Vrije Universiteit Brussel
Faculty of Computer Science
Department of Computer Science
Academic Year 2001-2002

Business Rules & WSDM Methodology of Web Design

Jehad Najjar

Promoter: Prof. Dr. Olga De Troyer

Dissertation Submitted in View
Of obtaining of a degree of
Master of Science in Computer Science

TO

My parents

My wife and my two Kids

All my Professors

All my friends

Abstract

There are various web design methodologies and strategies that use different mechanisms and algorithms. However, none of them deals with integrating business rules during the website development process. Business rules, which have been around since ancient times, covers many human needs and their daily life business, and therefore, should be considered in parallel with other design issues.

The aim of this study is to investigate the theoretical approach of integrating business rules into the web site development process. This approach is an extension of the Web Site Design Method (WSDM), which is developed by the Web & Information Systems Engineering research group (WISE) at Vrije Universiteit Brussel.

The methodology goes in parallel throughout the phases (of the web site development process life cycle) defined in WSDM methodology. In the first phase business policies of the enterprise are defined. The second phase addresses and documents the business rules and matches them with the different audiences. In the third phase business rules are modeled conceptually using ORM and RIDL. Then the resulted model is integrated into business information model of WSDM where the chunks and audiences are defined.

Keywords: Business Rules, WSDM, Conceptual Modeling

Acknowledgment

First of all, I would like to thank **Prof. Dr. Olga De Troyer**, for the valuable time she spends for me during the year, advising and correcting my thesis. It was a nice experience to do a study with her group.

Secondly, I want to thank **Sven Casteleyn**, for his valuable advice during the year of my study at the lab, also I want to thank all nice members of **WISE** group who helped me in several issues.

Also great thanks go to my **wife,** for her support and patience, to my parents, for their support and encouragement.

Finally, I would like to thank all my friends who advice and encourage me during this study, so I thank all Palestinian students in Belgium, specially Mustafa Jarrar and Jamal Said for their advices and encouragement.

I also would like to acknowledge Al-Quads University for funding my master degree.

Table of contents

List of figures

Structure

This thesis consists of two parts: the background part and the research part.

Background

The first part of the thesis provides the information that is needed to understand the research part. Firstly, we will discuss Business Rules in e-business web sites. Next, the most important existing web site design methods are described, and then we discussed how some of these methodologies covered some small aspects of business rules. Finally, WSDM methodology is discussed in more details.

Research

The second half of the thesis, the research part, describes the research that has been done. First we discuss the need for defining and document the enterprise policies and rules and how to assign business rules to audience classes, second discussed conceptual modeling techniques and finally we represent these rules at the business information model of the website. According to each of the audience classes functional and information chunks.

Chapter One

Overview

There has been a tremendous increase in the usage of the Internet to serve our daily needs and operations. For instance, buying or renting a car, reserving a room in a hotel, booking a flight ticket, etc. These kinds of businesses carry some rules and constraints. Therefore when a web designer develops an enterprise application he has to take into consideration these rule and polices. Furthermore he has also to satisfy the customer needs (e.g. usability, simplicity, understandability, reach ability, efficiency, etc).

Achieving the needs of the customer and satisfying the enterprise policies and constraints, requires the developer to be skillful, experienced. However experience and skills differs from one developer to another. Consequently we need a methodology that could bring insight to the experienced and support the novice ones. Furthermore this methodology should support the business polices and constrains as well as the audience needs to be integrated during the development process. Our goal is to develop a strategy that will satisfy the needs of the customer and cover all the business policies and operations. So business rules must be discovered, documented and modeled from the beginning phases of the web site development process besides the requirements of the different audiences. Contrary to the existing methodologies as [7,27], which focus on either data available at the organization or the structure of the Organization, WSDM [15-18] is audience-oriented methodology. It is a powerful methodology; it starts from the requirements of the audience, document, model and implement them.

However, WSDM does not consider explicitly the business rules and constraints while developing the web site. The method could be improved if the business rules were integrated through out the life cycle of the web development. This approach can easily be performed due to the fact that WSDM approach distinguishes between conceptual and logical design of the website. Therefore, business rules will be addressed at the early phases of the web site development life cycle. In other words, we will not leave these business rules for the implementation phase. This approach increases the tractability of the model during development because during the development process the rules evolves from phase to the other in a kind of transformational way, which keeps links between the previous phase and the next one. This approach can cope with the changes in the specifications during development. Furthermore, it increases the maintainability and the efficiency of the website. Which will lead to a successful e-business.

1.1 What are Business Rules?

Integrating the business rules at the early stages of e-business sites development pledges to be the most practical and desirable way that leads to successful e-business. When the web design developer is aware of the given constraints of the business from the beginning it would be easier for him to develop the website that satisfy both audiences and business. However, what are business rules? There have been various definitions of business rules:

A business rule is a statement that defines or constraints some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business [2].

Or,

Business rules describe and control the structure, operation and strategy of an organization [3].

Business rules have been around since ancient times. Egyptian wall paintings show examples of business transacted according to well-understood rules. At the enterprise usually Business Rules are formulated in policy and procedure manuals, supplier agreements, customer contracts, marketing strategies, and as expertise embodied in employees. They are dynamic and likely to be changed with time, found in most types of application as discussed in the next section.

1.2 Categories of Business Rules

A Business Rule is classified into four categories as defined in [2]:

- ***Definition of business terms***

The most basic element of a business rule is the language used to express it. The simplest definition of a term is itself a business rule, which describes how people think and talk about things. Thus, defining a term is establishing a category of business rule.

Terms have been traditionally documented in glossaries or as entities in an entity/ relationship model.

E.g., "Car model", " customer", "general motors", "5000"

- ***Facts relating terms to each other***

The nature or operating structure of an organization can be described in terms of the facts that relate terms to each other. To say a customer can place an order is a business rule. Facts can be documented as natural language sentences or as relationships, attributes, and generalization structures in graphical model.

E.g., " Name is an attribute of customer."

- ***Constraints ("action assertion")***

Every enterprise contains behavior in some way, and this is closely related to constraints on how data may and may not be updated. To prevent a record from being made is, in many cases, to prevent an action from taking place.

E.g., "If a car customer age less than ¨25¨ years then add 20% to the rental charge "

- ***Derivations***

Business rules (including laws of nature) define how knowledge in one form may be transformed into other knowledge, possibly in deferent form.

E.g.: "the rental amount, depends on the customer category, and initial rent of the specific car.¨

1.3 Why business rules for web development

Often, at any enterprise there are one or two key people who can tell you the criteria for making daily business decisions. Having the enterprise and business decisions been dependent on one or two people might risk the business because they might leave the company at any time. This could lead to a huge loss of the knowledge they have, and leave you with only the source code of the web site that is not readable by you. However, if the business rules are documented and arranged, this would prevents this loss and also makes the web site more flexible and adaptable to future changes. The illustration below describes three employees at a business who have conflicting ideas about some decisions taken in a day-to-day selling operation, because the company have different policies in dealing with different customers (e.g., Golden and Silver customers). However if this policy was documented and in-lined as a business rule this conflict between employees could be avoided. Persons involved in the chat are:
Jian: financial department, Adam: Production Department, Christine: Sales department

Jian, This morning we received a Sales Report. Total payment for customer X was 800$. How was this amount determined? *Adam*
Adam, Since customer X is a golden customer, we gave a discount of 20%. He become a golden customer before 20.12.2001, so his type changed from silver to gold customer, Hope you don’t have a problem with this (?). *Jian*
Jan, I found a mistake at the financial statement, when a customer changes his category. *Adam*
Adam, I think we can do nothing now. It’s too late. *Christine*

From the above scenario we observe the importance of documenting business rules at the early stages of the any web site development process.

1.4 Advantages of using business rules approach in building E-Business websites

Using business rules as Meta Data have many advantages and benefits. They:

- Allow maximum flexibility

  By addressing and implementing business rules as Meta data elements, an enterprise can easily make changes to rules as policies, guidelines, strategies, and environments change. Code chunks do not need to be changed. Only the content of the “business rule tables” is changed.

- Reduce system maintenance

No need to change code every time a business rule changes will obviously reduce system maintenance. Instead of having to re-design, re-code, re-compile, re-integrate, and re-implement components, business rule changes are a simple process. Furthermore, can be change without affecting Implementation When business rules are modeled and implemented as Meta data, changes in the rules have little or no impact on site development, and implementation. Thus, Site built with this approach can truly reflect the most current business requirements and the requirements can change even during the final stages of development.

1.5 Web Design Methodologies and Business Rules

There exist different types of web design methodologies each of them has a specific approach that differ from the others.

1.5.1 Current and old Web Design methodologies

1.5.1.1 ARANEUS

ARANEUS, as discussed in [9], is a project developed by researchers of Università di Roma Tre. It aims at defining an environment for managing unstructured and structured Web content in an integrated system, called Web Base Management System (WBMS). Araneus adopts a mix of database and hypermedia concepts, both in the modeling notations and in the development lifecycle. The structure of the application domain is described by means of the Entity Relationship Model, whereas the navigation aspects of the application are specified using the Navigation Conceptual Model (NCM). Conceptual modeling is followed by logical design, using the relational model for the structural part, and the Araneus Data Model (ADM) for the navigation aspects. ADM is based on the notion of page scheme, a language-independent page description notation based on such elements as attributes, lists, link anchors, and forms. Development proceeds according to a structured process organized along two tracks: database and hypertext. Database design and implementation are conducted using the Entity-Relationship Model and mapping it into relational structures. Hypertext design consists of conceptual modeling, which formalizes navigation by turning the ER schema into a NCM schema, and of hypertext logical design, which maps the NCM schema into ADM page-schemes. Finally, implementation requires writing page-schemes as declarations in the Penelope language, which specifies how physical pages are constructed from logical page schemes and content stored in a database.

1.5.1.2 STRUDEL

STRUDEL, as discussed in [13], is a project of AT&T Labs which aims at experimenting a novel way of developing Web sites based on the declarative specification of the site's structure and content. In Strudel both the schema and the content of a site are described by means of a set of queries over a data model for semi-structured information. Content is represented using the Uniform Graph Model, a graph-based data model capable of describing objects with partial or missing schema. The design of a Web site is done in a declarative way, by writing one or more queries over the internal representation of data, using the Strudel query language (StruQL). Such queries identify the data to be included in the site, and the links and collections of objects to be provided for navigation. In this way, Strudel separates the description of content from the definition of the structure and navigation of the site. Presentation is added as a separate dimension by means of HTML templates, which specify the rendering of the site definition queries in HTML.

1.5.1.3 Hypertext schema design (HDM)

HDM, as discussed in [7], is one of the pioneering works in hypermedia and hypertext modeling. It introduced a notion of model-based design, clearly separating the activities of authoring in-the-large (hypertext schema design) and authoring in-the-small (page and content production).

HDM can be used to model applications, giving the designers and users a better idea of what has been specified. With the provision of appropriate development tools, HDM specifications can also be used to implement applications. Indeed, this ability to simultaneously describe, document, and prototype applications is one of the strengths of such hypertext-model-based efforts. The method is not used anymore but its successor, OOHDM is very popular.

1.5.1.4 W2000

W2000 as discussed in [11], is the last evolution from HDM. It consists of integration between an extended and customized version of UML and some concepts of the traditional HDM model.

