A Requirements Engineering Approach for Data Modelling of Process-Aware Information Systems*

Jose Luis de la Vara[1], Michel H. Fortuna[2,3], Juan Sánchez[1], Cláudia M.L. Werner[3], and Marcos R.S. Borges[4]

[1] Centro de Investigación en Métodos de Producción de Software, Universidad Politécnica de Valencia, Spain
{jdelavara,jsanchez}@pros.upv.es

[2] Universidade Federal de Juiz de Fora, Brazil
michel.fortuna@ufjf.edu.br

[3] COPPE/Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Brazil
werner@cos.ufrj.br

[4] IM-NCE/Programa de Pós-Graduação em Informática Universidade Federal do Rio de Janeiro, Brazil
mborges@nce.ufrj.br

Abstract. Business process modelling is a common activity when modelling an organization during the requirements engineering process of an information system. It is a must for modelling of process-aware information systems, and it can be considered the main activity. Nonetheless, business process diagrams must be complemented with a data perspective of these systems. This paper presents a requirements engineering approach that provides methodological guidance to meet this need through the integration of two other approaches. First, functional requirements are elicited and specified from business process diagrams. Next, the information flows of these requirements are specified. Finally, data modelling is carried out by following a set of guidelines.

Keywords: Process-aware information system, business process modelling, functional requirement, data modelling, info cases, BPMN.

1 Introduction

Understanding of the application domain is essential for the requirements engineering (RE) process of an information system (IS) for an organization. As a result, the need of organizational modelling has been widely acknowledged (e.g. [2][4][17]). Business process modelling is part of most of the organizational modelling-based approaches, and it is a must for the development of a process-aware information system (PAIS). PAISs manage and execute operational processes involving people, applications,

* This work has been developed with the support of the Spanish Government under the project SESAMO TIN2007-62894 and the program FPU AP2006-02324, and co-financed by FEDER. Marcos Borges was partially sponsored by a grant from CNPq # 304252/2008-5.

W. Abramowicz (Ed.): BIS 2009, LNBIP 21, pp. 133–144, 2009.

and/or information sources on the basis of process models [7], and their characteristics imply that RE approaches for their development must differ from traditional ones [1]. First, detailed process models are necessary in the RE process. Second, new systems must support new ways of (better) running an organization.

Although business process modelling can be considered to be the main activity for PAIS modelling, it is not sufficient to completely model a system. The behavioural perspective of a PAIS that is provided by business processes diagrams (BPD) must be complemented with a data perspective. Both functional requirements (which indicate what the system shall do and can be specified from business processes) and data models (which indicate what the system shall store) must be taken into account.

Functional requirements and data models provide different system perspectives; thus, they complement each other [19]. However, problems may arise when modelling both parts of a system. System analysts may have difficulties modelling data from functional requirements [20], and inconsistencies and contradictions may appear if data models and functional requirements are not properly managed [9].

This paper presents a RE approach (referred to as new approach hereafter) that provides methodological guidance for data modelling of a PAIS through the integration of two other approaches. The first approach is based on business process modelling and system purpose analysis [4][5] (referred to as business process-based approach hereafter), and it focuses on organizational modelling for the elicitation of functional requirements. The second approach, called info cases [8], focuses on the derivation of data models from functional requirements. As is detailed below, both approaches have been modified to make their integration possible.

The main purpose of the new approach is to help system analysts model the data of a PAIS from its functional requirements, which are elicited from the business processes of an organization. The specification of functional requirements and data modelling are integrated so that the problems described above do not arise.

On the one hand, the business process-based approach is well-suited for organizational modelling and elicitation of functional requirements, but it must be extended to address data modelling. On the other hand, the info cases approach is useful for data modelling from functional requirements, but it needs detailed guidance to elicit these requirements from BPDs. As a result, the new approach takes advantage of the features of both approaches and mitigates the weaknesses of their separate use.

