A software model of the Enterprise and its Environment

Henk de Swaan Arons
Marc Jellema
Erasmus University Rotterdam
Dept. of Computer Science
PO Box 1738, H4-21
3000 DR Rotterdam, The Netherlands
E-mail: deswaanarons@few.eur.nl, hjellema@worldaccess.nl

Abstract

One of the major aims of ongoing research at Erasmus University Rotterdam is to design and develop an expert system that will be able to analyze the strength of an enterprise and to advise about the strategies this enterprise must follow. A part of this research concerns the design of an object-oriented (OO) model of an enterprise and its environment. Such a model can be used as framework for expert system design and development. In this paper such an OO model is presented and related issues are discussed such as which object classes will have to be selected and how they are interrelated. For this OO model design Rumbaugh's Object Modeling Technique has been used. Some attention is paid to the implementation of this model in Aion DS, aan OO knowledge representation tool.

Keywords: strategic analysis, object-oriented modeling, OMT

1. Introduction

In the last decade object orientation (OO) has become extremely popular. Particularly in programming this trend has led to several well-known OO programming languages such as C++, Visual C++ and Delphi. The main advantages are the improved reusability and maintainability of software. The availability of appropriate notation techniques could further enhance its portability. OO methods have not only advantages in programming. Also in modeling problem domains for simulation and knowledge system development purposes OO is very useful. Then it can provide a more consistent and readable syntax. This advantage is even more emphasized when an appropriate modeling technique is used such as Rumbaugh's Object Modeling technique [6]. Introducing OO in modeling has not only the advantages concerning reusability and maintainability but also could it improve the model's quality.

The main objective of this article is to design a detailed and coherent OO model of an (industrial) enterprise and its environment [2]. This environment consists of components such as the branch of industry to which the enterprise belongs, its competitors, suppliers and its customers. This research study is one component of a broader study in which it is aimed to design and develop an expert system that is able to advise on the strategy the enterprise has to follow. Some of these kinds of knowledge systems are already available but they suffer from some conceptual drawbacks that can be redirected to the model on which it is based.

In the next section some general remarks concerning OO modeling are made and some details of Rumbaugh's Object Modeling Technique relevant to this article are presented. After these introductory remarks the OO model of an (industrial) enterprise and its environment are discussed. As an example of the usefulness this OO model, its implementation in Aion DS (a development tool for knowledge systems) [2] is discussed.

In a sense the results of the expert system are different from those generated by similar expert systems heavily leaning on general expert knowledge. Business Insight, for example, generates a report on the strategic performance of a business, based on the response to about 400 questions and a few hundred production rules of expert knowledge. Since this last system is widely recognized as a good step towards using decision support tools, this system can very well be used as a benchmark to measure the usefulness of the expert system.

2. OO modeling and OMT

In the context of this paper we restrict ourselves to discuss those aspects of OO modeling and OMT that are relevant to understanding the subject. Important notions of OO are - of course - the object, that can loosely be defined as a representation of some coherent set of data and processes in some problem domain. Such an object mostly is a member of class of objects and is sometimes referred to as an instance. An object is described by a number of attributes (sometimes called slots). Crucial to OO modeling is its capability to represent the various relations between objects, such as aggregation and inheritance. An example can be found in a submodel of the enterprise, see figure 1.

