> create or replace view car of t_car with object old(schema,company,model)

```

as select a.company,a.model,a.speed,ref(b)color,
    map_grade_car(a.grade)grade,'local5_2'schema
from local5_2.car a,color b
where a.color.red = b.red and a.color.green = b.green and
a.color.blue = b.blue
union
select a.company,a.model,a.speed,ref(b)color,
    map_grade_car(a.grade)grade,'local5_1'schema
from local5_1.car a,color b
where map_color_name(b.red,b.green,b.blue)=a.color

```




ภาคผนวก จ
บทความที่ตีพิมพ์จากงานวิจัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Integrating OODB Schemas Using Heuristics and Richer Semantics

Supattra Sawatrukkit¹ and Twittie Senivongse²
 Department of Computer Engineering
 Chulalongkorn University
 Bangkok, Thailand

¹E-mail: a0471521@student.chula.ac.th ²E-mail: stwittie@chula.ac.th

Abstract: The correctness of an integrated database schema, resulting from the integration of two database schemas, relies on the richness and precision of their semantic information that identifies their similarities and differences. This paper reports a progress in our work that proposes to improve the method for schema integration. In our method, heuristics that defines the level of relationships between the schemas will be used and semantics of the classes that constitutes the schemas as well as their integrity constraints will also be considered. With this additional semantic information, the integration would produce an integrated schema that is more semantically complete without generating information loss.

Key words: Schema integration, Semantics, Heuristics, Integrity constraints.

1. Introduction

Schema integration is a process for multidatabase systems that constructs a global schema by merging pre-existing heterogeneous local schemas. This allows database users to transparently retrieve data from multiple heterogeneous sources via a single read-only view [1]. The schemas that are integrated tend to be in the same context but are designed by different database designers (e.g. the schemas represent the employee databases of different branches of a company). This will result in several kinds of semantic heterogeneity such as name conflict (same information represented by different names), structure conflict (e.g. same information represented as a class attribute in one schema and as a class in the other), and scaling conflict (e.g. an attribute 'salary' with different currencies in different schemas) [2]. To integrate two schemas, the database integrator must understand the meaning of the schemas very well and can identify the similarities and differences in the semantics that underlies the schemas.

Several research works suggest ways to integrate two object-oriented schemas [3, 4, 5, 6, 7]. Primarily, the integrator is required to supply semantic information on attributes of classes in the schemas so that similarities and differences between those classes can be identified. Such information will be used to decide how the data in the two databases should be classified when the schemas are merged and how the resulting classes relate. This paper proposes that more semantic information, apart from the semantics of class attributes, can contribute to better integration. Our method additionally considers class semantic which is the meaning hidden in a particular class. Unlike other works that allow the semantics of a class attribute in a pair of (semantic name, single value) format, our method allows the semantics of a class

attribute to be listed as pairs of (semantic name, multiple values). The integrity constraint of each class attribute will also be considered. With semantic information, the relationships between classes in the two schemas will be rated using the heuristics that is adapted from that in [8], which is originally designed for canonical data models, to suit object-oriented models. Apart from equivalence and inheritance (is-a) relationships, we also consider aggregation (has-a) and sibling relationships in the integration process.

In the next section, our schema integration is described with an example of the integration of the schemas of two university databases. Note that class methods will not be considered in our process. The last section concludes the paper and discusses further work.

2. Integrating OODB schemas

The schema integration process will be described through an example of two university databases. Their schemas in Figure 1 are represented according to the grammar in Figure 2. Note that parts of the schema definition (i.e. class names, class attributes, and integrity constraints) are already available from the corresponding database but information like class semantics, attribute semantics, and a synonym list must be specified separately by the database integrator.