The business process-based approach is the result of a collaborative project with the company CARE Technologies (http://www.care-t.com). CARE uses OO-Method [14], which is a methodology for automatic software generation based on data-centred conceptual modelling. Therefore, extending the approach with data modelling is essential in order to properly integrate it in the development process of the company, and the info cases approach has been chosen to meet this need.

The new approach starts from the business processes that an organization wants to execute. The BPDs are analysed to elicit and specify functional requirements of a PAIS, and then the information flows of the system for their execution are specified. Finally, the data of the PAIS are modelled in a domain class diagram (DCD), which is derived from the information flows by following a set of guidelines.

The rest of the paper is organized as follows: section 2 presents background and related work; section 3 describes the new approach; finally, section 4 explains our conclusions and future work.

2 Background and Related Work

This section describes the two RE approaches that have been integrated and presents related work to the new approach.

2.1 The Business Process-Based Approach

The new approach addresses organizational modelling by means of a RE approach [4][5] whose purpose is to avoid three problems of IS development for an organization: lack of understanding of the business and lack of focus on the purpose of the system by system analysts, and miscommunication with stakeholders.

The main characteristics of the approach are: 1) joint use of business process modelling and system purpose analysis; and 2) active stakeholders' participation. The approach consists of three stages, and it is based on the existence of a need or problem in the current organizational environment that could be fulfilled by an IS. The organization will change to fulfil the need, and business processes will be affected.

In the first stage, the current state of the organization for which an IS is going to be developed is modelled by means of a glossary, the business events, a domain data model (in the form of a class diagram in which just domain entities and relationships among them are modelled), the business rules, a role model, and a process map. BPDs are modelled from this information, and stakeholders must validate them.

The organizational need is analyzed during the purpose analysis stage. The aim is to find ways of fulfilling the need by means of the development or modification of an IS, and agree on the effect that the IS may have on the business processes with stakeholders. As a result, to-be BPD elements are labelled according to the IS support that they have, and changes in the business processes may occur. The labels of the BPD elements are: "O", if the element will not be part of the system; "IS", if the element will be controlled by the system; or "U", if the element will be controlled by a user.

Finally, functional requirements are specified by means of task descriptions, which detail the system support to business process tasks in a textual template.

2.2 The Info Cases Approach

In the new approach, data modelling is based on the info cases approach [8]. It provides an integrated model whose purpose is to jointly model use cases and domain (data) models in a single conceptual framework.

The approach has two main principles: 1) adoption of a special abstraction level called Informational Level of Objectives (ILO) to which a use case must belong to; and 2) systematic capture and precise description of the information flows between an IS and its actors. Use cases that conform to these principles are called info cases.

A use case is at the ILO if its realization allows a stakeholder to achieve a goal, which means causing a change of state in the system and/or its environment. When the goal is achieved, the state of the system must be steady, so no rollback to a previous state is necessary even if no other use case is subsequently activated.

The information flows of info cases are specified in a formalism that is capable of capturing the elements of a domain model and of permitting the identification of these elements. This formalism has two parts: a specification of the composition of flows, and a dictionary of elementary items of information.

As a result of belonging to the ILO and of precisely specifying the information flows, info cases capture the elements of a domain model and provide a set of semi-automatic rules for deriving it. According to their proponents, info cases can also increase the uniformity of the domain models produced by different modellers.

2.3 Related Work

The most common practice in the RE approaches that deal with functional requirements and data models is to model classes from uses cases or jointly with them. They are based on mechanisms such as linguistic patterns [6], sequence diagrams [10], activity graphs [11] or consistency guidelines [9]. These approaches are solution-oriented, so they do not properly analyse the application domain, nor do they provide guidance for the elicitation of functional requirements. When compared with the info cases approach, these approaches are more complex because they require the use of more models or models that are less flexible than information flows to obtain data models. In addition, they do not provide a homogeneous abstraction level for functional requirements such as ILO.