1.5.1.5 Web Architect

Web Architect as discussed in [10], is a project aimed at developing methods and tools for the construction of Web Based Information Systems (WBIS). The authors propose a structured design process that goes through the analysis, design, construction and maintenance of a Web site. Analysis includes both static and dynamic modeling; the former is conducted with the Entity Relationship Model, the latter requires the identification of scenarios, in the tradition of object-oriented modeling. During ER modeling, entities are classified according to the different role they play in the definition of the site (agent, product, or event). Design is conducted in parallel to scenario analysis and aims at pinning down the structure and navigation schema of the Web site. Design results are represented using a variant of the Relation Management Data Model by Isakowitz, which incorporates the roles of the entities forming the Web site.

1.5.1.6 Object-Oriented Hypermedia Design Method (OOHDM)

OOHDM as discussed in [5], is an object-oriented evolution of HDM, which uses abstraction, and composition mechanisms in an object oriented framework to describe complex information items and specify navigation patterns and interface transformations. In OOHDM, a hypermedia application is built in a four-step process supporting an incremental or prototype process model. Each step focuses on a particular design concern, and an object-oriented model is built. Classification, aggregation and generalization/specialization are used throughout the process to enhance abstraction and reuse.

In the next paragraphs, we describe each step in detail.

Conceptual Design

In this phase a model of the application domain is built using well-known object-oriented modeling principles with a notation similar to UML. The model consists of classes, subsystems and relationships, which are built using aggregation and generalization/specialization hierarchies. The most important in this phase is that the domain semantics must be taken as neutral as possible; users and tasks are not yet to be considered.

OOHDM doesn't demand to use a particular method to produce the conceptual class schema; any of the well-known methodologies (OMT, UML) may be employed.

Navigational Design

OOHDM sees an application as a navigational view over the conceptual model. During navigational design, user profiles and tasks are taking into account. With one conceptual schema, different navigational models are possible; each navigational model corresponds with another view or application on the same domain.

Abstract Interface Design

Once we have a navigational design, an abstract interface model is built. During abstract interface design, the interface objects which the user will perceive are defined: how will the navigational objects look like, which interface objects will activate navigation, etc.

The clear distinction between navigational and abstract interface design leads to a higher degree of independence from user-interface technology: different interfaces may be built for the same navigational model.

Implementation

During implementation, the navigational and abstract interface models are transformed to concrete objects, available in the chosen implementation environment. The uniform modeling constructs, objects and classes, used in the OOHDM method, allow a fluent transition of the domain modeling to navigational and interface design.
It is not necessary to use object-oriented mechanisms in the implementation-phase, but it will make it easier. Anyway, there are lots of techniques to map an object-oriented specification on a non object-oriented runtime environment.

1.5.1.7 TIRAMIS

TIRAMIS as discussed in [4], is an ongoing project at the University of Washington. It defines a simple interface to which external implementation tools can comply to create web content. Tiramis also allows a web master to graphically view the web site as a graph of web content connected together by hypertext and inclusion links. Although it is built enough to be able to create complete web sites, it would still be considered alpha software in a commercial environment.

1.5.1.8 PIPE

PIPE as discussed in [23], aims at modeling personalization in a flexible way, using the programmatic notion of partial evaluation. Partial evaluation is a technique used to automatically specialize programs, given incomplete information about their input.

1.5.1.9 Web Modeling Language (WebML)

WebML, as discussed in [22], provides a model-driven approach for specifying websites on conceptual level, with both a graphical notation and a textual XML-syntax [31].

The specification of a site in WebML consists of four perspectives:

- The structural model: concerning data content
- The hypertext model: concerning the pages and the links between pages
- The presentation model: concerning the layout and graphical requirements
- The personalization model: concerning the customization features for one-to-one content delivery

These models are explained in the next section.

Structural model

This model describes the data content of the website, in terms of relevant entities and relationships. Classical notations such as the E/R model [28], the ODMG object-oriented model [19] and UML class diagrams [20] can be used for the data modeling.
Calculated information can be expressed using a simplified OQL-like query language.

Hypertext model

The hypertext model formulates the hypertexts that correspond with the different site views. Site view descriptions in turn have two sub models: the composition model and the navigation model. The composition model contains the pages of the hypertext and the content units of the pages, while the navigation model expresses the linking between pages and content units.

Presentation model

In this phase, the layout and graphical look of pages are formulated by means of an abstract XML syntax.

Personalization model

The personalization model gives a model of the users and user groups, specified in terms of predefined entities called User and Group. The properties of these entities permit for saving group-specific or individual content (e.g. shopping suggestions or list of favorites).
Then, OQL-like declarative expressions can be added to the structure schema, to define derived content, based on the profile data in the User and Group entities. With this personalized content, units can be composed and presentation specifications can be defined.
With WebML, high-level business rules, written in simple XML syntax, can also be defined to handle site-related events like user clicks and content updates. This way, new user-related info (e.g. shopping histories) is formed and site content is updated (e.g. new offers matching users' preferences).

1.5.1.10 Relationship Management Methodology (RMM)

RMM as discussed in [27], is another ancestor of current research projects. It proposed a hypermedia modeling language built upon the Entity-Relationship model, supported by a seven-steps hypermedia design lifecycle in the tradition of software engineering. RMM also included guidelines for the typical hypermedia design tasks.

1.5.1.11 World Wide Web Design Technique (W3DT)

W3DT as discussed in [12], was invented especially for the development of large Web sites. The W3DT methodology is capable of modeling highly structured, database-like information-domains on the one hand and conventional hypertext on the other and thus represents the dichotomy of structured as well as unstructured information. A prototypical tool called Web-Designer supports the graphical notation of the methodology and is capable of generating a framework of HTML-pages and CGI-Scripts at every step of the design process.
W3DT provides the necessary modeling constructs to fill this gap. The notation is intentionally kept as simple as possible to make W3DT easily comprehensible for the novice user.

1.5.2 Conclusion

Some of the web design methodologies covers in simple way some business rules at their data model at model-driven methodologies. For example (W2000, OOHDM, WebML) methodologies are using UML [20] to model the content of the sites and implement it afterwards without documenting the business rules and don't conceding integrating them during the development process. Other methodologies, like (Web Architect, ARANEUS, RMM, W3DT) using Entity Relationship [28] to model the content of the web site. In other words, they integrate business rules at the implementation level, which is not a successful approach, because the business policies might change during or after the development process. This change in business rules requires the model to be changed and updated to cope with these new policies. Therefore, changing the model at the implementation level affects the maintainability and increase complexity.

In our view the business rules and policies should be considered and integrated at the early stages of the web development process. This is done through finding, documenting, modeling and implementing these rules. This approach will increase tractability and support the evolving new business operations. It increases consistency of the model at the different stages (i.e. requirement, design and implementation) of the development process.

Chapter Two

Web Site Design Method (WSDM)

In this section we now give a description of WSDM [15-18]. Since WSDM is the methodology framework of this research. First, we discuss WSDM characteristics followed by a description of all WSDM phases.

2.1 WSDM Characteristics

Audience driven

Most web site design methods are data or organization driven. The Data-driven methods take the data available in the organization as a starting point: they consider how to structure the web site based on the data. Organization-Driven methods take the structure of the organization, as a starting point and the structure of the websites is usually a reflection of the structure of the organization. The main characteristic of WSDM is the audience driven approach. So WSDM takes the requirements of the users of the web site as a starting point and uses this as basis for the structuring of data and the web site afterwards. WSDM gives consideration to the fact that web sites usually have different types of visitors that may have different needs.

Distinction between conceptual design and Implementation Design

WSDM conceptual design, which is free of any implementation details, is separated from the actual implementation design, like the presentation: the grouping in pages, use of menus, static and dynamic links etc.

Audience classes

From the fact that each intended audience(s) has his own requirements, and each web site has different ‘kinds’ of users, WSDM uses the concept of Audience Class. As an example, at the web site of Car rental Company we may distinguish the following users: Rental-Operator, Customers, and organizations-customer. Different users have different requirements, sometimes even different presentational needs, e.g. old people need special font size and color, while children may prefer a more use of pictures and animations.

WSDM classifies the users of a web site into audience classes: each audience class has its own requirements and characteristics. This may be reflected in the context (not all info for all users) and in the interface (language, jargon, look-and-feel, etc.).

The division into audience classes has as the advantage that the site is more adapted to each user’s needs, which will enhances the usability and the user satisfaction.

2.2 **An Overview**

At Figure 2.1 an overview of the WSDM method is given. The first step is to define the *Mission Statement*. The Mission Statement should express the purpose and subject of the web site, and declares the target audience.

Based on the Mission Statement, the *Audience Modeling* is performed, in two steps: *Audience Classification* and *audience class Characterization*. During the *Audience Classification* phase, the different types of users are identified, while during *Audience class Characterization*, characteristics of the different Audience Classes are given.

Next, *Conceptual Design*, in three steps: Information, Functional and Navigational Modeling. During *Information Modeling* we observe what kind of information is needed, while in the *Functional Modeling* phase we observe the functionality. The *Navigation Design* phase, we consider the global navigation through the information and the functionality.

The next phase, *Implementation Design*, performs *Page, Presentation and Logical Database Design*.. During these phases, grouping in pages, specifying the look and feel, and designing the database is included. The logical data base schema can be derived from the Business information model, which is made during the Conceptual Design.
The last phase, Implementation, is the actual realization of the website using the preferred implementation environment.

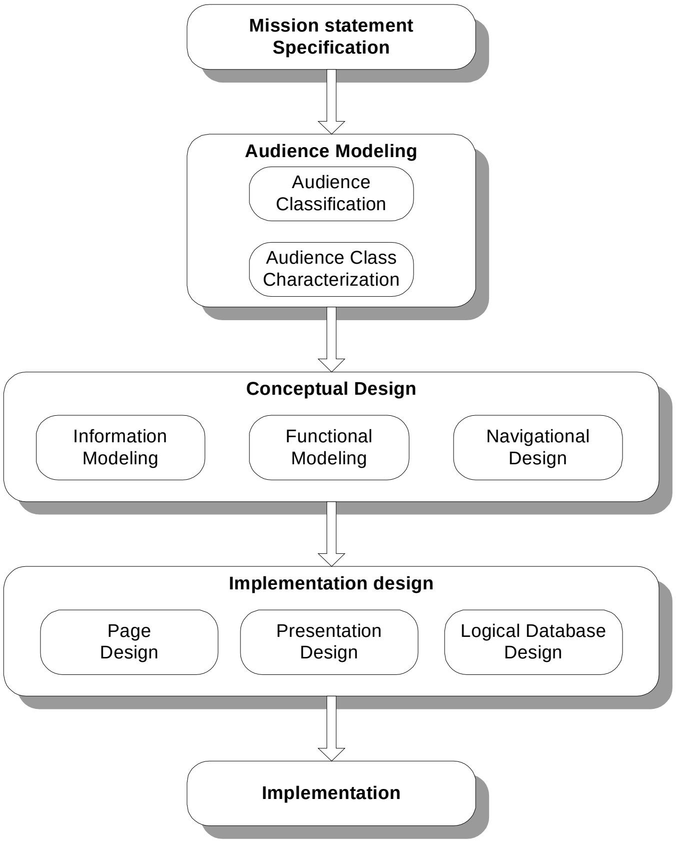


Figure 2.1: Overview of WSDM.

- ❑ **Mission Statement Specification**

The mission statement must answer the following three questions:

- What is the purpose of the site?
- What is the subject of the site?
- Who is the target audience(s)?

The mission statement for the Car Rental site can be formulated as follows:

“Provide an online renting where users can browse a list of cars for rent, get information about cars or company and rent cars. Company operators must be able to add or delete information about the cars.”