The added class semantics represents some hidden meaning that the database designer aims for that particular class, while attribute semantics refers to the hidden meaning of that particular attribute in the class. The semantics can be defined as (semantic name, multiple semantic values) pairs where the semantic values may refer to a range of values specified by operators =, ≠, >, <, ≥, and ≤. To specify semantics, the integrator must follow the following rules:

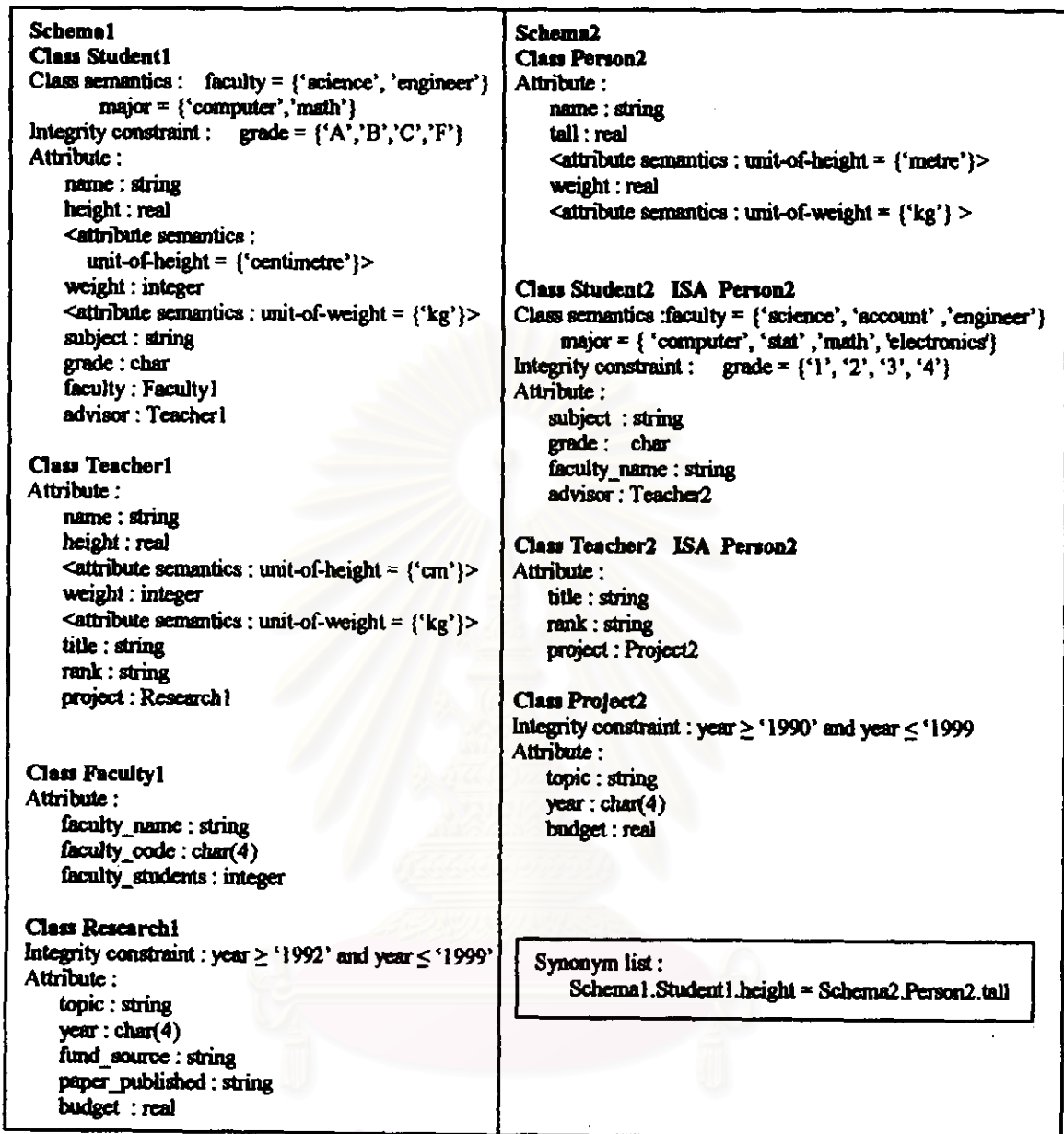


Figure 1: An Example of Schema Definitions to be Integrated

- If two things with different names refer to the same thing, they must be specified in a separate synonym list, e.g. the attribute 'height' in class 'Student1' in 'Schema1' and the attribute 'tall' in class 'Person2' in 'Schema2' from Figure 1.
- If two things with the same name refer to the same thing, their semantics (if any) will have the same semantic name but the semantic values may differ, e.g. the class semantics 'faculty' in classes 'Student1' and 'Student2' from Figure 1.
- If two things with the same name refer to different things, their semantic names must differ.