All the organizational modelling-based RE approaches that model business processes deal with functional requirements and data models (e.g. EKD [2] and ARIS [17]). However, they do not provide a homogeneous abstraction level for functional requirements, and they lack precise guidance for elicitation and specification of functional requirements and for assurance of consistency and completeness in data models. Some approaches focus on data modelling from BPDs (e.g. [16]), but the models that are obtained are incomplete and guidance for completion is not provided. When compared with the business process-based approach, these approaches do not explain how to improve business processes (which is essential for a PAIS), and they do not focus on system purpose or on communication with stakeholders.

Finally, several works have acknowledged the importance and benefits of a data-centred perspective when modelling business processes. They address issues such as the notion of business artifact [12], product-based workflow design of manufacturing processes [15], detection of data flow anomalies [18], and document-driven workflow systems [21]. The main difference of these approaches with the new one is that they take data into account from a perspective of data flow through tasks rather than from a perspective of information flows between a system and its actors. Also, they do not regard BPDs as a means for understanding the application domain and for the elicitation and specification of functional requirements.

3 Description of the New Approach

This section describes the new approach and explains how the business process-based approach and the info cases approach have been integrated.

Both approaches have been modified to make the integration possible. With regard to the business process-based approach, the BPMN notation [13] and the content of the textual template of task descriptions have been extended, and the granularity of the task descriptions is homogeneous as a result of adopting the ILO. With regard to the info cases approach, the information flows are created on the basis of the domain entities that are used in the task descriptions and the content of their textual templates, and the way of specifying information flows and the rules for the derivation of data models have been adapted.

The new approach (Fig. 1) consists of two stages: elicitation and specification of functional requirements, and data modelling. The first one is divided into modelling of the consecutive flows of to-be BPDs and specification of the task descriptions of a PAIS, whereas the second stage is divided into specification of the information flows of the task descriptions and modelling of the domain class diagram (DCD).

The new approach has been used and initially evaluated in several small/medium-size projects with CARE Technologies. As a case study, an actual rent-a-car company (http://www.rentacar-denia.com) is used. Nonetheless, the complete case study is not explained and just some of the information is used for the description of the new approach. The company is located in a tourist area, and its fleet of cars varies between the summer and the winter seasons. As a result, cars are usually bought at the beginning of the season and sold at the end. Its main activity is car rental, but it involves other activities (car maintenance, extras rental…).

The stages of the new approach and their activities are explained in the following subsections.

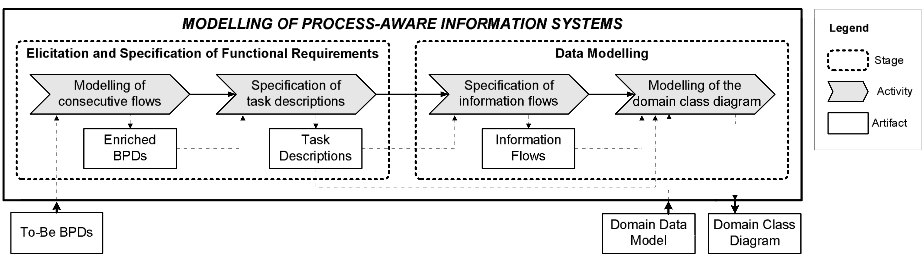

Fig. 1. RE approach for modelling of process-aware information systems

3.1 Elicitation and Specification of Functional Requirements

In the first stage of the new approach, it is necessary to elicit and specify the functional requirements of a PAIS from the business processes that an organization wants to execute according to its needs. Furthermore, these requirements must be part of the ILO so that the two RE approaches are integrated.

Before this stage is developed, the to-be business processes of an organization are modelled from the current business processes and the system purpose, and stakeholders collaborate in the process, as described in [5]. BPDs are completed with the textual specification of the business rules that have not been graphically modelled and a table that specifies the input and output data objects (domain entities in BPMN terminology) of the BPD tasks. All business rules and data objects are not always represented graphically to facilitate the understanding of the BPDs. Finally, BPD elements are labelled and system analysts and stakeholders agree upon the business rules and data objects that will be part of the PAIS. Afterwards, consecutive flows of to-be BPDs are modelled and task descriptions are elicited and specified.

