CHAPTER 7

THE MODELING OF THE ONTOLOGY-BASED METADATA DICTIONARY: A CASE STUDY

A case study will demonstrate how the proposed metadata dictionary solves the semantic heterogeneity. In this chapter, the ontology modeling technique discussed in Chapter 4 can be applied to the design of ontology-based metadata dictionary through a practical case study. This ensures that the proposed approach can be applied to a practical case to validate the viability of model realization. A case study illustrates an example of semantic heterogeneity occurring in most organizations. The example in a case study is also used as a basis for illustrating the query processing the HIS on the WWW in the next chapter.

7.1 An Example of the Semantic Heterogeneity

To illustrate how the proposed metadata dictionary solves the semantic heterogeneity, this section demonstrates the three different physical information sources in a university referred to as a domain of discourse. The overall hierarchical concepts is shown in Figure 7.1. This example will consider four main concepts, namely, `Staff`, `Administrator`, `Instructor`, and `Official`. All members of the categories `Administrator`, `Instructor`, or `Official` are contained in the concept `Staff`, denoted by ($\forall a$ in `Administrator` $\wedge$ $\forall i$ in `Instructor` $\wedge \forall o$ in `Official`) $\in$ `Staff`. Hence, the concept `Staff` is a superconcept and the concepts `Administrator`, `Instructor`, and `Official` are subconcepts of concept `Staff`. However, the partial IS-A relationship between `Administrator` and `Instructor` occurs when some administrators (not all) are instructors and some instructors are administrators, denoted by ($\exists i$ in `Instructor` $\in$ `Administrator`) $\wedge$ ($\exists a$ in `Administrator` $\in$ `Instructor`). Meanwhile, the concept `Administrator` and `Instructor` are independent of concept `Official` since neither administrators nor instructors are officials, denoted by ($\forall a$ in `Administrator` $\wedge$ $\forall i$ in `Instructor`) $\notin$ `Official`.

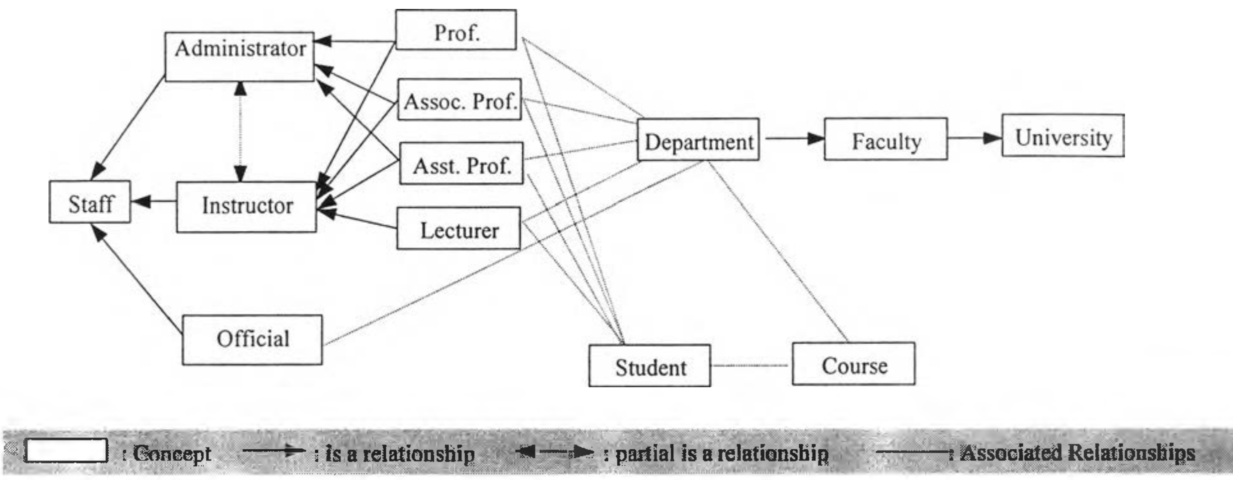


Figure 7.1 An overview of the hierarchical concepts of a university system.

The three different physical information sources are shown in Figures 7.2 (a), (b) and (c). The first information source `Source1`, shown in Figure 7.2 (a), consisting of the concepts `Staff_Member` and `Department` that are represented by relational data model. The second information source `Source2`, shown in Figure 7.2 (b), consisting of the concepts `Instructor_Member` and `Course`, as well as the relationship, `Course_Teach`, which links between `Instructor_Member` and `Course`. The concepts and relationships in `Source2` are represented by XML-based data model. The third information source `Source3`, shown in Figure 7.2 (c), consists of the concepts `Admin_Member` and `Department` that are represented by object-oriented data model.