Our integration method treats integrity constraints as another kind of semantic information since they

usually define either a) range of value b) how the value of one attribute relates to that of another. In the current status of our integration process, only a) is considered. We follow the format of the integrity constraints in [9] and can cater for the constraints that are defined with the same operators used to specify semantic values. When two attributes of the two schemas relate, their integrity constraints will be considered. The constraints will tell whether some mapping is required for the representation of these attributes in the integrated schema. For example, the attributes 'grade' in classes 'Student1' and 'Student2' from Figure 1 refer to the same thing but a mapping is required to map their values to a chosen grade format (i.e. either alphabetic, numeric, or other format) when the two 'grade' attributes are merged into one in the integrated

```

<schema> ::= <schema-name> <class-list>
<schema-name> ::= <schema-name1> | <schema-name2>
<schema-name1> ::= string
<schema-name2> ::= string
<class-list> ::= <class> | <class> <class-list>
<class> ::= <classDefinition> [<attribute-list>]
<classDefinition> ::= <class-name> ['ISA' <class-name>] [<integrity-constraint>] [<class-semantic-list>]
<class-name> ::= string
<integrity-constraint> ::= <integrity-constraint> + 'AND' + <integrity-constraint> |
    <integrity-constraint> + 'OR' + <integrity-constraint> |
    'NOT' + <integrity-constraint> |
    <attribute-name> <compare-operator> + <numeric-string>
<compare-operator> ::= =, *, >, <, ≥, ≤
<numeric-string> ::= numeric | string
<class-semantic-list> ::= <meaning> | <meaning> <class-semantic-list>
<attribute-list> ::= <attribute> | <attribute> <attribute-list>
<attribute> ::= <attribute-name> <attribute-type> [<attribute-semantic-list>]
<attribute-name> ::= string
<attribute-type> ::= <baseType> | <setType> | <class-name>
<baseType> ::= 'INTEGER' | 'REAL' | 'CHAR' + '(' + <length> + ')'
<length> ::= numeric
<setType> ::= '+' + <attribute-type> + ')'
<attribute-semantic-list> ::= <attribute-semantic> | <attribute-semantic> <attribute-semantic-list>
<attribute-semantic> ::= ['Meaning' + <meaning>] |
    ['Dimension' + <dimension> + '=' <unitOfMeasure>] |
    ['Scale' + '=' + <scale>]
<meaning> ::= <meaning-name> + '=' + <meaning-value-list>
<meaning-value-list> ::= <meaning-value-list> + 'AND' + <meaning-value-list> |
    <meaning-value-list> + 'OR' + <meaning-value-list> |
    'NOT' + <meaning-value-list> |
    <compare-operator> <meaning-value>
<meaning-name> ::= string
<meaning-value> ::= string | numeric
<dimension> ::= string
<unitOfMeasure> ::= string
<scale> ::= string

```

Figure 2: BNF Grammar for a Schema Definition

schema. In fact, how the constraint of 'grade' in the integrated schema would be is not so critical because the integrated schema is read-only; the constraint of 'grade' in the integrated schema could simply be a union of the constraints from the original schemas.

2.1. Schema comparison

Schema comparison is based on *class comparison* to find out the relationships between classes in the two schemas. In class comparison, pairs of classes from the two schemas are picked one by one, and *class semantics comparison*, *attribute comparison*, and *integrity constraint comparison* are conducted to find the relationship that this pair of classes have with each other (Figure 3).

2.1.1. Semantic comparison for classes and attributes

Semantic comparison is based on some heuristics that is developed from [8]. The heuristics is a numeric value that tells the level of the relationship between two terms that are compared. Let T1 and T2 be the terms in classes C1 and C2 respectively

whose semantics are to be compared. α is the number of semantics of T1, β is the number of semantics of T2 and $\alpha \leq \beta$. Note that if T1 and T2 refer to the semantic information of the classes, we are comparing the class semantics and if T1 and T2 refer to class attributes, we are comparing attribute semantics.