- Purpose
    - Offer information about cars and company online
    - Offer cars for rent online to all interested people
- Subject
    - Cars
- Target audience
    - Rental Operators
    - Everyone who wants information about cars for rent.
    - People who want to rent car online (private & Organizations)

- ❑ **Audience Modeling**

As discussed before, Audience Modeling phase consists of two phases:

- Audience Classification.
- Audience Class Characterization.

Audience Classification

A mission statement gives a general indication of the target audiences of the site. So at the *audience classification* the job is to identify and classify the different type of users participating in the activities supported by the site.

Target audience: Customers and everyone who wants information about cars or wants to rent cars online.

Audience Classes

- *Customer*
- *Organizations-customer*
- *Rental operator*

We can formulate the following high-level requirements for each audience classes as follows:

Audience Class Customer	
Information Requirements	*- Information about cars - information about Rental company contacts.*
Functional Requirements	*- Search by (car type, model,) for cars - rent a car online*
Usability Requirements	*- Flexible ways to search for information about cars.*
Audience Class Organizations-Customer	
Information Requirements	*- Information about cars - information about Rental contacts.*
Functional Requirements	*- Search by (car type, model,) for cars - rent car online - get specialized information about cars.*
Usability Requirements	*- Flexible to search for the needed cars* *- Flexible ways to rent cars online.*
Audience Class Rental Operator	
Information Requirements	*- Information about cars - information about customers*
Functional Requirements	*- Add information about cars - delete information about a car - add information about offers - delete information about offers*
Usability Requirements	*- Flexible ways to add or delete information about cars.*

WSDM has a feature to support subclasses where members of a subclass have all the requirements of the members of the super class plus additional requirements. At the above example, *Organization-customer* is a customer with extra requirements; he is customer that represents an Organization that asks to rent cars from Rental Company but with many number of cars each time.

The audience class hierarchy shows all Audience Classes in terms of sub and super classes. The class on the top is always the *Visitor*-class. Visitor groups all requirements common to all audience classes.
Audience class hierarchy for Car Rental Company as follows:

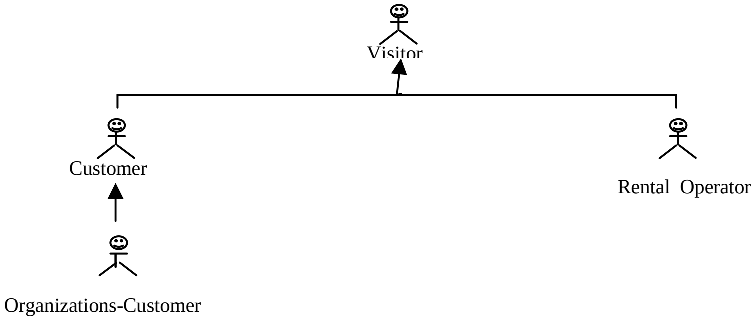


Figure 2.2. Audience Class Hierarchy for Car Rental example

Audience Class Characterization

As mentioned before, all members of an audience class have the same information and functional requirements. But members of one audience class may diverge on how the information should be presented to them, i.e. they may have different usability requirements. E.g. people who live in some areas like to have special colors or small kids like to see more pictures and things from their world. So there is a need for analyzing these characteristics. Some of these characteristics may be mapped into a usability requirement and others can be mapped during the implementation phase.

Audience class Characterization for car Rental site is formulated as follows:

Audience Class Customer

Characteristics:

- *Mostly adults*
- *Experience with WWW may vary*

Audience Class Organizations-Customer

Characteristics:

- *Ages older than 20 years*
- *Have reasonable experience with websites*
- *Have a good English language for communication*

Audience Class Rental Operator

- *Mostly adults*
- *experience with WWW may vary*
- *experienced with Rental company database applications*

❑ Conceptual Design

The conceptual design consists of three steps, the Information Modeling, the Functional Modeling and the Navigation Design. The information and functional modeling covers the conceptual what and the navigation design covers the conceptual how. The goal of the conceptual design is to turn the requirements into high level, formal description which can be used later on to generate automatically or semi-automatically the website.

Information Modeling

The purpose of the information modeling is to model the structured data of the data intensive web sites. This is achieved by modeling the information requirements of the different Audience Classes. Each information requirement is translated into so-called Information Chunks. An information chunk is a tiny conceptual schema that describes the information needed to satisfy a single information requirement. Any information modeling technique can be used; in this case we will use ORM to represent these chunks.

Information modeling for Rental company site is formulated as follows:

Customer Audience Class

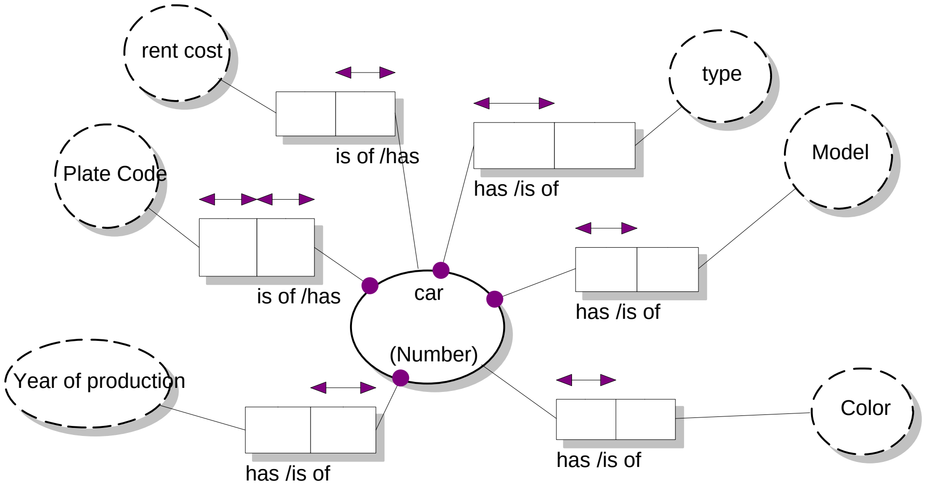


Figure 2.3. Information Chunk: Display Car Info.

Functional Modeling

The Purpose of the functional modeling is to model the functionality for the different Audience Classes. The functional requirements are, as well as the information requirements, elaborated into elementary requirements, and for each elementary requirement we make a Functional Chunk using ORM with extensions.
Audience subclasses inherit the Object Chunks of their audience super classes.

Functional modeling for EU-Rent site is formulated as follows:

Customer Audience Class

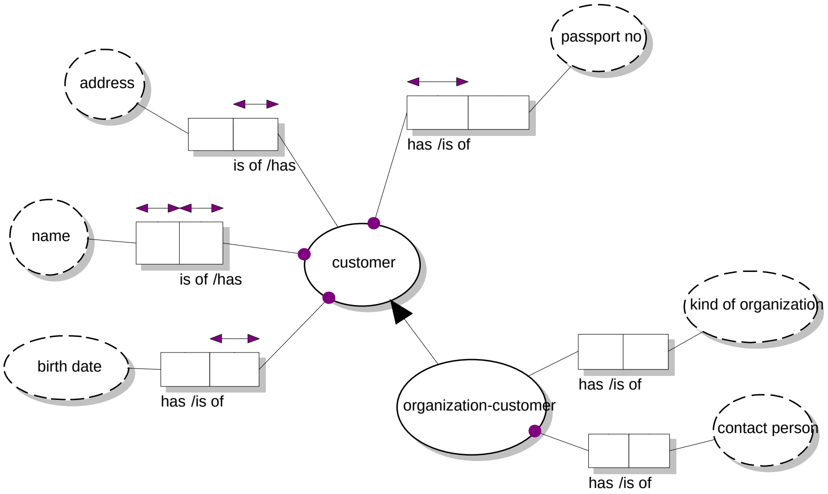


Figure 2.4. Informational Chunk: Submit customer info.

Navigational Design

Navigation design is to describe the (conceptual) structure of the web site and a model how the members from the different audience classes will be able to navigate through the site. For each audience class a different navigation track is created. So all navigation tracks together form the navigation model. Tracks are composed of components that represent units of information or functionality, and links that connects those components. In addition, links can be conditional depending on the logical true of a condition.

Navigational Design for EU-Rent site is formulated as follows:

Customer Audience Class

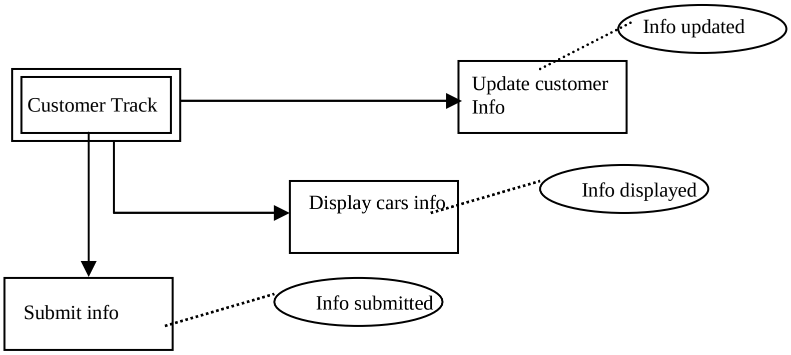


Figure 2.5 shows part of the Navigational Model for the Customer Track.

❑ Implementation Design

Implementation Design is the last phase of the design.
At this phase we essentially design:

1. The page structure of the website (Page Design step).
2. The " look and feel" of the site and the layout of the individual pages (presentation Design).
3. If there is a need for a Database, the logical data schema for that database is designed here.

Page Design

The page structure can be derived from the navigation model. For the components and links in the navigation model pages and hyperlinks will be defined. Each component can be assigned to a page and each link to a hyperlink. An important issue to be considered is that the download time of a page should not be too long.

Presentation Design

The look and feel and page layout is specified during the Presentation Design. Different navigation tracks may use different styles.
This step of the method is not yet elaborated.

Logical Database Design

A database is designed during this phase if needed. The logical database schema can be generated automatically (using tools like Info Modeler [30]), from the business information model or may be built manually.

- ❑ **Implementation**

This phase consists of the actual implementation of the web site. The implementation environment can be chosen: HTML, XML, WML, … The implementation could be automated using available tools.

2.3 **Conclusion**

From the detailed description of WSDM we notice that the different users of a web site receive the information concerning them in the way and flavor they like in complete correct form, also in exact and short time.

Chapter Three

Integrating Business Rules into WSDM Approach

3.1 Overview

The main theme of our approach is to include business rules in parallel with the audience requirements to obtain a successful and efficient e-business. This development goes in parallel with the current development process proposed by WSDM. Therefore, it also has four phases, which will be defined as follows:

- Mission statement and Business Policies: at this level the business policies are defined.
- Audience classes and Business Rules modeling: at this phase Business rules are written in natural language in parallel process then assigned to the different audience classes. Forming an input business rules to the Audience modeling phase.
- Conceptual Modeling and Business Rules: at this phase business rules are integrated into the conceptual model.
- Implementation: At this level the rules are implemented and managed using a rule engine.

The structure and the evolution of the business rules are shown in figure 3.1 below.