The physical source examples not only illustrate the differences in data models and query formulation, but also semantic heterogeneity, which results in three types of conflicts. First, naming conflicts occur as the property `Staff_id` in the relation `Staff_Member` of `Source1`, `Inst_id` in the element `Instructor_Member` of `Source2`, and `Adm_id` in the class `Admin_Member` of `Source3` are semantically equivalent properties, since they all refer to the same fact. This is known as a synonym conflict. Second, data type and scaling conflicts are caused by the same property `Salary` of `Staff_Member` and `Instrutor_Member` having different predefined types and units of measure. Finally, generalization conflicts induce from the derivation of the concept `Staff` subsuming the concepts `Administrator` and `Instructor`. These examples will serve as the basis for the ontology-based metadata dictionary design in the sections that follow.

Staff_Member

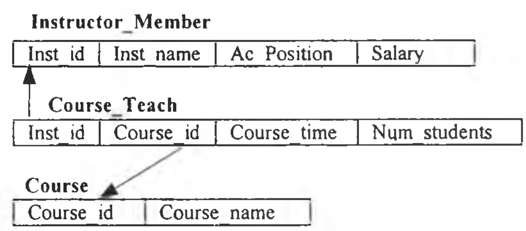
PhyProperty	PhyDataType	PhyUnitType
Staff_id	String	NULL
Staff_name	String	NULL
Salary	Float	US$
Dept_id	Integer	NULL

Department

PhyProperty	PhyDataType	PhyUnitType
Dept_id	Integer	NULL
Dept_name	String	NULL

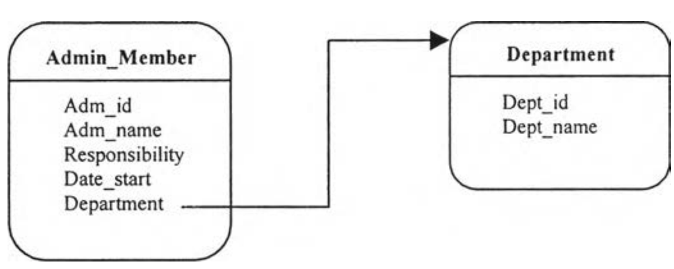
(a) Physical information sources at Source1 represented by relational data model.