Each pair of (semantic name, semantic values) will be compared and a heuristic value will be returned according to Table 1. If the semantics pair have disjoint semantic values, a flag is set that a mapping maybe required to appropriately present data values in the integrated schema. The result of the comparison of each semantics pair will be gathered in a set $R = [r_1, r_2, \dots, r_\alpha]$, and $M_s = \sum r_i$ where $r_i \in R$ and $i = 1, \dots, \alpha$ is computed. Table 2 shows the kind of relationship that T1 has with T2.

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 attributes as a whole. From the method so far, we can obtain $M_{s_{\text{class-semantic}}}$ which indicates the kind of relationship between all of their class semantics, and $M_{s_{\text{attribute-pair}}}$ which indicates the kind of

relationship between a particular pair of their attributes. To obtain $Ms_{all-attributes}$ which will indicate the relationship that C1 has with C2 with respect to all of their attributes, all 's_i' values (c.f. Table 2), each returned from the comparison of a particular pair of attributes, will further be gathered in a set $R_{all-attributes} = \{s_1, s_2, \dots, s_{\alpha_{C1}}\}$ where α_{C1} is the number of attributes in C1. Then, $Ms_{all-attributes} = \sum s_i$ where $s_i \in S, i = 1, \dots, \alpha_{C1}$ is computed. Again, $Ms_{all-attributes}$ will be checked with Table 2 for the relationship that the attributes of C1 have with those of C2. With the relationships derived from $Ms_{class-semantic}$ and $Ms_{all-attributes}$, we can obtain the

Table 2: Relationships of Terms with Semantic Consideration

Ms	Result	Relationship	W
α	Equivalence	Equivalence	1
$> \frac{1}{2}\alpha$	Equivalence ($\alpha=\beta$)	Equivalence	1
$> \frac{1}{2}\alpha$	Inclusion ($\alpha<\beta$)	Superclass	1
$> \frac{1}{2}\alpha$	Tight-intersection	Sibling	1
$< \frac{1}{2}\alpha$	Loose-intersection	Disjoint	0
$\alpha(0.1)$	Val-disjoint	Sibling	0.1
0	Disjoint	Disjoint	0

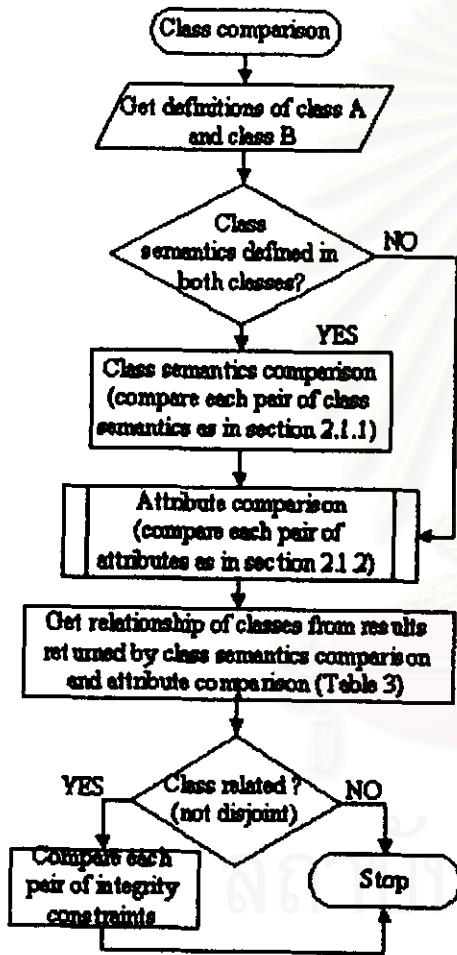


Figure 3: Class Comparison

Table 1: Result of Comparison of Each Semantics of T1 with That of T2

Semantic name	Semantic values	W
Same	Same	1
Same	Subset	1
Same	Disjoint	0.1
Same	Overlap	0.1
Different	Any	0

Table 3: Relationships of Classes

Number of class members in superclass	Number of all members in superclass	Relationship
Equal values	Equal values	Equivalence
Equal values	Superclass	Superclass
Disjoint	Sibling	None
Sibling	Superclass	Sibling

relationship that C1 has with C2 from Table 3. See Section 2.1.3 for detailed example of semantic comparison.