3.1.1 Modelling of Consecutive Flows

To-be BPDs are analysed and enriched graphically by specifying the sequences of flow objects that are executed one after another without an interruption until the business processes have reached a steady state. This activity is necessary to elicit functional requirements that are part of the ILO.

These sequences of flow objects are modelled using a connecting object that is called consecutive flow, which does not exist in BPMN. The aim of this new type of connection is to graphically represent the fact that two flow objects are always executed consecutively. Its graphical representation is an arrow with two arrowheads.

The identification of consecutive flow is carried out as follows. For each sequence flow of a BPD, system analysts have to determine if the target flow object is always executed immediately after the source flow object when a token is in the sequence flow so that the business process reaches a steady state or advances in that direction. If so, both flow objects are linked by means of a consecutive flow.

Stakeholders' participation is essential to develop this activity. Stakeholders are the source of information from which the execution order of the flow objects is modelled, and they must validate that the consecutive flow has been properly modelled according to how the organization executes or wants to execute its business processes.

Fig. 2 shows an example of enriched BPD for the business process "car rental" of the rent-a-car company. Since the example is straightforward, the business process is not explained in great detail.

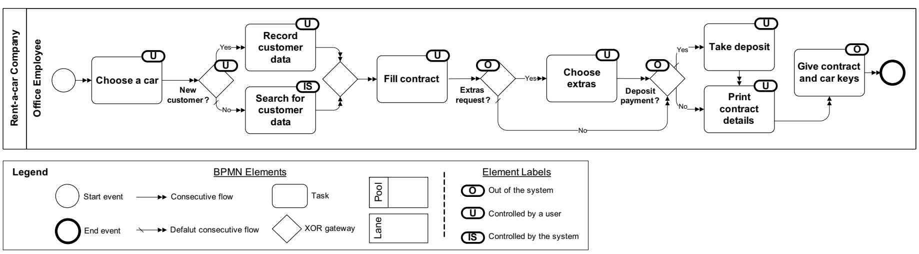

Fig. 2. Example of enriched business process diagram

3.1.2 Specification of Task Descriptions

In the second activity of the first stage, functional requirements are elicited from the enriched BPDs and specified in the form of task descriptions. Three issues must be pointed out. First, functional requirements are called task descriptions because their purpose is to specify adequate support for business process tasks, whereas the purpose of use cases is to specify required usages of a software system or interactions with it. Second, task descriptions must belong to the ILO, and this condition is assured thanks to the analysis of consecutive flows. Third, unlike the business process-based approach, the granularity of BPD tasks and task descriptions may not be the same. A task description can support several BPD tasks.

A task description specifies the support of a PAIS to the execution of a set of consecutive flow objects that denote tasks. These flow objects will be controlled by the same user role or by the system, and they jointly represent a functional requirement that belongs to the ILO. As a consequence, the execution of a task description allows a stakeholder to achieve a goal and the system to reach a steady state.

On the one hand, stakeholders' goals are the execution of the business processes, which allow strategic and operational goals of an organization to be achieved [3]. On the other hand, we consider that the steady states in a PAIS are the same as the ones in the BPDs that will be part of the system and will be determined by those sequences of flow objects that denote tasks that are executed consecutively. For example, the whole business process shown in Fig. 2 corresponds to a task description whose execution will allow the business process and the system to reach a steady state. If the complete sequence of elements is not executed, a steady state will not be reached.

Task descriptions are specified in a textual template that includes the following information: its name; the tasks and the business process that are supported; the role responsible for its execution; the triggers, preconditions and postconditions of the task description; the input and output data objects and their states; an abstract description of the interaction between a user and the PAIS through user intention and system responsibility; the alternatives and extensions to this interaction (which were not taken into account in the business process-based approach); and the business rules that affect the task description. An example of a textual template is not shown, but examples of the previous version of the template (which is identical to the new version except for the alternatives and extensions) are shown in [4] and [5].