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE Instructor_Course [
<!ELEMENT Instructor_Course        (InstructorMember_List,
Course_List)>
<!ELEMENT InstructorMember_List (Instructor_Member)+>
<!ELEMENT Instructor_Member (Inst_name, Ac_Position, Salary,
Course_Teach)>
<!ATTLIST Instructor_Member Inst_id  ID  #REQUIRED>
<!ELEMENT Inst_name  (#PCDATA)>
<!ELEMENT Ac_Position  (#PCDATA)>
<!ELEMENT Salary  (#PCDATA)>
<!ELEMENT Course_Teach (Crs)+>
<!ELEMENT Crs (Course_time, Num_students)>
<!ATTLIST Crs  Course_id   IDREF #REQUIRED>
<!ELEMENT Course_time (#PCDATA)>
<!ELEMENT Num_students (#PCDATA)>
<!ELEMENT Course_List (Course)+>
<!ELEMENT Course (Course_name)>
<!ATTLIST Course Course_id ID #REQUIRED>
<!ELEMENT Course_name (#PCDATA)>
]>
```



Relational schemas describing the relationships between concepts of XML-DTD in Figure 5 (b).

(b) Physical information source at Source2 represented by XML-DTD.



(c) Physical information source at Source3 represented by object-oriented data model.

Figure 7.2 Three different data models of physical information sources.

7.2 Domain Ontology Representation

7.2.1 The conceptual level representation

The conceptual level of the ontology has been designed to solve data type, unit type, and generalization conflicts. The design is based on the proposed modeling technique outlined in Chapter 4 and illustrated by the EER model in Figure 7.3. Each virtual concept possesses its own virtual properties, for example, `Staff(st_id, st_name, st_salary, dept_id)`. The virtual property `st_id` is an object identifier or key, while `st_name`, and `st_salary` are ordinary properties, and `dept_id` is an object identifier reference or foreign key. The virtual concept `Staff` relates to `Department` by an associative relationship. To solve data type and unit type conflicts, the object identifier and ordinary properties can further designate additional domain properties to specify predefined type and scaling domain. For example, the domain properties of `st_salary` are of the predefined type "Float" and scaling domain "US$." To solve generalization conflicts between the concepts `Staff` and `Instructor`, `Instructor` is designed to associate with `Staff` by an IS-A relationship, since `Instructor` is a subconcept of `Staff`. As such, `Instructor` inherits `st_id`, `st_name`, `st_salary`, and `dept_id` from `Staff`. Consequently, `Instructor` also associates with `Department` by an N:1 associative relationship.

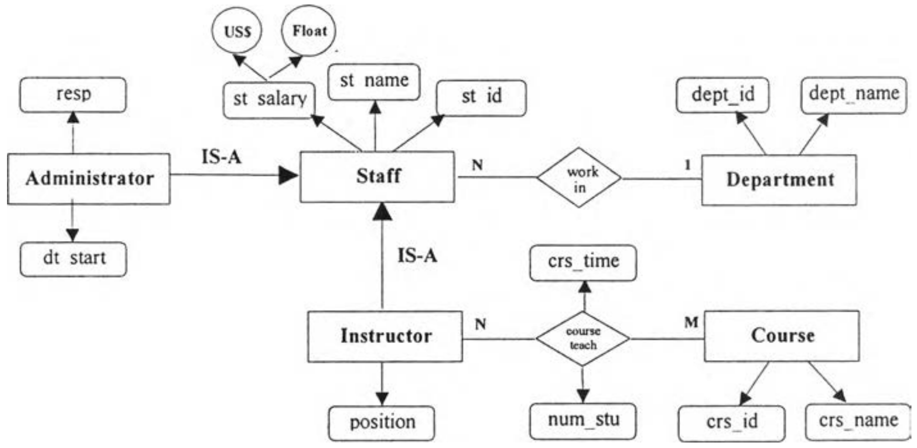


Figure 7.3 The logical ontology structure at the conceptual level of abstraction.

As mentioned earlier, the relationship `course_teach` defines its own properties `crs_time` and `num_stu` in addition to those of the participating concepts `Instructor` and `Course`. Hence, the relationship `course_teach` is treated as a concept in the ontology.

7.2.2 The physical level representation

The physical level of the ontology illustrated in Figure 7.4 is designed to solve the naming conflicts. Since synonym conflicts of the physical property `Staff_id` of `Staff_Member`, `Inst_id` of `Instructor_Member`, and `Adm_id` of `Admin_Member` are common encounters in the HIS environment, synonym terms could be designed as the physical instances of the virtual property `st_id` through the instantiate relationships. Each physical instance, `Staff_id` for example, is the physical property name which defines its physical information properties for storing additional physical information associated with `Staff_id`. For example, the values of physical information properties named `PDataType`, `PUnitType`, `PCname`, and `PSname` of `Staff_id` are "integer", "NULL", "Staff_Member", and "Source1", respectively. This means that `Staff_id` is a physical property name having the physical data and unit types, "integer", and "NULL", and the physical concept name "Staff_Member" reside in "Source1."

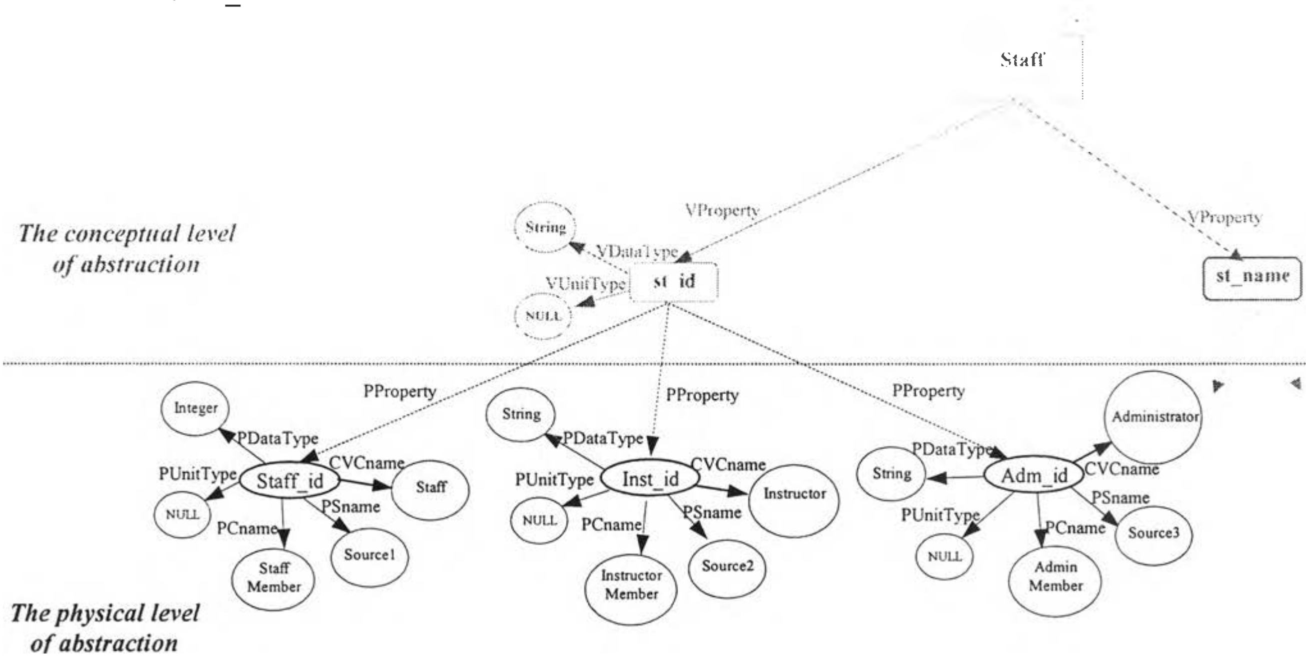


Figure 7.4 A portion of internal structure of the ontology at the physical level of abstraction.

7.3 The XML-based Metadata Dictionary Representation

The logical ontology structures based on existing entities in Figures 7.3 and 7.4 are translated into XML-based metadata dictionary consisting of XML-DTD (as shown in Figure 6.3) and XML document. A partial XML document structure storing well-formed and valid data is given in Figure 7.5.

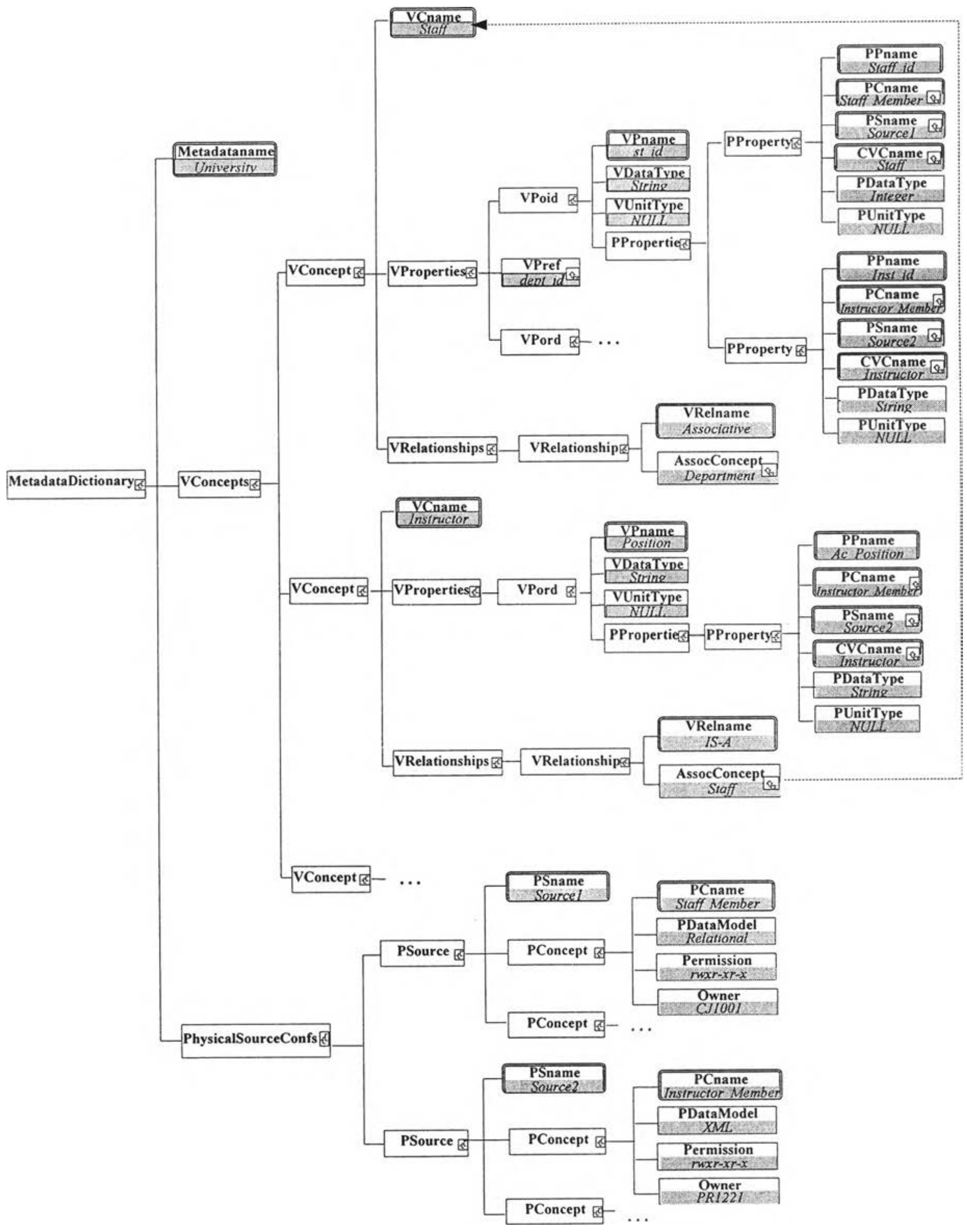
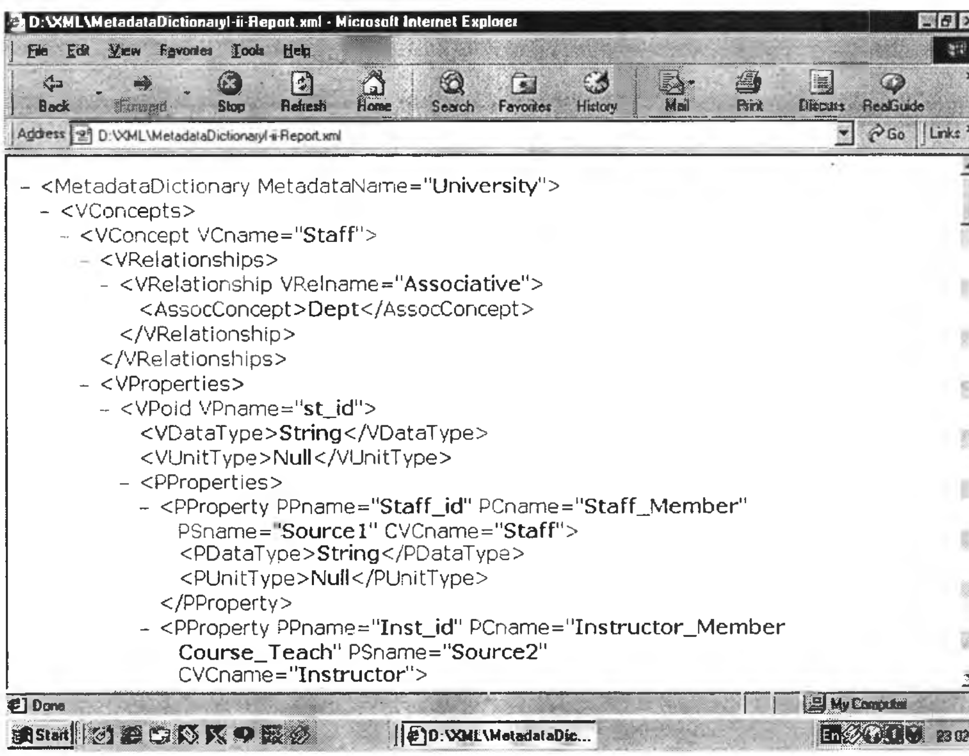


Figure 7.5 A portion of the XML document structure conforming to earlier XML-DTD.

In Figure 7.5, the IS-A relationship is transformed to a derived `VConcept` of its based `VConcept Staff` for relationship preserving and object derivation conformance. As such, the `VConcept Instructor` needs only define its own specialized properties, whereby all base relationships and properties are automatically inherited from its base class `Staff`. This fact reaffirms the proposed metadata dictionary principles of object orientation. The resulting XML document from Figure 7.5 is shown in Figure 7.6.



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

Figure 7.6 A portion of the XML document based on metadata dictionary.

Hence, the formality so introduced will enhance the formulation and design of more sophisticated metadata ontology-based components, in particular, rigorous verification that leads to correctness and reliable operations on the HIS.