2.1.2. Attribute comparison

The comparison of attributes in two classes is based on the comparison of attribute semantics (as in Section 2.1.1) and the comparison of data types of the attributes (Figure 4).

As seen in the previous section, the relationship that a class has with the other class is (partially) determined by the semantics of attributes. Data types of the attributes are not taken into account in that respect but the comparison of the attribute data types will still be useful to determine if any mappings are necessary to map data values of the related attributes when presented in the integrated schema. When two attributes are related and their data types are different primitive types, one type will be, if possible, automatically coerced, by the database system that underlies the integrated schema, to the other type, e.g. integer is coerced to real. Otherwise, the integrator must define a mapping to map values of the two types. If any of the compared types is a class, it will be expanded to consider the enclosed attributes and their types. Such expansion of the class is how we take care of the aggregation relationship.

2.1.3. Example of a class comparison

Figure 5 shows the comparison of classes 'Student1' and 'Student2' from Figure 1. The comparison follows the steps in section 2.1.1 and 2.1.2. The result is that 'Student1' has class semantics and set of attributes that are equivalent to those of 'Student2', and therefore 'Student1' is equivalent to 'Student2'.

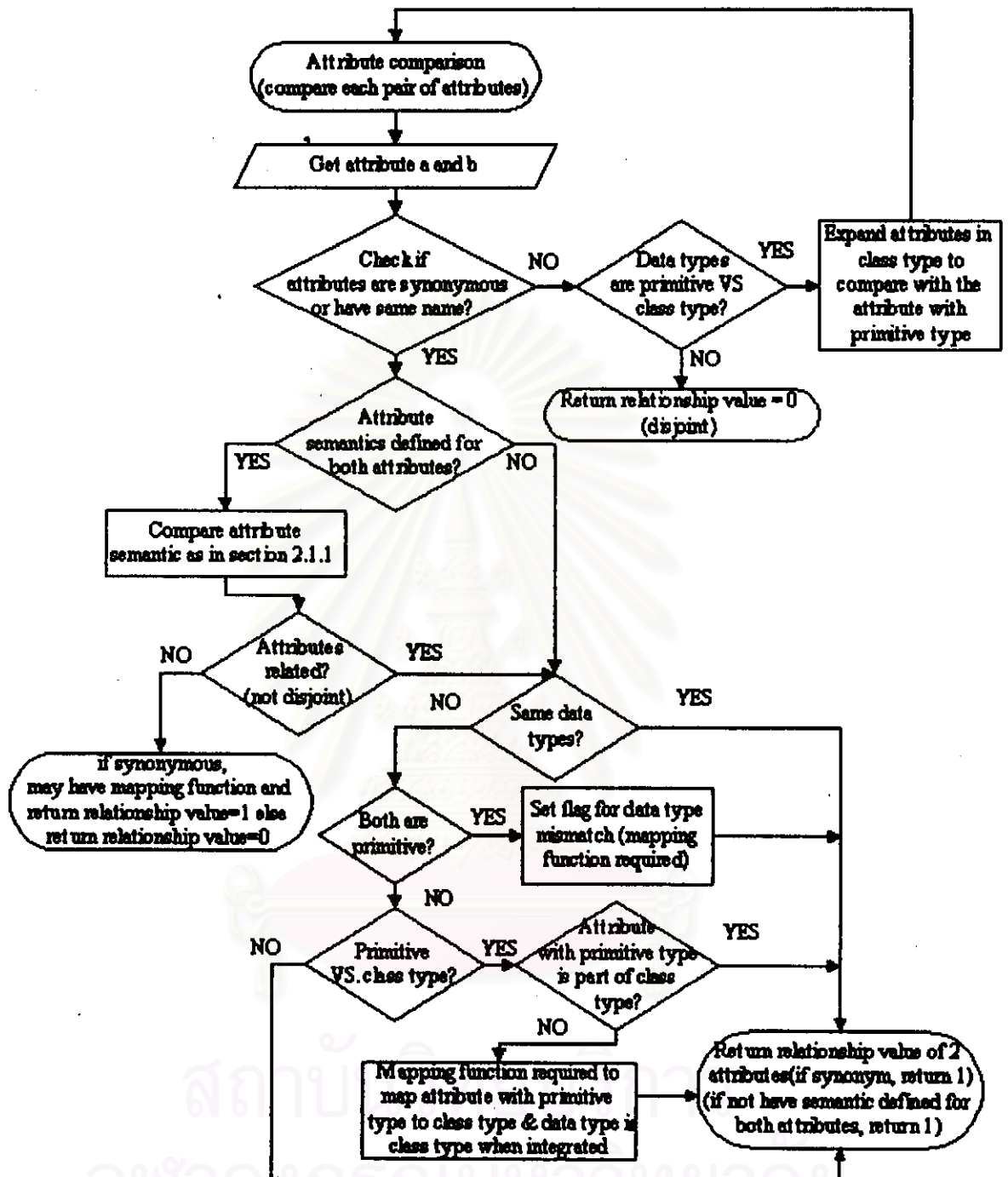


Figure 4: Attribute Comparison

2.2. Schema Integration

Schema comparison results in the relationship between each pair of the classes that are compared being identified (i.e. equivalence, superclass, or sibling). From the result, the rules below are followed to construct a new integrated schema:

- If a class has an equivalence relationship with the other, merge them into one class.
- If a class has a superclass relationship with the other, the latter has is-a relationship with the former and its common attributes are removed.

- If a class has a sibling relationship with the other, a new generalised class is introduced. The common attributes in the two classes are moved to the generalised class and the two classes have is-a relationship with the generalised class.

2.3. Linking integrated classes

From schema integration, some resulting classes in the integrated schema may be connected and some may not, e.g. classes in one schema that have no

R_{class-semantic}
 Student1.faculty is subset of Student2.faculty, Student1.major is subset of Student2.major