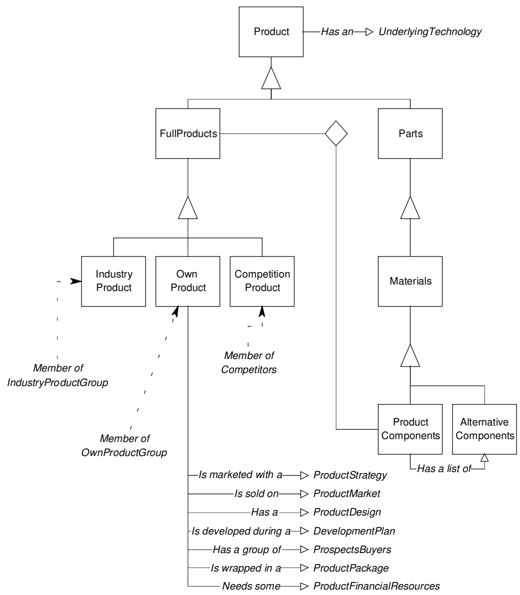

Figure 1*. The class structure for Product*

The class of objects Product has two subclasses, FullProduct and Parts, inheriting all attributes, methods and messages from *Product*. The subclass *FullProduct* itself has again three subclasses, *IndustryProduct*, *OwnProduct* and *CompetitionProduct*. It can also be noticed that *FullProduct* consists of a number of *ProductComponents*, which is a subclass of *Materials*, etc. In the figure there are links to other object classes: *Product* has a link to *UnderlyingTechnology*, and for example, *IndustryProduct* is a member of the *IndustryProductGroup.* When a domain has to be modeled in an object-oriented way, it is not only important to be able to identify the relevant objects with their attributes and relations. It is equally crucial to have a consistent notation technique available. In [6] extensive attention has been paid to object modeling and to OMT, the Object Modeling Technique and to the concepts on which this notation is based (OMT distinguishes between static and dynamic models, but in this context the static will be adequate). A few of the most relevant OMT notations are given in figure 2. Now, the diagram given in figure 1 can be better understood.

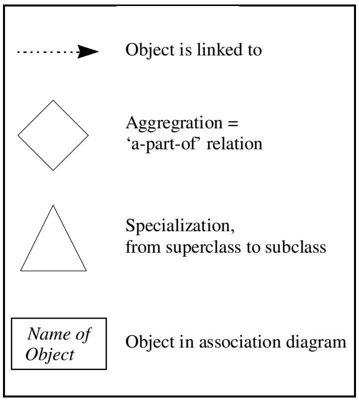

Figure 2. *Some relevant notations in OMT*

Although this object model has been designed to serve as the backbone of an expert system on strategic analysis, the power of this approach is that it also could be employed for completely other purposes. It can be used for simulation where the objectives could be different. For example, a more or less similar model can be used as a part of a Decision Support System (DSS). Naturally, for this it is required that the simulation package to be used supports OO modeling in a comparable way as, for example, the knowledge system tool Aion DS does.

3. OO model of enterprise and environment

It has been stated before that the OO model under study is meant to be used as the basis of an expert system on strategic analysis. There are commercial systems of the kind available, but none of these is really satisfactory. First the subject of strategic analysis is one of the expert fields that are not sufficiently understood and in this sense a lot of work can still be done in this field of research. Furthermore, the systems that are available have various restrictions. The above mentioned expert system aims to come to a better and more complete business tool.

One such a system in this category is Business Insight [1] that according to its tutorial, is *'A business factors analysis tool that is used to gather knowledge and formulate strategies for business planning'*. In figure 3 an overview of the functionality of Business Insight is given. Data concerning all kinds of aspects is input (*Data Input*). Altogether about 400 questions have to be answered subdivided in 8 categories: Enterprise Information, Product or Product Line, Market definition, Marketing and sales, Competition, Production, Suppliers and Financials, see figure 4 as an example to input the value for the attribute *Commitment To Industry*.

After this tedious though necessary task has been completed there are several options possible. One is the function *Summary provides critique* that provides an analysis of the company with its pros and cons based on the input data and with hypertext links to expert rules back to the data. There are several other functions, for example to carry out what-if analyses, to provide a summary of high and low lights, inconsistency checking, etc. These are quite often illustrated with useful presentations such as tables, charts or graphs. Business Insight's functionality is based on several strategic models, but especially on the one of Porter [4]. This model basically distinguishes three strategies: Differentiation Strategy (e.g. a unique product), Cost Leadership Strategy (e.g. emphasis on low costs and cost control) and Focus Strategy (a particular part of the market).