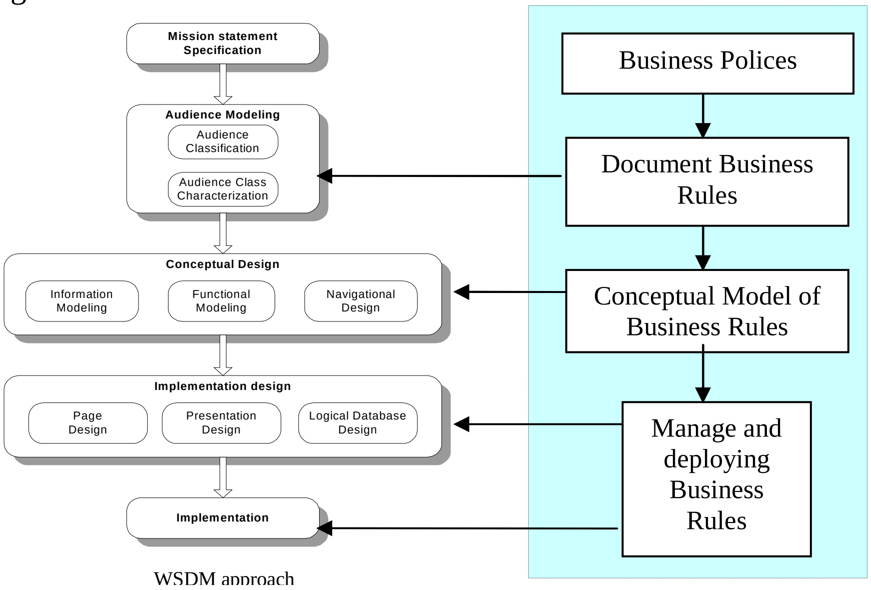


Figure 3.1: Approach overview

Evolution of business Policies throughout the web site development process

The business Polices should be involved during the complete development process and should be consistent with their counter parts at each phase. In other words these rules should not be violated during the early development process. This process ensures traceability and supports the changes in the rules that might occur during the development process. In addition to that it simplifies the model making it easier to be understood.

3.2 Declaring Business Rules parallel to Mission Statement Specification

In this step we define the enterprise business policies, which is a source of any business rule that might be found at any business. The first step of WSDM is to define a mission statement of the site. We suggest defining all business policies of the enterprise in parallel or independent with the Mission Statement phase.

A Business Policy could be defined as follows:

> ¨*Business Policy is a general statement, whose purpose is to guide the enterprise* ¨

Business Policies provide the basis for Business Rules. Generally, Business Policies exist to control, guide, and shape the strategies and tactics at the enterprise. Each policy might be composed of one or more general policies. For example a Business Policy for Car Rental business might be:

> ¨ *Always attempt to give customers special offers.* ¨

Another example of a Business Policy for E-Business Company might be:

> *¨A business representative will personally contact each customer who makes a complaint¨.*

The Mission statement of the Car rental enterprise site can be formulated as follows:

> *“To Provide online rent where users can browse a list of cars for rent, get information about cars or company and rent cars. company operators must be able to add or delete information about the cars.”*

In addition to that, Car Rental business policies can be formulated as follows:

- *¨ We only rent cars in legal, roadworthy condition to our customers. ¨*
- *¨Safety first. ¨*
- *¨ Always attempt to give customers special offers. ¨*
- *Etc. …*

Business policies that exist merely to enable a strategy in direct manner should be avoided. For example, suppose the enterprise has the following strategy:

¨Encourage repeat business. ¨ A business policy that says *¨Repeat Business should be encouraged¨* is trivial and does not need to be addressed.

Finally as mentioned above, each of these business policies that we defined in parallel or independent with the Mission Statement phase will form the bases of the enterprise Business Rules.

3.3 Integrating Business Rules Parallel with Audience Modeling Phase

In this section, business rules will be defined and documented according to the enterprise Business Rules analyst and in natural language.

3.3.1 Business Rules Modeling

In this section, we define Business Rules and document them. According to our understanding of WSDM Audience Modeling phase, different audience classes and characteristics are defined at this phase. We suggest in this phase also to declare and document business rules in a natural language as an independent model. As a result rules of the model will be assigned to the audience classes at the Audience Modeling phase, this model is called Business Rules Model, this because each of the business rules that might be defined at this model is related to one or more of the audience classes. Since these business rules are written in a natural language and addressed at the early design stages, they are easy to be understood by both the developer of the site and the enterprise side.

A Business Rule could be defined as bellow:

¨A business Rule is directive, intended to influence or guide business behavior, in support of Business Policy. It is a single element of guidance that does not require additional interpretation. Often a Business Rule is derived from Business Policy¨.

As already mentioned earlier, the aim of adding the Business Rules Model to the Audience Modeling phase of WSDM is to address and document business rules at the early stages of the web development; as a result the Business Rules Model now includes all of the enterprise business rules, which are declared according to the Business Rules Analyst decision, who should be an expert in both business rules and business operations. The output of this approach is that each of these business rules is assigned to one or more of the audience classes.

To clarify the picture of this approach, let us consider the following example, at the Car Rental site, the requirements of both *Customer* and *Rental Operator* audience classes are defined, besides that, the *Business Rules Model* where each business rule is assigned to a related audience class, is defined.

Customer Audience Class

Functional Requirements:

Fun_Req1: *Search by(car type, model, …) for cars*
Fun_Req2: *Rent a car online*

Information Requirements:

Info_Req1: *Information about cars*
Info_Req2: *Information about Rental Company contacts*

Rental Operator Audience Class

Functional Requirements:

Fun_Req1: *Add information about cars*
Fun_Req2: *Delete information about a car*

Information Requirements:

Info_Req1: *Information about cars*
Info_Req2: *Information about customers*
Info_Req3: *Add information about offers*
Info_Req4: *Delete information about offers*

Business Rules Model

Business Rules related to each of the audience classes could be formulated as shown below in the table, for simplicity, Business Rules are presented in a table form.

Rule #	Business Rule	Audience
BR1	*¨A car with a car motor more than 5000 CC must be of high rate cars group.¨*	Rental Operator, Customer
BR2	*¨A customer must provide a valid driver's license in order to rent a Vehicle¨*	Customer
BR3	*¨The rental of a car within 500 miles of its next service mileage may no be extended by online¨*	Rental Operator, Customer
BR4	*¨Rental charge is based on base rental price, optional insurance, and refueling charge¨*	Customer

As described above in the last example, each of the audience classes may share one or more business rules. As a result of defining and documenting all of the business rules that might control the operations performed by each of the audience classes, it's easy for the business analyst or the enterprise manager to understand how each of the operations and calculations is performed. Also it's easier for him to change or update these business rules. In the next section we will discuss different types of Business Rules that is necessary to be declared and documented at the *Business Rules Model*.

3.3.2 Types of Business Rules necessary at the Business Rules Model.

In this section we describe the different types of business rules necessary to be defined and documented at the Business Rules Model in natural language during the development phases of e-Business sites.
There are four types of business rules; each of these types is discussed as below:

- **Definition of business terms**

The most basic element of a business rule is the language used to express it. The very definition of a term is itself a business rule, which describes how people think and talk about things. Thus, defining a term is establishing a category of business rule.

E.g., "Car model", " customer", "general motors", "5000"

This type of business rules is covered in some way in the requirements of each audience class, at the Audience Classification part of the Audience Modeling phase of WSDM; it is also covered at the conceptual model of the different requirement chunks and the Business Information Model of WSDM. ORM Info modeler is used to represent each of these Business Rules. Therefore, at this phase, this type of business rules will not be declared and documented as an independent terms.

- **Facts relating terms to each other**

The nature or operating structure of an organization can be described in terms of the facts, which relate terms to each other. To say a customer can place an order is a business rule. Facts can be documented in natural language sentences or as relationships, attributes, and generalization structures in graphical model.

E.g., " Name is an attribute of customer."
" Color is an attribute of car"

Generalization Example might be:*" A rental branch manager is an employee."*

A participation (relationship) expresses a fact in which a set of terms is associated in some sense, which is meaningful to the business, for example:

"A car model can be requested by many customers "

Also this type of business rules is covered at the conceptual model of the different requirement chunks and at the Business Information Model. ORM Info modeler is used to represent each of these Business Rules.

- **Constraints ("action assertion Rules")**

Every enterprise contains behavior in some way, and this is closely related to constraints on what data may and may not be updated. To prevent a record from being made is, in many cases to prevent an action from taking place.

"If a car category is ¨A¨ then add 4% to the rental charge "

This type of Business Rules is our goal of study in this thesis; also it's not directly supported by any of the web design methodologies that we discussed at the background part including the WSDM methodology. Therefore, These rules are declared and documented at the Business Rules Model of our approach. As a result it's easy for the business to analyze or change this important type of business rules at the early development phases.

- **Derivation Rules**

Business rules (including laws of nature) define how knowledge in one form may be transformed into other knowledge, possibly in deferent form. Derivation is of two types, computation rules, and Inference Rules:

Computation Rule: any rule that computes a value automatically, following standard mathematical operations.

> *E.g.: "the rental amount, in rental is calculated from the rental rate multiplied by the ¨number of days ¨*

Inference Rule: A derivation that produce a derived fact using logical induction (from particulars) or deduction (from general principles).

> *E.g.: "A car model (e.g., Sedan) is in a car class (e.g. Class C)."*

This type of Business Rules is also our goal of study in this thesis, and it's not supported by any of the web design methodologies that we discussed at the background part including WSDM methodology. Computation Rules type of these rules are documented at the Business Rules Model of our approach. Also when we document this type of business rules, it becomes easy for the business to analyze or change this important type of rules at the early development phases. Inference Derivation rules are to be covered at the conceptual and implementation levels we will discuss in the next sections.

3.4 Modeling of Business Rules at the Conceptual Model

This section shows how business rules can be modeled at the conceptual level. ORM notation is used to represent static rules and the RIDL language [14] is used to model dynamic and procedural rules. ORM and RIDL rule systems complement each other, and they are powerful enough to enable representing the business policies of a domain.

3.4.1 Modeling business rules using ORM

In this section, we discuss ORM's capability in modeling business rules at the conceptual level, we show, and illustrate by examples, the importance of modeling business rules at the conceptual level of websites.

Object Role Modeling (ORM) [26] is a conceptual modeling methodology, derived from NIAM (Natural-language Information Analysis Method), which is also one of the other known names for ORM.

The essential concepts of ORM are those of *object*, *role* and *fact.* Objects are the things being modeled (e.g., terms of an schema), while facts are the declarations that are made about objects (e.g., their relationships). A fact relates two or more objects; each of those objects playing a different role in the fact. Objects are put into group of what is called *object types* (denoted graphically by circles) and facts are put into group of what is called *fact types* each of these fact types are made of one or more role (denoted by rectangles with a line pointing to an object type).

ORM notation is preferred by WSDM developers to be used as a modeling graphical notation at the conceptual model. WSDM methodology collects and defines all the requirements of the different audiences at the audience modeling phase, and afterwards they model these requirements at the information and functional chunks of the website also according to each audiences class. Thus ORM is used to capture all the requirements because ORM has many advantages in modeling, the most important one that is related to our work is that business rules can be captured in its powerful graphical notation, which is represented at the conceptual design in a way that could be easily understood by domain experts. Other methods like Unified Modeling Language (UML) class diagrams and Entity Relation diagrams are closed to the final database implementation [26]. ORM is also used to assisting the modeling, validating, and maintaining processes. Therefore it provides the possibility for managers, analysts, or domain experts to be involved during the modeling of object types, where business rules are expressed using their own vocabulary.

Another advantage of ORM as already discussed by [8] is that its wide and powerful graphical notation used to represent a business or domain in a declarative manner as a network of elementary facts and their constraints can be easily re-verbalized into statements in pseudo natural language in a structured and fixed syntax. Therefore business rule modelers could represent a business policy in one or both representation ways, graphically or textually, which will in general improve, simplify, help to validate, as a result of that modeling process is speeded up.