 $\therefore R_{class-semantic} = [1, 1] \quad M_{class-semantic} = 2 \quad \alpha = 2 \quad \text{Result} = \text{Equivalence} \quad \text{Relationship} = \text{Equivalence}$

R_{attribute-pair(name)}
 Student1.name has no semantics and the same name as Person2.name \Rightarrow return 1
 $\therefore s_{name} = 1$

R_{attribute-pair(height)}
 Student1.height has disjoint semantic value with Person2.tall while they are synonymous \Rightarrow return 1
 $\therefore R_{attribute-pair}(height) = [0, 1] \text{ <mapping required> } M_{attribute-pair} = 0, 1 \quad \alpha = 1 \quad \text{Result} = \text{Val-disjoint}$
 Relationship = Sibling $\therefore s_{height} = 1$ (synonymous)

R_{attribute-pair(weight)}
 Student1.weight has the same semantic name and semantic value with Person2.weight
 $\therefore R_{attribute-pair}(weight) = [1] \quad M_{attribute-pair} = 1 \quad \alpha = 1 \quad \text{Result} = \text{Equivalence} \quad \text{Relationship} = \text{Equivalence} \quad \therefore s_{weight} = 1$
 <mapping required for integer and real data type>

R_{attribute-pair(subject)}
 Student1.subject has no semantics and the same name as Student2.subject \Rightarrow return 1
 $\therefore s_{subject} = 1$

R_{attribute-pair(grade)}
 Student1.grade has no semantics and the same name as Student2.grade \Rightarrow return 1
 $\therefore s_{grade} = 1$

R_{attribute-pair(faculty)}
 Student1.faculty has different name from student2.faculty_name. The type Faculty1 of Student1.faculty is expanded to get the attribute faculty_name to be compared with Student2.faculty_name. Student1.faculty.faculty_name has no semantics and the same name as Student2.faculty_name. \Rightarrow return 1
 $\therefore s_{faculty} = 1$

R_{attribute-pair(advisor)}
 Student1.advisor has no semantics and the same name as Student2.advisor. \Rightarrow return 1
 $\therefore s_{advisor} = 1$

R_{all-attributes} = [1, 1, 1, 1, 1, 1, 1] $M_{all-attributes} = 7 \quad \alpha_{C1} = 7 \quad \text{Result} = \text{Equivalence} \quad \text{Relationship} = \text{Equivalence}$

\therefore From Table 3, Relationship_{class-semantic} + Relationship_{all-attributes} = Equivalence + Equivalence = Equivalence

Figure 5: Comparison of Classes Student1 with Student2

Table 4: Result of the Comparison of Classes from Figure 1

Schema2 \ Schema1	Student1	Teacher1	Faculty1	Research1
Person2	Superclass	Superclass	Disjoint	Disjoint
Student2	Equivalence	Sibling	Disjoint	Disjoint
Teacher2	Sibling	Equivalence	Disjoint	Disjoint
Project2	Disjoint	Disjoint	Disjoint	Superclass

relationship with any classes in the other schema. The final step is to link these classes into the integrated schema using is-a or has-a relationships that these classes have with other classes in its original schema.