Task descriptions are ordered according to their occurrence. Fig. 3 shows the order of the task descriptions that are related to car lifecycle for the rent-a-car company.

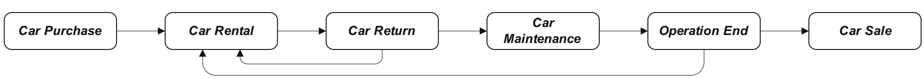

Fig. 3. Example of sequence of task descriptions

3.2 Data Modelling

The second stage of the new approach is data modelling. First, the information flows of each task description are specified on the basis of the domain entities that will be part of the system and the possible interactions that are specified in a task description. Second, a DCD is derived by following a set of guidelines. The DCD contains classes and their attributes, methods and associations.

3.2.1 Specification of Information Flows

System analysts specify the pieces of information that a PAIS and its actors exchange for the execution of its task descriptions by means of information flows. They are specified on the basis of the BNF grammar shown in Fig. 4, which is an adaptation of the way of specifying information flows in the info cases approach. The complete grammar is not shown due to page limitations.

```
<Information flow> ::= <Input flow> | <Output flow> | <Input flow> <Output flow>
<Input flow> ::= → <Data expression>
<Output flow> ::= ← <Data expression>
<Data expression> ::= <Domain entity> | <Domain entity> / <Attribute> /  |
          <Data expression> + <Data  expression> | <Data expression> '|' <Data expression> |
          <Lower limit> { <Data expression> } <Upper limit> | ( <Data expression> ) |
          [ <Data expression> ]
<Attribute> ::= <Attribute name> | <Attribute> + <Attribute> | <Attribute> '|' <Attribute> |
          ( <Attribute> ) | [ <Attribute> ]
```

Fig. 4. Excerpt of the BNF grammar for the specification of information flows

The semantics of the symbols that can appear in an information flow is as follows: the symbols ‘→’ and ‘←’ depict input and output pieces of information to and from a PAIS, respectively; ‘/ /’ depicts membership; ‘+’ depicts composition; ‘|’ depicts alternative; ‘{ }’ depicts repetition; ‘( )’ depicts grouping; and ‘[ ]’ depicts option.

It is essential that the information flows of the task descriptions of a PAIS allow system analysts to completely and correctly identify the elements of a DCD. These elements will be those that are needed for the execution of the task descriptions, and thus for the support of the business processes. In addition, precisely specifying the information flows from a BNF grammar allows the automation of the derivation of a DCD to be possible, although not completely.

Unlike the info cases approach, the new approach carries out the specification of information flows from the domain entities that are used as input and output of task descriptions. Composed, alternative, repeated, grouped and optional elements are based on the normal, alternative and extension interactions of the task description for which an information flow is specified (for brevity, the way to carry out this specification is not explained in detail). Membership elements, which refer to attributes of the domain entities, must be obtained from stakeholders and organizational documentation.

Examples of information flows for the case study are shown in Fig. 6, in which domain entities of the domain data model (Fig. 5) are depicted in bold and in italics.

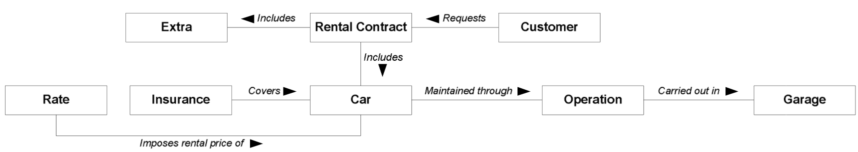

Fig. 5. Example of a domain data model

3.2.2 Modelling of the Domain Class Diagram

At last, the DCD of a PAIS is modelled from the information flows of its task descriptions by following a set of guidelines. The guidelines have been adapted from the rules that are proposed in the info cases approach for the derivation of a domain model. There are 9 guidelines, and they allow system analysts to model the classes of a DCD (guideline 1) and their attributes (guideline 2), methods (guidelines 3, 4, 5 and 6) and associations (guidelines 7, 8 and 9).

Task Description: **CAR PURCHASE**
→ ***Insurance*** + ***Rate*** + ***Car*** / number of plate + model + engine + colour + seats + purchase date + office /
Task Description: **CAR RENTAL**
→ ***Car*** + (***Customer*** | ***Customer*** / number + name + surname + ID number + address + city + telephone number + credit card type + credit card number + credit card expiration date /) + ***Rental Contract*** / contract number + current date + current time + office + return date + return office + [deposit] / + [$_0${ ***Extra*** }$_n$] ← ***Rental Contract*** / contract number + current date + current time + office + return date + return office + rental cost + extras cost + VAT + deposit + total cost / + ***Car*** / model + plate number / + ***Customer*** / name + surname + ID number / + [$_0${ ***Extra*** / name / }$_n$]
Task Description: **CAR RETURN**
→ ***Rental Contract*** / return date / ← ***Rental Contract*** / amount to pay /
Task Description: **CAR MAINTENANCE**
→ ***Car*** + ***Garage*** + ***Operation*** / number + current date + description /
Task Description: **OPERATION END**
→ ***Operation*** / end date + price /
Task Description: **CAR SALE**
→ ***Car*** / sale date /

Fig. 6. Examples of information flows

The guidelines are presented in detail below. We consider that a detailed description is essential given that the DCD of a PAIS is the main outcome of the new approach.

Guideline 1 (classes): A class is modelled in a DCD for each domain entity of an information flow.

For the case study, the classes are "Insurance", "Rate", "Car", "Customer", "Rental Contract", "Extra", "Garage" and "Operation".

Guideline 2 (attributes): An attribute is modelled in a class of a DCD for each attribute that belongs (membership) to the domain entity from which the class was modelled and that is in an input flow; a data type must be specified for each attribute.

For the case study, the attributes of the class "Operation" are "number", "current date", "description", "end date", and "price".

Guideline 3 (creation method): A creation method is modelled for each class of a DCD; its parameters are the attributes of the domain entity from which the class was modelled in the first task description where the domain entity appears; a data type must be specified for each parameter.

For the case study, the creation method of the class "Operation" is "create operation (number, current date, description)".

Guideline 4 (deletion method): A deletion method is modelled in a class of a DCD if: 1) there exists a task description in which the domain entity from which the class was modelled is part of the input flow; 2) the domain entity does not have attributes; 3) the domain entity does not appear in the information flow of any later task description; and 4) the system analyst can confirm that the domain entity is no longer needed.

For the case study, this guideline is not applied.

Guideline 5 (modification method): A modification method is modelled in a class of a DCD for each task description in which the domain entity from which the class was modelled has attributes in an input flow, and the creation method of the class was not modelled from the task description; its parameters are the attributes of the domain entity from which the class was modelled in the task description where the domain entity appears; a data type must be specified for each parameter.

For the case study, a modification method of the class “Operation” is “end operation (end date, price)”.

Guideline 6 (calculation method): A calculation method is modelled in a class of a DCD for each attribute that: 1) belongs to the domain entity from which the class was modelled; 2) is in an output flow; and 3) does not correspond to an attribute of the class; a return data type must be specified for each calculation method.

For the case study, a calculation method of the class “Rental Contract” is “calculate rental cost ()”.

Guideline 7 (associations): An association between two classes of a DCD is modelled if: 1) the domain entities from which the classes were modelled are part of the same input or output flow; and, 2) there exists an association between the entities in the domain data model.

For the case study, the classes “Car” and “Rental Contract” are associated.

Guideline 8 (minimum multiplicity): The minimum multiplicity of a class in an association is 0 if the association is not modelled from the task description from which the creation method of the class was modelled; otherwise, the minimum multiplicity is the minimum number of occurrences of the domain entity from which the class was modelled in the information flow from which the association was modelled (1, 0 if optional, or lower limit of repetitions).

For the case study, the minimum multiplicities of the association “Car – Rental Contract” are 0 for the class “Car” and 1 for the class “Rental Contract”.

Guideline 9 (maximum multiplicity): The maximum multiplicity of a class in an association is the maximum number of occurrences of the domain entity from which the class was modelled in the information flow from which the association was modelled (1 or upper limit of repetitions); the maximum multiplicity could be increased on the basis of business rules.

For the case study, the maximum multiplicities of the association “Car – Rental Contract” are indeterminate (‘*’) for the class “Car” (based on business rules) and 1 for the class “Rental Contract”.

Fig. 7 shows the DCD that has been derived from the information flows shown in Fig. 6 and the domain data model shown in Fig. 5. Note that the information flows are just a part of the whole case study, and thus the DCD is incomplete. In addition, parameters and data types have not been modelled to keep Fig. 7 as small as possible.

Once the guidelines have been presented, two important aspects of a DCD must be pointed out. First, it is evident that a DCD and a domain data model are very similar in the new approach. The classes and associations of the DCD of a PAIS are a subset of the entities and relationships of the domain data model of the organization for which the system will be developed. Nonetheless, we do not consider this fact to be a weakness or problem of the new approach since this is a reflection of common practice in IS development.

The pieces of information (data) that are stored in an IS correspond to a part of the application domain that will be controlled by the system. In the new approach, a domain data model is a part of the application domain that is later analysed and refined

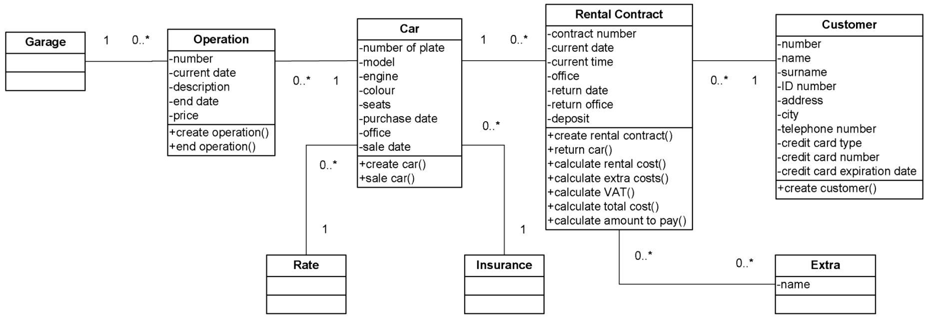

Fig. 7. Example of a domain class diagram

to model a DCD that depicts the part of the domain data that will be controlled by a PAIS. The new approach aims to provide methodological guidance for system analysts in order to model a complete and correct DCD from a domain data model and task descriptions. Furthermore, the purposes of a domain data model and of a DCD are different in the new approach. The purpose of a domain data model is to understand the application domain, whereas the purpose of a DCD is to model the data that will be controlled by a PAIS.

Second, it is known that any class diagram must be complemented with the textual specification of the data constraints that could not be modelled graphically. In the new approach, these constraints are usually specified in the business rules that affect task descriptions, but they might also be specified in a DCD depending on the preferences of system analysts.

4 Conclusions and Future Work

Organizational and business process modelling as a means for understanding the application domain are essential for PAIS development. BPDs play a major role in PAIS modelling, but system analysts must not limit their focus on them alone. They must also take other aspects such as data models into account.

This paper has presented a RE approach that provides methodological guidance to help system analysts model the data of a PAIS. The new approach is based on two other RE approaches whose integration can be regarded as the main contribution of this paper. BPMN has been extended with consecutive flow, task descriptions have been improved though the extension of their textual template and the adoption of a homogeneous abstraction level, and new guidelines for the specification of information flows and for DCD modelling have been presented.

The integration takes advantage of the strong points of the two approaches and also extends them. The business process-based approach now addresses data modelling, and the info cases approach now addresses organizational modelling for the elicitation of functional requirements.

As future work, the development of tool support is planned to facilitate the use of the new approach. A technique for the analysis of non-functional requirements and guidelines for the derivation of the presentation model (user interface) of OO-Method are also necessary. Lastly, the new approach must be applied in more projects in order to further evaluate it and so that improvements might be made. It is important that the new approach is used in large projects and in projects in which a legacy system exists.

Acknowledgements. The authors would like to thank the reviewers for their useful comments and suggestions for the paper.

References

1. Alexander, I., Bider, I., Regev, G.: REBPS 2003: Motivation, Objectives and Overview. Message from the Workshop Organizers. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681. Springer, Heidelberg (2003)
2. Bubenko, J., Persson, A., Stirna, J.: EKD User Guide (2001), `http://www.dsv.su.se/~js`
3. de la Vara, J.L., Sánchez, J.: Business process-driven requirements engineering: a goal-based approach. In: BPMDS 2007, Trondheim, Norway (2007)
4. de la Vara, J.L., Sánchez, J.: Impoving Requirements Analysis through Business Process Modelling: A Participative Approach. In: Abramowicz, W., Fensel, D. (eds.) BIS 2008. LNBIP, vol. 7, pp. 165–176. Springer, Heidelberg (2008)
5. de la Vara, J.L., Sánchez, J., Pastor, Ò.: Business process modelling and purpose analysis for requirements analysis of information systems. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 213–227. Springer, Heidelberg (2008)
6. Díaz, I., Sánchez, J., Matteo, A.: Conceptual Modelling Based on Transformation Linguistic Patterns. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 192–208. Springer, Heidelberg (2005)
7. Dumas, M., van der Aalst, W., ter Hofstede, A. (eds.): Process-Aware Information Systems. Wiley, Chichester (2005)
8. Fortuna, M.H., Werner, C.M.L., Borges, M.R.S.: Info Cases: Integrating Use Cases and Domain Models. In: RE 2008, Barcelona, Spain (2008)
9. Glinz, M.: A Lightweight Approach to Consistency of Scenarios and Class Models. In: ICRE 2000, Schaumburg, USA (2000)
10. Insfran, E., Pastor, Ó., Wieringa, R.: Requirements Engineering-Based Conceptual Modelling. Requirements Engineering 7(2), 61–72 (2002)
11. Kösters, G., Six, H.W., Winter, M.: Coupling Use Case and Class Models as a Means for Validation and Verification of Requirements Specifications. Requirements Engineering 6(1), 3–17 (2001)
12. Nigam, A., Caswell, N.S.: Business artifacts: An approach to operational specification. IBM Systems Journal 42(3), 428–445 (2003)
13. OMG: Business Process Modelling Notation (BPMN) Specification v1.1, (2008), `http://www.bpmn.org`
14. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice. Springer, Heidelberg (2007)
15. Reijers, H.A., Liman, S., van der Aalst, W.M.P.: Product-Based Workflow Design. Journal of Manegement Information Systems 20(1), 229–262 (2003)
16. Rodriguez, A., Fernández-Medina, E., Piattini, M.: CIM to PIM Transformation: A Reality. In: CONFENIS 2007, Pekin, China (2007)
17. Scheer, A.W.: ARIS - Business Process Modeling, 3rd edn. Springer, Heidelberg (2000)
18. Shun, S.X., Zhao, J.L., Numaker, J.F.: Formulating the Data-Flow Perspective for Business Process Management. Information Systems Research 17(4), 374–391 (2006)
19. Siau, K., Lee, L.: Are use case and class diagram complementary in requirements analysis? Requirements Engineering 9(4), 229–237 (2004)
20. Svetinovic, D., Berry, D.M., Godfrey, M.: Concept Identification in Object-Oriented Domain Analysis: Why some Students Just Don’t Get It. In: RE 2005, Paris, France (2005)
21. Wang, J., Kumar, A.: A Framework for Document-Driven Workflow Systems. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 285–301. Springer, Heidelberg (2005)