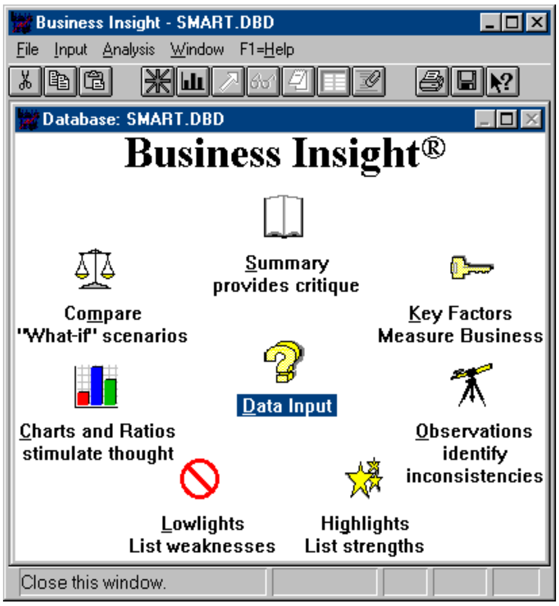

Figure 3. *Overview of the functionality of Business Insight*

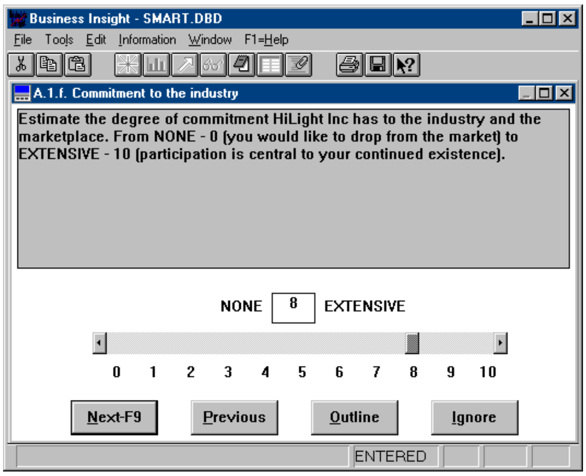

Figure 4. *Data input for the variable Commitment to Industry*

Business Insight's expert tasks involve a reasoning model that combines input data and expert rules based on these models, ultimately leading to an analysis and advice. Although the expert rules are not explicitly provided to the user, with some efforts a few hundred of such rules can be detected. One such expert rule is the following:

The product is very advertizable if:

1. product is differentiated (58,10);
2. product has strong emotional appeals (60,6) (cf. B.3.1);
3. product form is differentiable (64,10);
4. market is in early stages of its life cycle (75,10) (cf. B.1.c);
5. decision maker faces minimal personal risk (80,3) (cf. C.4.a);

6. product offers obvious advantage (89,10).

The assertions 1, 3 and 6 are concluded by other expert rules (forward chaining), those in 2, 4 and 5 refer to the data earlier input by the user. The first of the two numbers associated to an assertion indicates how true it is considered in this enterprise, the second number how relevant this assertion is to the conclusion to be drawn (weight factor).

It is beyond the scope of this article to discuss Business Insight in more depth. However, it has some severe restrictions. The analysis can only be achieved:

- for only one enterprise at a time.
- for an enterprise carrying only one product;

Furthermore, the environment of the enterprise is restricted to competitors, suppliers, prospects and the kind, but the branch of industry is not considered.

It is our opinion that a better analysis will be possible if such a system allows an enterprise to carry more than one product at a time, to consider the synergetic effects of several Strategic Business Units (SBUs) and to take into account the effects due to the environment. For this reason it was decided to develop and design a new system that could also cope with these aspects. It was considered to be the best choice to have an OO model to serve as a backbone for this new system. The reasons for this are the same as for the use of OO modeling in general: better reusability and maintainability of the final product. However, it is also expected that such an approach would facilitate the description of the field. In doing so, most of the variables used in Business Insight have been incorporated as attributes (also called slots) in the OO model, be it that their place in the model quite often differs from their original ones.

The final model is too large to be discussed in full detail. It consists of more than 110 classes, almost 100 instances and more than 700 slots. Globally it consists of 17 separate groups of object classes, i.e. a group contains a number of object classes that are closely related to each other. Important kinds of relations are inheritance and aggregation. Other types of relations (links) are defined between the various groups. One such group is depicted in figure 1, which describes the relevant relations concerning the object class *Product*. The object classes used are given in the table 1.

1. *Analysis*	7. *Group*	13. *Program*
2. *Design*	8. *Market*	14. *Resources*
3. *Enterprise*	9. *Package*	15. *Strategy*
4. *Enterprise Process*	10. *Plan*	16. *Technology*
5. *Equipment*	11. *Person*	17. *Industry and Product Market Combination*
6. *Facilities*	12. *Product*	

Table 1. *Groups of object classes*

Analysis has two subclasses *BusinessFactorAnalysis* and *AnnualAnalysis* and concern the financial analysis of the enterprise. The group *Design* concerns the product design; the group *Enterprise* contains the information of the enterprise itself (*OwnEnterprise*), of *Competitor*, *Supplier*, *Distributor*, i.e. all kinds of enterprise. The group *EnterpriseProcess* describes the manufacturing process, the group *Equipment* the various types of equipment, such as *ManufacturingEquipment* and *DevelopmentEquipment* as subclasses of *Equipment*; the group *Facilities* contains the object class *Facilities* with *ServiceFacilities*, *SalesFacilities* and *ManufacturingFacilities* as subclasses; the group *Group* exists of all kinds of groups such as *ManagementTeam*, *SalesPersonnel, ProductBuyers, Employees,* etc. Some of these are a list of objects, such as *Employees*, being a list of *Employee*, others are only used as group such as *ProductBuyers*. The group *Market* is also quite extensive, with classes such as *Market* wit subclasses *ConsumerGoodsMarket* and *ProducerGoodsMarket*, each of which with their own subclasses (e.g. *ConvenienceGoodsMarket*, *SuppliesMarket*, etc.). The group *Package* deals with the packaging of the product, *Plan* with the development of the product. The group *Person* describes the object class *Person*, its subclasses *Employee* and *Founder*, the first of which has the subclass *Manager* that on its turn has subclasses such as *ChiefOperatingOfficer*, *ChiefFinancialOfficer*, etc. The group *Program* defines all kinds of programs such as its subclasses *EmployeesBenefitprogram*, *DevelopmentProgram*, *ProductSalesProgram*, etc., the group *Resources* concerns *FinancialResources* with subclasses *ProductStartupCapital* and *ProductFinancialResources*. The latter is a part of *ProductStartupCapital*. The group *Strategy* differs between *LongtermStrategy* and *ProductStrategy*. As expected from Porter's model *LongtermStrategy* is subdivided in the subclasses *CostLeadership*, *Differentiation* and *Focus*. Finally, the group *Technology* concerns *ProductTechnology* and *Industrytechnolog; Industry&ProductMarketCombination* deals with the environment.

4. Implementation in Aion DS

The original objective of this research has been the design and development of an expert system with more extensive concepts than Business Insight has. As a development tool Aion DS has been selected. It offers extensive possibilities: production rules, an inference engine, an advanced user interface, and above all, it fully supports OO modeling [4]. The OO model as presented earlier can be implemented without difficulty. It supports classes of objects, both static and dynamic instances, attributes with relevant properties (such as name, type, default, explain functions), methods and messages.

As an example, figure 5 shows, among other things, the representation of the earlier mentioned *Product* class structure. The first crucial OO relation, that is inheritance, is accomplished in Aion DS by supplying each object with a so-called owner: each object inherits all the slots and methods of its owing class structure. Top level class structures like *Product* are the exception to this rule: they do not have an owner. The fact that subclasses inherit the slots from their superclasses can be seen in figure 5, because all the slots have their respective owner specified (see the top right window named *Slots*): a *Product* has a *ProductCode* because it is a subclass of *Product*. Aggregation, the second crucial OO relation, is implemented as a list of pointers to the object classes that are aggregated. Figure 5 shows (bottom left window) this kind of relationship through the fact that the members of an *OwnProductGroup* are specified as pointers to *OwnProduct* objects. Other relations between objects should, according to Rumbaugh, be modeled as separate so called 'relation objects'. One of the main reasons behind this concept is the fact that neither of the objects that are related in fact own the relation: giving one of these objects a pointer slot to the other can be seen as incorrect modeling of the real world. On the other hand, a model that consists of a combinations of relation and real world objects, can be less straightforward to comprehend. Due to this last remark, the current OO model models ordinary relations between objects as pointers. An example is shown in figure 5 as the *Product*'s slot *ProductStrategy* is a pointer to a *ProductStrategy* object.

5. Results, conclusions and further research

The main objective of this study has been to design and develop a detailed and coherent OO model of an (industrial enterprise and its environment (among other things, Branch of Industry, Competitors, etc.). Although it has not been an aim on itself, the model had to be as independent as possible from the actual purpose of its development.

The authors are convinced that this model could also be used in a study with simulation purposes. Naturally, because of its changed objectives, some slight modifications will be necessary. A major requirement is that the simulation package to be used will be suited for implementing OO models.

The OO model has shown to be adequate in an expert system that performs a strategic analysis of an enterprise and give advice. Although this research study is still going on, now already it can be stated that it can serve as a real backbone of the system as a whole.

Naturally, the OO model presented in this paper can easily be extended. Especially, the synergetic effects of SBUs could be worked out in more detail. This can be done easily without disturbing the existing model.

6. Literature and references

[1] Business Insight® for Windows, a Tutorial, Business Resource Software, Inc.

[2] Hellriegel, D., Slocum, J.W. Jr., Management - 5th, Addison-Wesley Publishing Company, 1989.

[3] Mulford, R., Stipp, L., Developing Applications with AionDS - Student Guide, Version 1.0, Trinzic Corporation, 1994.

[4] Payne, E.C., McArthur, R.C., Developing Expert Systems: A Knowledge Engineer's Handbook for Rules & Objects, John Wiley & Sons, Inc., 1990.

[5] Porter, M.E., Competitive advantage: creating and sustaining superior performance, The Free Press, London: Collier Macmillan, cop., 1985.

[6] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., Object-Oriented Modeling and Design, Prentice-Hall International, Inc., 1991.

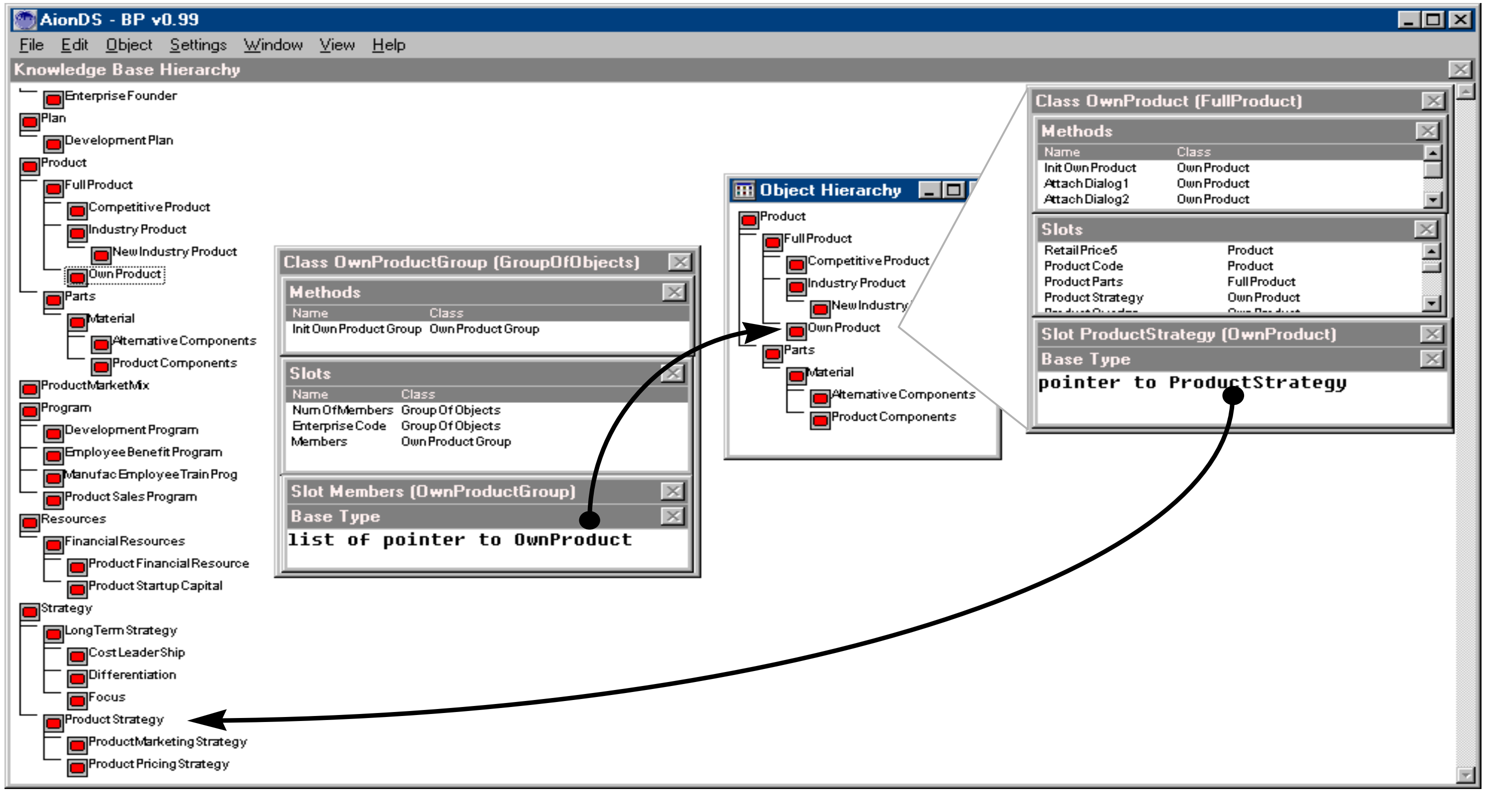

Figure 5. *The class structure for Product implemented in Aion DS*