2.4. Example of integrated schema

Pairs of classes in the two schemas in Figure 1 are compared in a similar manner to 'Student1' and

'Student2' in Figure 5. Table 4 shows the results of the comparison of the pairs of the classes and Figure 6 presents the final integrated schema of our university example. Note that a mapping is required to map the height values of the data retrieved from the two databases to a chosen unit of measurement for the class 'Person' the integrated schema. Also, the attribute 'grade' in the merged class 'Student' in the integrated schema will be associated with a

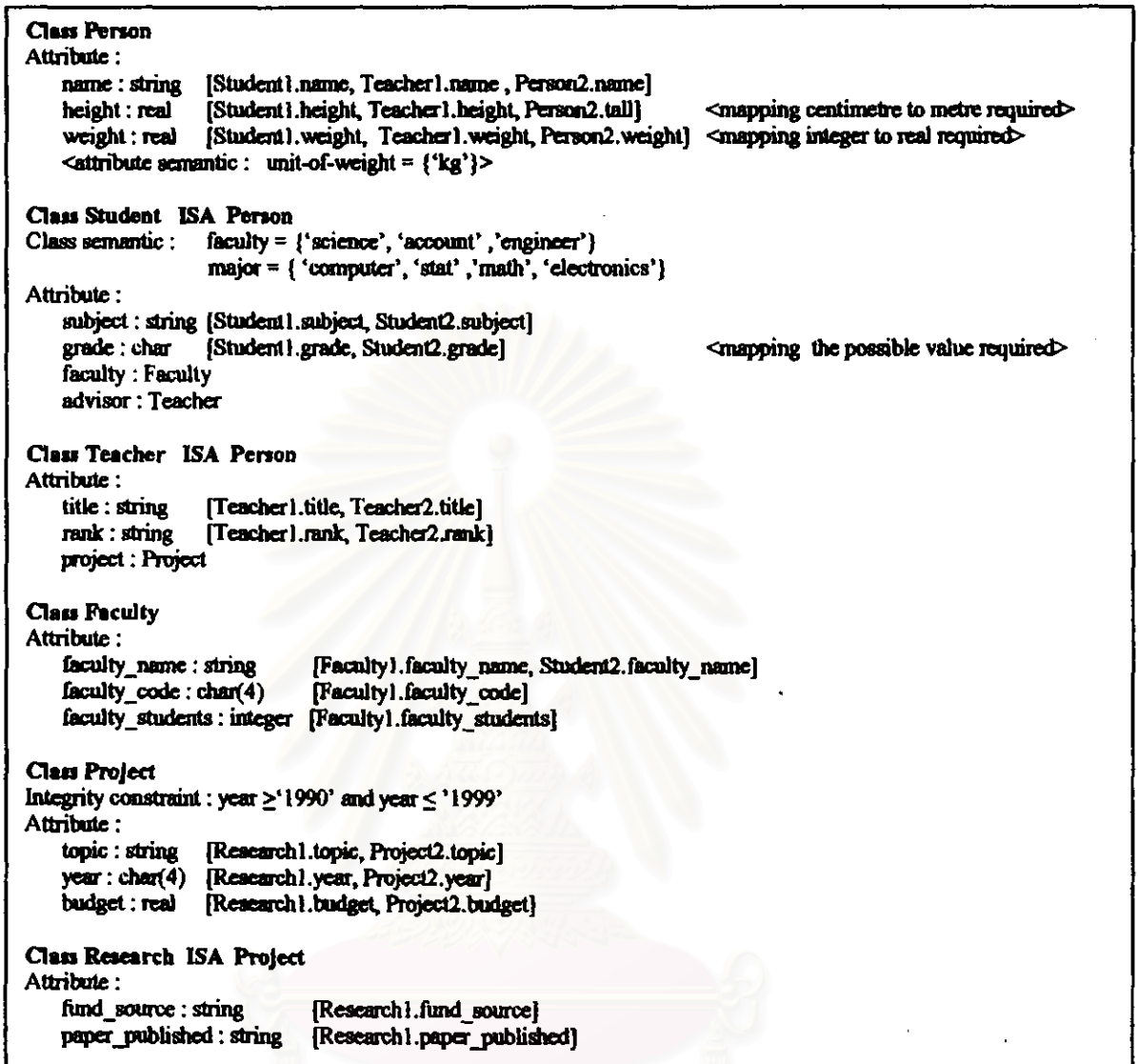


Figure 6: Integrated Schema of University Databases Example

mapping that maps the grade values of the conflicting integrity constraints to a constraint of choice.

3. Conclusion

More semantic information that are considered in our schema integration method, i.e. class semantics, integrity constraints, and richer semantic definitions, will contribute to better and more accurate integration of the schemas. However, the work is still in progress to define a more complete table of relationships of classes (c.f. Table 3) based on more various combination of class semantics relationships and all-attributes relationships. A prototype of the integration method is being constructed to test that the information retrieved from a resulting integrated schema is the same as that separately retrieved from the original schemas, i.e. the integrated schema does not incur

information loss. We also plan to study if our method produces good integrated schemas according to the software matrices used to measure and evaluate object-oriented models.

Acknowledgement

The authors would like to thank the National Science and Technology Development Agency for its partial financial support.

References

- [1] Bukhres, O. A. and Elmagarmid, A. K. Object-Oriented Multidatabase Systems (A Solution for Advanced Applications), *Prentice-Hall*, 1996.