In order to facilitate the discussion of the next section, the most usually used notation for modeling any domain using ORM are provided:

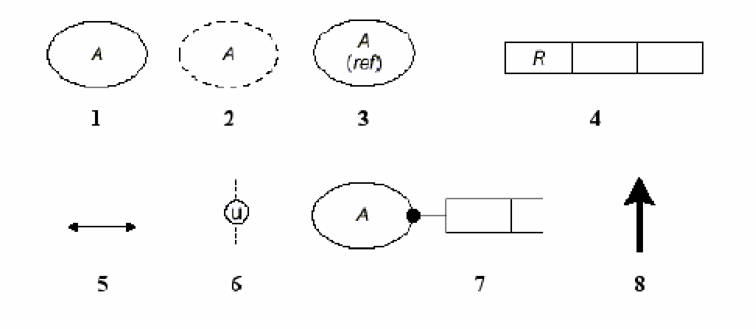


Figure3.2: Common ORM symbols

1. Entity type
2. Value type
3. Abbreviated reference scheme
4. Ternary predicate compromised of three roles
5. Internal uniqueness constraint
6. External uniqueness constraint
7. Mandatory role constraint
8. Sub typing

Types of rules represented by ORM

ORM has static and some dynamic constraint types and derivation rules that are expressive enough and can be used to represent many of the business rules rising during modeling of the enterprise. Those constraints and rules include classical types such as mandatory and uniqueness constraints, in addition to other types as value, subset, subtype, equality, ring, and derivation. Other types of rules that are not mentioned above will be formulated using RIDL, an expressive general-purpose language.

In order to model some of the business rules that could be expressed with ORM constraints, various ORM chunks are shown bellow.

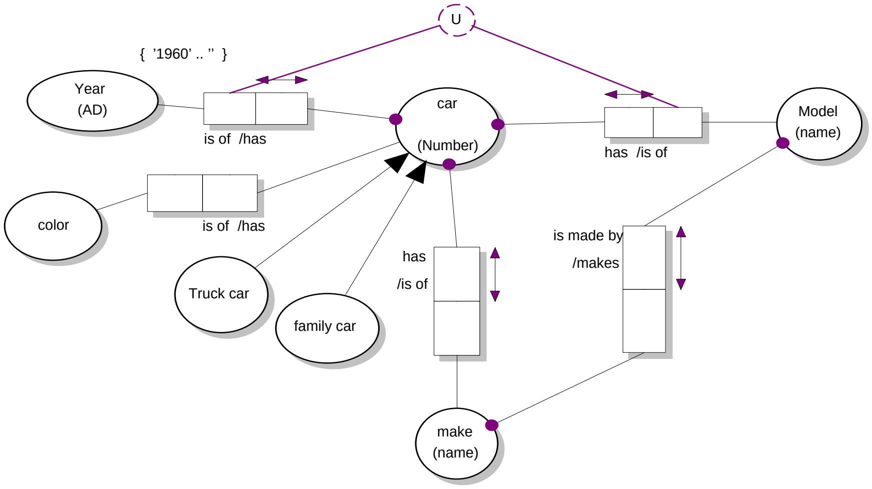


Figure 3.3 Functional Chunk: Submit car info.

- Mandatory constraint: used to indicate that all objects of a type must take part in a given role. Mandatory roles are indicated by a black filled circle at the line of the role. In the figure above the roles ¨has model¨, ¨ has make¨, ¨ has year¨ are mandatory roles, while the roles ¨ has color¨ is not mandatory.

Business rule example: ¨ *it is not allowed to add cars to the list of cars for rent if its company of make is not provided¨*

This type of constraint is modeled and verbalized as follows*:*
¨Each car has at least one make¨

- Uniqueness: indicates whether entities of a type can participate more than one in a role. A uniqueness constraint is indicated by the double arrow over the role, e.g. on ¨has model¨, which means that each car has at most one car model.

Business rule example ¨ *each car number at the company represents only one specific car ¨*

This type of constraints is verbalized as: *¨Each car number is at most of one car ¨.*

- External uniqueness: is used to indicate that the combination of roles is used to identify a specific entity. As shown in figure 3.3, the uniqueness constraint indicates that ¨ is year of¨ and ¨is model of ¨ as a combination identifies a specific car.

Business rule example ¨ *each of our cars are identified at the company by both of its production year and model¨*

- Value constraint: enumerates in extension or through ranges, the values that a value type can take. In figure 3.3, a value constraint is placed over the type AD indicating that the company only deals with years greater than or equal to 1960.

Business rule example ¨ *don't allow adding a new car if its production year before 1960 ¨*

- Subtype: captures the notion of subtypes, and is denoted by a directed arrow, e.g. between organization-customer and customer object type; it also allow multiple inheritance from super types.

Business rule example ¨ *each organization that rent cars from our company is a customer and they are contacted by their contact person ¨.*

This type of constraints is verbalized as*: ¨Organization-customer is-a customer. ¨*

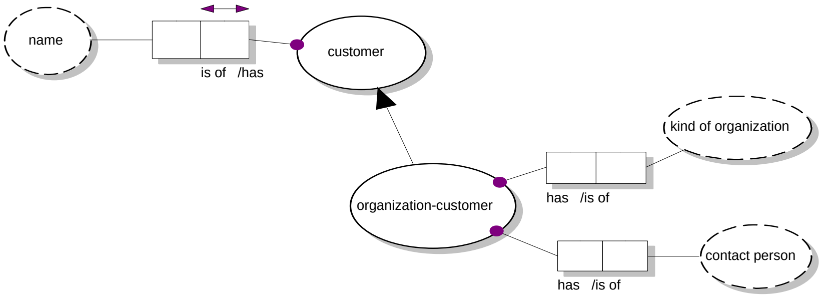


Figure 3.4 part of information Chunk: org_ customer info.

- Subset constraint: is used in comparison and restricts the existence of some fact in relation with another fact, This type of constraint is represented using an arrow in between the two facts as shown in figure 3.5. This type of constraint is verbalized like this*: ¨ each customer who reserve a car must has a driving license. ¨*

Business rule example: ¨ *we only rent cars to customers who have driving License*

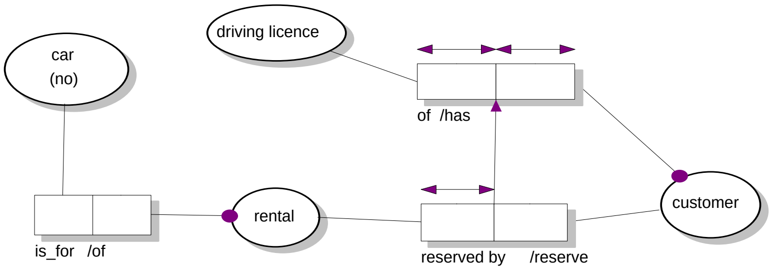


Figure 3.5 part of functional Chunk of subset constraint

- Derivation: is used to define how one knowledge form could be transferred into another. One can distinguish two types of knowledge as shown in the two figures below. Both types are represented using an asterisk beside the derived fact. The first type as show in figure3.6 is computation derivation, which cannot be fully specified in ORM, the asterisk only shows that the fact is a derived fact without providing explicitly the computation formula, but this can be formulated using a language like e.g. ConQuer[1]. For this reason, in the next section RIDL will be used to model this type of derivation. The second type, which is the inference derivation, could be modeled using ORM as illustrated in figure 3.7, where car model is derived from the model of class.

Business rule example of computation derivation as shown in figure 3.6: *¨Rental charge is based on base rental price, optional insurance, and refueling charge¨.*

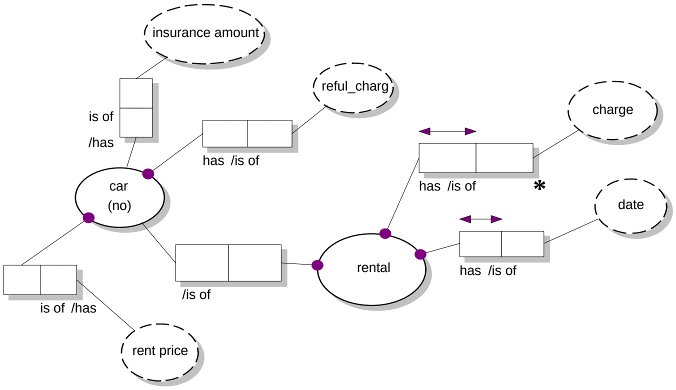


Figure 3.6 ORM Chunk shows a computation derivation

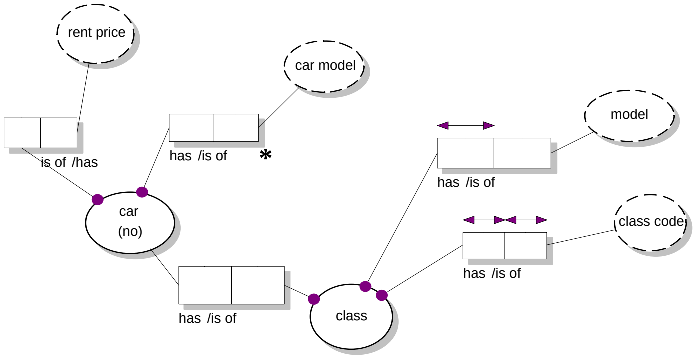


Figure 3.7 ORM Chunk shows the inference derivation

The above mentioned examples about modeling specific types of business rules using ORM notations, illustrate how powerful ORM is as a tool to model business rules using ORM constraints, however other essential types of business rules still can not be captured using ORM alone. Therefore the RIDL Language, which will be the subject of the next section, is needed.

3.4.2 Modeling business rules using RIDL

RIDL (Reference and IDea Language)[14] is a powerful conceptual language defined on top of ORM. It is an integrated formal syntactic support of information and process analysis, query/update, semantic specification, as well as constraint definition language at the conceptual level, rather than the logical. RIDL is considered as a complementary of the conceptual modeling notation ORM

RIDL syntactical elements

The referencing expression is the essential component of RIDL syntax, which is the expression that evaluates to one or more occurrences of the population of which the types are described in the conceptual schema, arithmetic expressions are also included at the referencing.

Using RIDL language and syntax, complex constraints as well as simple ones can be represented. This is due to the fact that the RIDL syntax includes all the syntactic concepts that might be needed to model constraints such as set expressions, predicates, procedures and functions, tables, naming statements, control statements, and verb statements. Therefore, by using a combination of these concepts, any constraint could be expressed. Below, some of these concepts as described in [14] are summarized.

Control statements:

Control statements are used in RIDL to direct the flow of control. Some of the control statements are listed below:

¨ WHILE . . . Do¨, ¨ REPEAT . . . UNTIL ¨, ¨ IF . . . THEN . . . ELSE¨, ¨ WHERE¨, ¨ FOR EACH ¨
All of these control statements should be used with arithmetic’s expressions such as (-, +, /, *), Equality, Increment, decrement… etc.

Predicates:

Truth-value of some Boolean expression is established using a predicate construction. A predicate may has several forms such as:

```
occurrence   IS   occur-relation-operator occurrence
occurrence   IS   set-relation-operator set
set          IS   set-relation-operator set
set          IS   reference-path
```

Where occurrence-relation-operator is one of: ‘<> ‘, ’ < ‘, ’ <=‘, ‘>=‘, ‘=’, ‘ >‘,
set-relation-operator is one of : ‘A’, ‘AN’, ‘INCLUDED IN’, ‘EQUAL TO’, ‘DISJOINT-FROM’

¨AND¨ and ¨ OR¨ could be used to combine two predicates; the result of this combination is a predicate as well. Predicates could also be used in control constructs as ¨ IF¨ as illustrated in the next paragraphs.
¨FOR EACH¨ is a construct that gives the ability to perform actions on each occurrence of a set individually.

Set Functions

Number of**:** which gives the number of elements in a specific set; this function is valid on all types of sets.

Average of: which gives the average of numbers that are in a particular set; this function is only valid for numeric sets (average = sum of all elements / number of elements).

Equality references

It is permitted to omit the equality operator when a reference end with an equality condition (= or EQ) on the last Object Type (OT) of the reference path. It is also permitted some times to omit the OT on which the equality condition is placed as shown bellow.

```
OT1    role1   OT2  = X   where x is an arbitrary value
                          of the type of OT2
```

May be abbreviated to:

```
OT1     role1      x
```

Example:

```
IF   Request-code   of   Request  = ¨2¨
```

Example:

```
IF   Request-code   of   Request   requested_by
customer  ¨2¨
```

Types of business rules modeled by RIDL constraint

As mentioned earlier, not all business rules can be represented by ORM notation, specially in the case of action assertion types and computation derivation, these types of dynamic rules might be changed often according to the business needs, thus, the goal is to model these types of rules using the powerful concepts of RIDL, such as different control statements, cardinality, and arithmetic operations. Before we start modeling these constraints, a simple RIDL fragment is provided with its equivalent ORM chunk, so as to show how object types and classical constraints are defined in RIDL.

Model optionally constraints using mandatory and uniqueness constraints

```
add conceptual schema Rent-payments
  add nonlex Customer(Cust#), car, money($)
  add fact rent
    roles (customer rents, car rented-by)
  add fact payment
    roles (customer pay, Money  related_to)
  add constraint One-payment-per-customer
    condition customer pays only-one Money holds
  add constraint Each-customer-must-rent-a-car
    condition customer always rents car holds
  add constraint customer-must-pay-money
    condition customer always pay Money holds
```

The above RIDL fragments could be represented in ORM, as in the following figure.

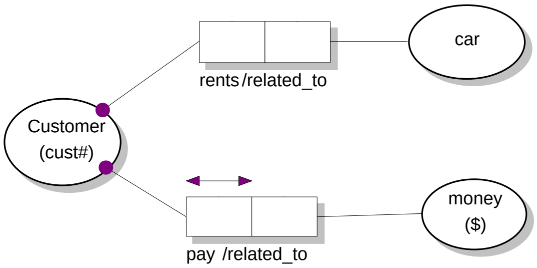


Figure 3.8: ORM chunk of classical constraints

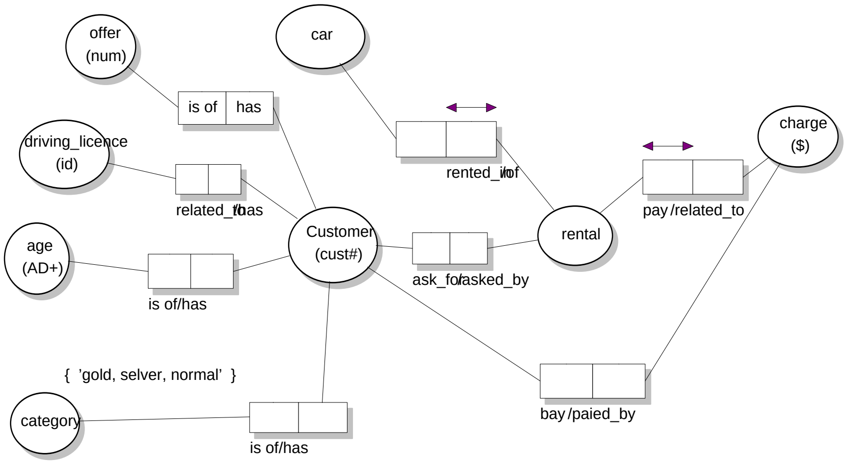


Figure 3.9 Information Chunk: customer rental info

- **Model action assertion business rules**

This type of business rules can be modeled by means of RIDL complex constraints according to the ORM chunk shown in figure 3.9.

- ❑ **Business Rules that used to reject an action to happen**

Business Rule 1: using ¨*for each* ¨and ¨*if* ¨control statements

¨It is not allowed that a customer rents a car if he doesn't pay the specific charges of it.¨

RIDL fragment:

```
 add constraint cars_require_payments
   Begin
     for each c: customer
        for each r: rental
          if (c is asks_for r)
           then
           condition (charge is related_to r asked_by c) holds
           end if
        end for
     end for
   end
```

Business Rule 2: using occurrence operator ¨>¨ & ¨and¨

¨any customer can rent a car if she has a suitable license for it, and

her age is greater than 25 years old ¨

RIDL fragment:

```
add constraint rent_require_licence
Begin
  for each c: customer
    for each r: rental
     if (c is asked_for r)
     then
      condition(c is has_licence and c is has_age >¨25 ¨)holds
     end if
    end for
  end for
end
```

Business rule 3: using a combination of concepts and in different styles of the above.

¨a customer can get more than one offer, if she is more than 25 years old, and she has a silver category or gold category. ¨

RIDL fragment:

```
add constraint offers_for_customers
  for each customer who (has category ¨silver¨ and has
                     age > ¨25¨)                or
                    (has category ¨gold¨)
    condition
    number of(offer for customer)>=1
    holds
  end for
```

Business rule 4: using path equivalent constraints

¨The employee who is responsible of confirming rentals at the company branch, is an employee at the branch. ¨

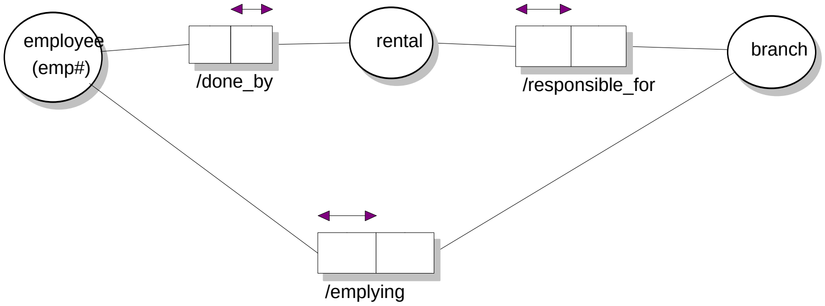


Figure 3.10 ORM chunk of path equivalent constraint

RIDL fragment:

```
add constraint path_equivalent
  condition
    For each employee emp
      condition
      branch employing emp
       equal to
         branch responsible_for rental done_by emp
      holds
    end for
  holds
```

- **Business Rules that do a kind of action**

Business rule 1: using cardinality constraints

¨ any customer of three successful rents through any promotion and the number of his successful rents is greater than the number of his cancelled ones becomes a gold customer ¨.

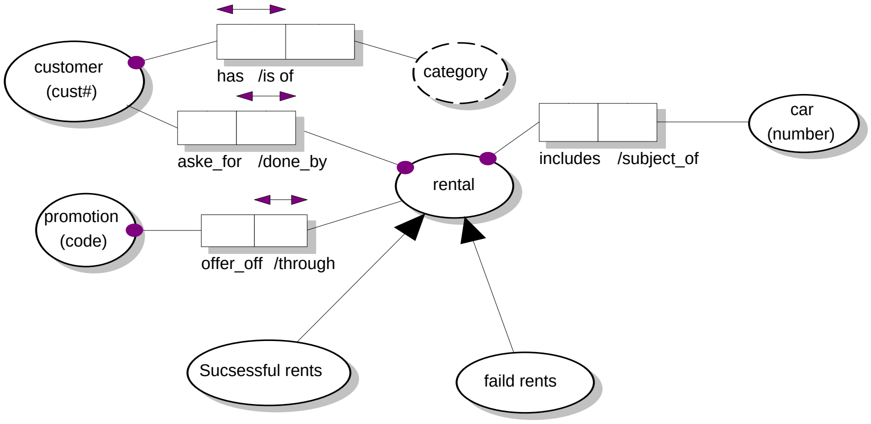


Figure 3.11 ORM chunk of promotion rental info

RIDL fragment:

```
add constraint become_gold_customer
   for each c customer aske_for rental through
            a promotion
   do
   Success: = successful rents (done_by c and
                     through a promotion)
   Fail: = failed rents (done_by c and through a
                         Promotion)
      If number of (success)<= number of (fail) and
         Number of (success) is > 3
      Then
      add category of c equal to ¨gold¨
      end if
   end for
```

Business rule 2: Excluding an item from a set

¨Disable all cars made in the same country of the customer from the list of cars for rent for that customer ¨.

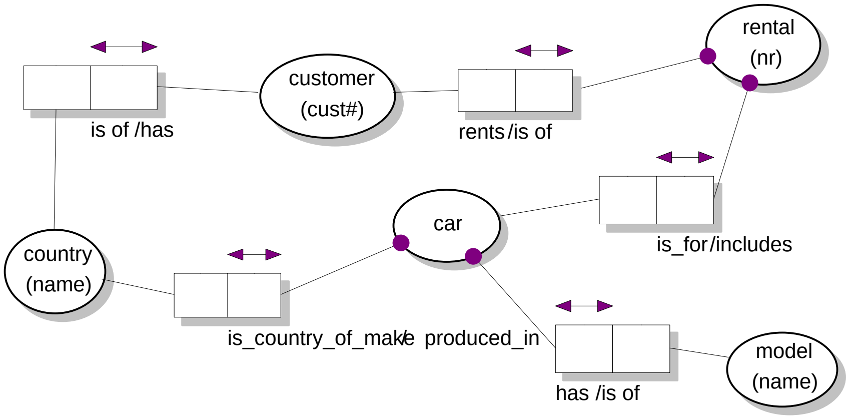


Figure 3.12 ORM chunk of cars for rent

RIDL fragment:

```
add constraint not_to_rent_your_countrymade_cars
 for each customer c
   Condition
    Country is_country_of_make car is_for rental
    of customer c is not equal to
    country is_of customer c
   holds
```

Business Rules of derivation type

Business rule: using number of expression and a combination of control statements to derive a value.

¨Each customer gets a 25% discount on his rentals after three successful rents¨

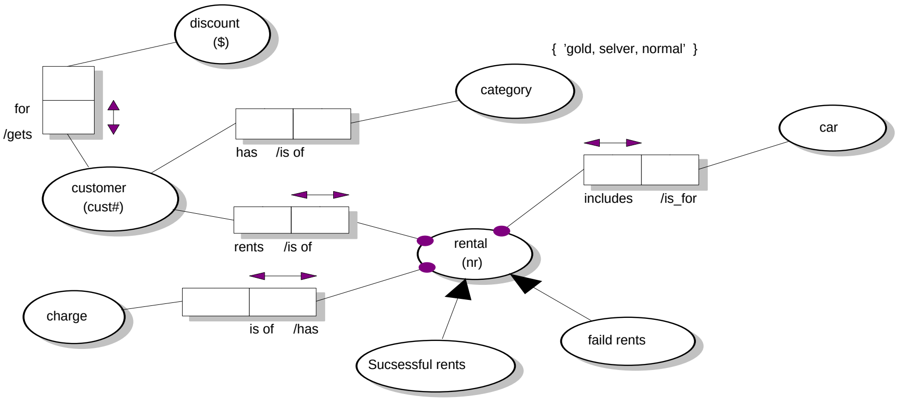


Figure 3.13 Info chunk of rentals

```
add constraint discount amount
  for each c customer
      if c has category ¨normal¨ and
        number of(successful rents of customer c)is >= 3
      then
       add discount for c equal to 25
      end if
  end for
```

3.4.3 Representation of business rules at the Business Information Model

In WSDM the Business Information Model includes the different information and functional chunks of the different audience classes. However this model does not consider the business rules defined by the enterprise.

In the following, we will present an approach to represent and integrate declarative business rules. Our approach is inspired from [30].

As different audiences can use different set of chunks, different chunks can also impose the same or different rules. Therefore we have to be careful in representing those rules to avoid redundancy and inconsistency in the business information model. The way to solve this is to start by the audience classes that use the same information and functional chunks as well as business rules, therefore we group the audiences in a table, the chunks used by them in an other table, and the rules imposed on those chunks in a third table. So, the business information model is represented by three tables (Audience classes, chunks, and business rules). This is illustrated at the figure below.

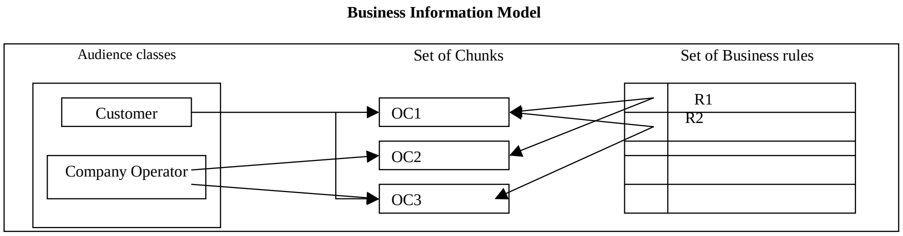


Figure3.14: business information model structure

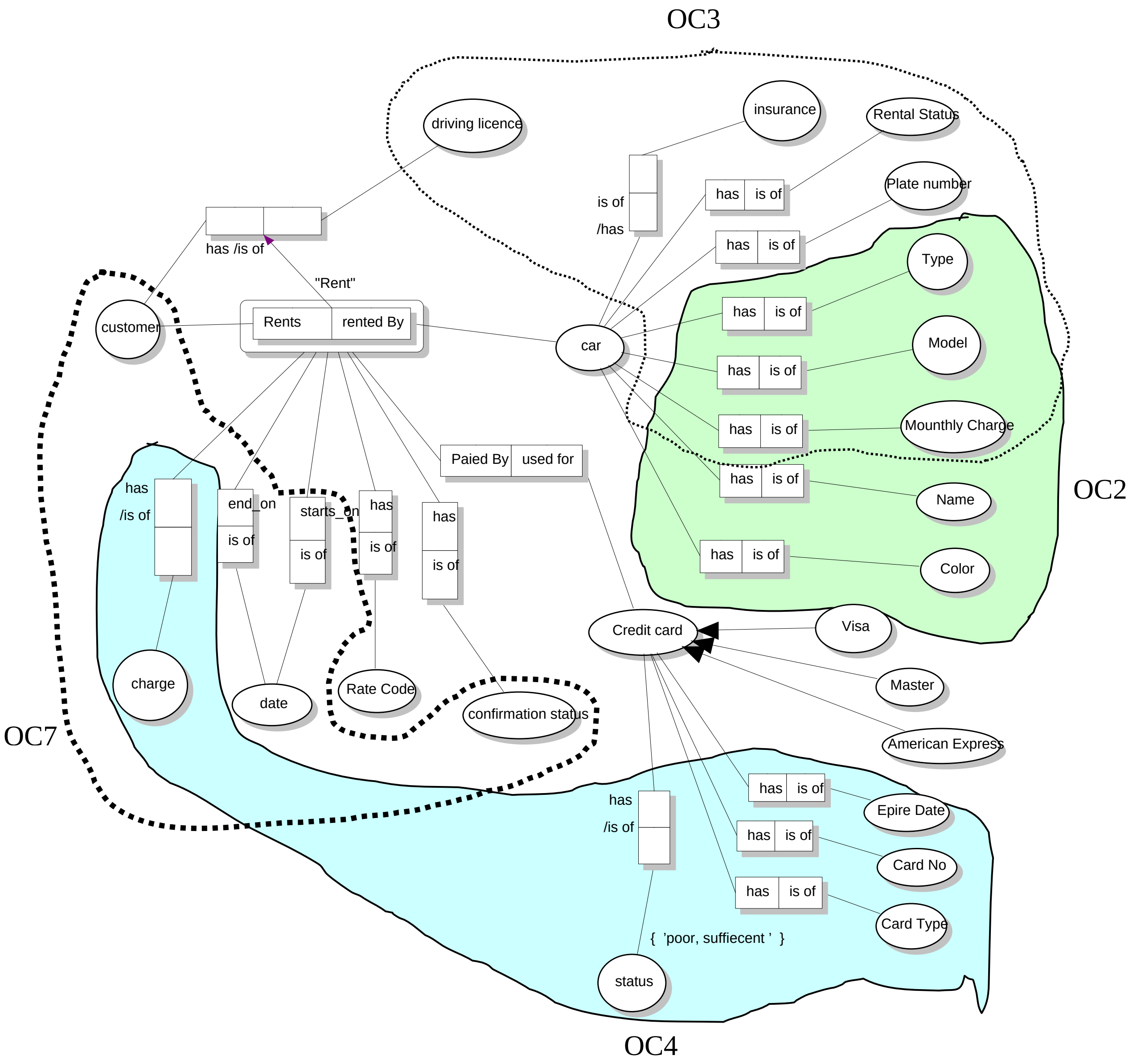
Example:

This example represents the car rental company as used before. It should allow the customers to rent cars, it allows the rental operator to add or change information about cars and rents related to customers. The customers who want to rent cars should have a valid driving license and the required rent amount in their credit card. Therefore, the customers and the rental operator represent the audiences, the renting operation represents a functional chunk, and the constraints of the valid driving license represent a business rule, other functional and information chunks and rules as well. Those together form the business information model, which are represented graphically and formally in the figures and tables below.

In figure 3.15 below, seven different information and functional chunks are shown, as:

- OC1: customer information;
- OC2: payment information;
- OC3: car info;
- OC4: submit car information;
- OC5: rent information;
- OC6: offer information;
- OC7: submit offer information.

The example shows that audience class customer use chunks OC1, OC3, OC5, OC6, while Rental-Operator audience class use chunks OC1, OC2, OC4, OC5, OC7.


OC3
driving licence
insurance
Rental Status
is of
/has
has
is of
Plate number
has
is of
has /is of
"Rent"
Type
has
is of
customer
Rents
rented By
car
Model
has
is of
has
is of
Mounthly Charge
has
is of
Name
OC2
Paied By
used for
has
/is of
end_on
is of
starts_on
is of
has
is of
has
is of
has
is of
Color
Credit card
Visa
Master
American Express
charge
date
Rate Code
confirmation status
OC7
has
is of
Epire Date
has
/is of
has
is of
Card No
has
is of
Card Type
{ 'poor, suffiecent ' }
status
OC4

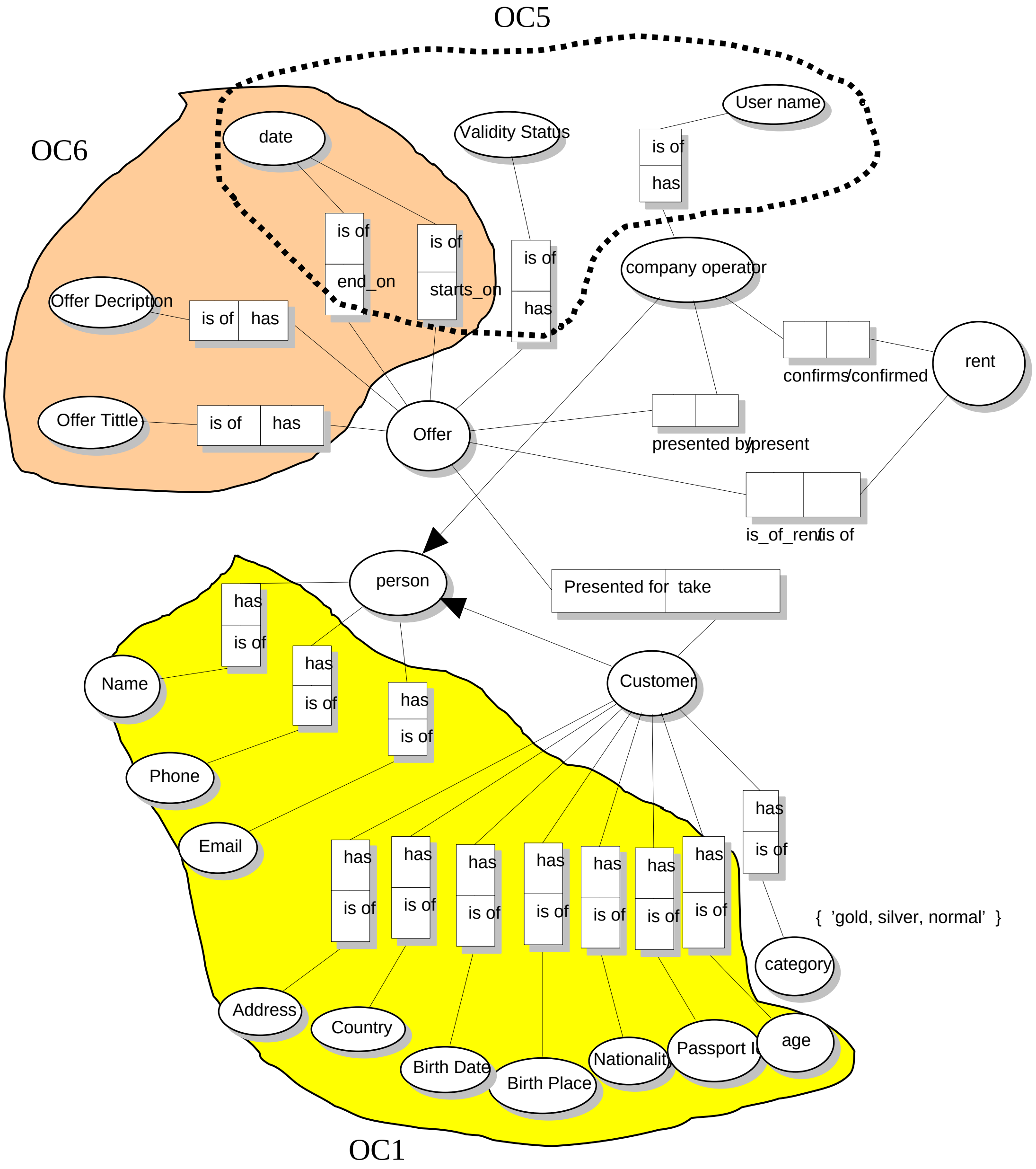


Figure 3.15: Chunks of the Car rental Business Information model

The following three tables show the organization and representation of business rules; these rules are listed according to the business information model in figure 3.15. For example the fourth rule shown in the table restricts each person to have a driving license in order to be able to rent a car.

Table 3.1: Set of business rules

RuleID	Rule
1	Each **person** has **name**
2	Each **person** has **phone**
3	Each **person** has **email**
4	Each **customer** has **driving-license**
5	Each **customer** has **address**
6	Each **customer** has **country**
7	Each **customer** has **birth date**
8	Each **customer** has **birth-place**
9	Each **customer** has **nationality**
10	Each **customer** has **passport-id**
11	Each **customer** has **age**
12	Each **customer** has **category**
13	Each **offer** has **offer-title**
14	Each **offer** has **offer-description**
15	Each **offer** has **start-date**
16	Each **offer** has **end-date**
17	Each **offer** has **validity-status**
18	Each **company-operator** has **user-name**
19	Each **Rent** has **charge**
20	Each **Rent** ends-on **date**
21	Each **Rent** has **rate-code**
22	Each **Rent** has **confirmation-status**
23	Each **Rent** ends-on **date**
24	Each **Rent** starts-on **date**
25	Each **Car** has **insurance**
26	Each **Car** has **rental-status**
27	Each **Car** has **plate-number**
28	Each **Car** has **type**

29	Each **Car** has **model**
30	Each **Car** has **monthly-charge**
31	Each **Car** has **name**
32	Each **Car** has **made-year**
33	Each **Credit card** has **expire-date**
34	Each **Credit card** has **card-no**
35	Each **Credit card** has **credit-type**
35	Each **Credit card** has **status**
37	**Visa** is-a **credit card**
38	**Master** is-a **credit card**
39	**American express** is-a **credit card**
40	**Company-operator** is-a **person**
41	**Customer** is-a **person**
42	Each **person** *has* at least one **name**
43	Each **customer** *has* at least one **driving-license**
44	Each **customer** *has* at least one **birth date**
45	Each **customer** *has* at most one **passport_id**
46	Each **Passport-id** *represents* at most one **customer**
47	Each **car** *has* at least one **name**
48	Each **car** *has* at least one **type**
49	Each **car** *has* at least one **monthly charge**
50	Each **car** *has* at most one **plate-number**
51	Each **plate-id** *represents* at most one **car**
52	Each **car** *has* at least one **rental status**
53	Each **rent** *has* at least one **starting date**
54	Each **rent** *has* at least one **end-date**
55	Each **rent** *has* at least one **confirmation status**
56	Each **rent** *has* at least one **charge**
57	Each **company operator** *has* at most one **user name**
58	Each **user name** *represents* at most one **manager**
59	Each **credit card** *has* at least one **card type**
60	Each **credit card** *has* at least one **expire date**
61	Each **credit card** *has* at most one **card no**
62	Each **card no** *represents* at most one **credit card**
63	Each **credit card** *has* at least one **status**

64	add constraint path_equivalent condition For each employee emp condition branch employing emp equal to branch responsible_for rental done_by emp holds end for holds

65	add constraint cars_require_payments
	Begin
	for each c: customer
	for each r: rental
	if (c is asks_for r)
	then
	condition (charge is related_to r asked_by c) holds
	end if
	end for
	end for
	end

66	add constraint offers_for_customers for each customer who (has category ¨silver¨ and has age > ¨25¨) or (has category ¨gold¨) condition number of(offer for customer)>=1 holds end for

67	add constraint rent_require_licence
	Begin
	for each c: customer
	for each r: rental
	if (c is asked_for r)
	then
	condition(c is has_licence and c is has_age >¨25 ¨)holds
	end if
	end for
	end for
	end

68	--------------

The following tables 3.2 show the formal representation of the chunks and their related business rules.

Table 3.2: Set of Chunks

OC ID	Chunk description	Rules
OC1	Submit customer information	{1-11, 42-46, 65,67}
OC2	Car information	{28-32, 47-49}
OC3	Confirm rent	{4,25-30, 32,48-52, 67}
OC4	Submit payment information	{19, 33-39}
OC5	Change offer information	{15-18, 64}
OC6	Offer information	{13-16, 66}
OC7	Change rent information	{19, 22-24,1}
OC8	…………….	…………

The following table 3.3 shows the audience-chucks organization, where every audience class has a set of chucks that are represented in table 3.2.

Table 33: Set of Audiences and their chunks

Audience	Chunks
Customer	{OC1, OC2, OC6}
Operator	{OC2, OC3, OC5}
………..	………………..

Figure 3.16 bellow, shows the implementation –DB schema- of how the business information model can be stored in a relational database. In other words, the meta model of the business information model.

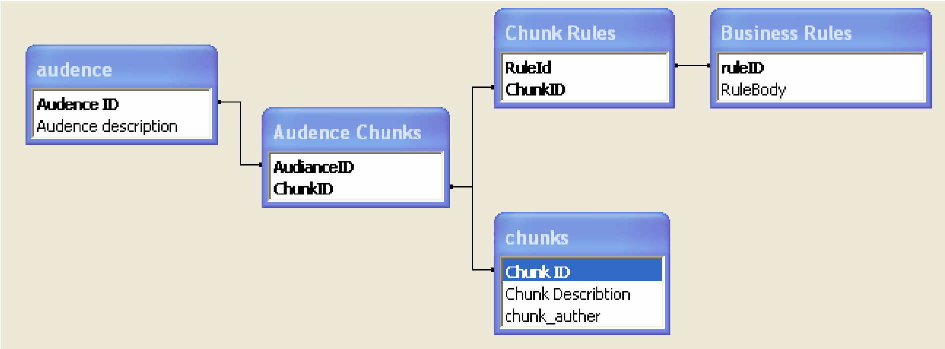


Figure3.16: DB schema: implementation of the business information model

Chapter Four

Conclusion and Future Work

In this thesis we presented an approach for modeling business rules for web engineering, as a contribution to WSDM methodology. In our approach the formulation of the business rules evolves and goes in parallel with the WSDM phases: in the first phase business policies of the enterprise are defined, in the second phase, we address and document the business rules and matches them with the different audiences, in the third phase, business rules are modeled conceptually using the ORM methodology. For some types of business rules we use classical and other ORM constraints, and to model the more complex rules the RIDL language is used. Then, we present an approach for representing, organizing, and integrating the business rules in the Business Information Model of WSDM. We suggested that our integrated business rules model could be used, exactly, for representing and integrating the whole business information model of WSDM.

This approach shows a flexible, reusable, and easy way of modeling web site that takes into consideration all the needs of the customer as well as all of the enterprise rules and policies. It also increases the maintainability and consistency of the enterprise web site. Furthermore this approach increases the tractability between the different phases of the website development life cycle.

As a future work, an executable code, -e.g. HTML forms with Java Script- can be generated automatically from business rules. Moreover, a consistency validation component can be implemented to validate ORM and RIDL constraints.

References

[1] A. Bloesch, T. Halpin ,"ConQuer: A Conceptual Query Language", in: Conceptual Modeling - ER'96 Proceedings, Thalheim B. (ed.), LNCS, Springer Verlag, (1996)

[2] The Business Rules Group, "Defending Business Rules – What are they Really? " rev. 1.3, July, (2000) – available at http:-//www.BusinessRulesGroup.org

[3] The Business Rules Group, "Organizing Business Plans – The standard Model for Business Rule Motivation" rev. 1.0, November 15, (2000)

[4] C. Anderson, A. Levy, D. Weld (1999), "Declarative web-site management with Tiramisu", University of Washington, Department of Computer Science and Engineering.

[5] D. Schwabe and G. Rossi (2001), "The Object-Oriented Hypermedia Design model (OOHDM) ", http://www.telemidia.puc-rio.br/oohdm/oohdm.html

[6] E. Gottesdiener, Guide Business Rules Project, 11/95, "GUIDE International Corporation", available at www.guide.org/ap/apbrules.htm. In Business RULES show power,Promise, President, EBG consulting, Issue of application Development Trends, vol4., no3, (1997).

[7] F. Garzotto, P. Paolini, Schwabe, D.: "HDM - A Model-Based Approach to Hypertext Application Design, in: acm Transactions on Information Systems", January 1993 - Volume 11 No. 1.

[8] Jan Demey, Mustafa Jarrar, Robert Meersman: ¨A Conceptual Markup Language that supports interoperability between Business Rule modeling systems¨, In Proceeding of the Tenth International Conference on Cooperative Information Systems (CoopIS 2002).

[9] G. Mecca, P. Merialdo, P. Atzeni, and V. Crescenzi. "The (Short) ARANEUS Guide to Web-Site Development". In WebDB'99.

[10] Kenji Takahashi, Eugene Liang "Analysis and Design of Web-based Information Systems", NTT Multimedia Communications Laboratories, College ofComputing Georgia Institute of Technology, USA

[11] L. Baresi, F. Garzotto, L. Mainetti, and P. Paolini , "Meta-modeling Techniques Meet Web Application Design Tools", Dipartimento di Elettronica e Informazione - Politecnico di Milano Piazza L. da Vinci 32, I-20133 Milano, Italy

[12] M. Bichler, S. Nusser: "W3DT - The Structured Way of Developing WWW-Sites", Proceedings of the ECIS 1996.

[13] M. Fernandez, D. Florescu, A. Levy, D. Suciu, J. Kang, S. Sudarsky and I. Tatarinov (1999), "Strudel Web-site Management System User's Guide".

[14] O. De Troyer, R. Meersman, F. Ponsaert , "RIDL User Guide", (1984).

[15] O. De Troyer, S. Casteleyn (2001), "The Conference Review System with WSDM", in Proceedings of IWWOST 2001, Valencia, Spain.

[16] O. De Troyer (2000), "Audience-driven Web Design", Vrije Universiteit Brussel, WISE research group, Belgium.

[17] O. De Troyer (2001), "WiSDoM Short description", http://wsdm.vub.ac.be/ShortDescription.htm, Brussels, Belgium.

[18] O. De Troyer (2002), "WSDM Web Site Design Method", used for VUB courses, Vrije Universiteit Brussel, Belgium.

[19] O_2 Technology (1998), "ODMG OQL User Manual", http://www.cis.upenn.edu/~cis550/oql.pdf.

[20] Object Management Group Inc. (2002), "UML Home Page", http://www.uml.org/.

[21] S. Casteleyn, O. De Troyer (2001), "Structuring Web Sites Using Audience Class Hierarchies", Proceedings of DASWIS 2001 workshop (attached to the ER 2001 conference), Yokohama, Japan.

[22] S. Ceri, P. Fraternali and A. Bongio (2000), "Web Modeling language (WebML): a modeling language for designing Web sites", Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy.

[23] Saverio Perugini, Jr., Priya Lakshminarayanan, Naren Ramakrishnan, and Mary Beth Rosson, "Customizing Information Access Without Enumerating Interaction Sequences", Department of Computer Science, Virginia Tech, Blacksburg, VA 24061

[24] T. Halpin, "Business Rules and Object Role Modeling", in: issue of Database Programming & Design, vol. 9, no. 10, pp. 66-72, 1996.

[25] T. Halpin (2001), "Conceptual Schema & Relational Database Design", Second Edition.

[26] T. Halpin, "Information Modeling and Relational Databases", 3rd Ed., Morgan-Kaufmann, 2001.

[27] T. Isakowitz, E.A. Stohr and P. Balasubramanian (1995), "RMM: A Methodology for Structured Hypermedia Design", Communications of the ACM, Vol. 38, No. 8.

[28] V. Markowitz (1996), "The Entity-Relationship (ER) Model".

[29] Visio Corporation (1997), “Visio Technology”, http://www.visio.com/.

[30] Mustafa Jarrar , Robert Meersman , “Formal ontology engineering in the DOGMA approach “ In International Conference on Ontologies, Databases and Applications of SEmantics, ODBASE 2002.

[31] W3C (2000), "Extensible Markup Language (XML) ", http://www.w3.org/XML/.