- [2] Kim, W. (ed.). *Modern Database Systems, The Object Model, Interoperability, and Beyond*, *ACM Press, Addison-Wesley*, 1995.

- [3] Sull, W. and Kashyap, R. L. A Self-Organizing Knowledge Representation Scheme for Extensible Heterogeneous Information Environment, *IEEE Transactions on Knowledge and Data Engineering*, volume(4), no(2): pp 185-191, April 1992.
- [4] Geller, J.; Mehta, A.; Perl, Y.; Neuhold, E. and Sheth, A. Algorithms for Structural Schema Integration, *In the Proceedings of ICSI' 92*, pp 604-614, 1992.
- [5] Thieme, C. and Siebes, A. Schema Refinement and Schema Integration in Object-Oriented Databases, *Technical report CS-R9354, CWI, Amsterdam, The Netherlands*, 1993.
- [6] Zhang, X.; Liu, J. and Li, H. Object-Oriented Integration of Heterogeneous Distributed Databases, *In the Proceedings of IEEE Regional Conf. on Computer, Communication, Control, and Power Engineering (TENCON' 93)*, volume(1): pp 331-334, 1993.
- [7] Schmitt, I. and Sake, G. Merging Inheritance Hierarchies for Database Integration, *In the Proceedings of the 3rd IFCS Intl. Conf. on Cooperative Information Systems*, pp. 322-331, 1998.
- [8] Larab, O.; Benharkat, A. N. and Wagner, R. R. Description Logics and Correspondence Refinement Process in Databases Interoperability's Service, *In the Proceedings of the 8th Intl. Workshop on Database and Expert Systems Applications*, pp 700-705, 1997.
- [9] Beneventano, D.; Bergamaschi, S.; Lodi, S. and Sartori, C. Consistency Checking in Complex Object Database Schemata with Integrity Constraints, *IEEE Transactions on Knowledge and Data Engineering*, volume (10), no(4): pp 576-598, July-August 1998.



ประวัติผู้วิจัย

นางสาวสุพัตรา สวัสดิ์รักเกียรติ เกิดเมื่อวันที่ 16 มกราคม พ.ศ.2519 ที่อำเภอบางกอกใหญ่ จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาตรีวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2539 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ.2540 และเคยได้รับทุนภูมิพล เมื่อ พ.ศ.2536



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย