Object Oriented Analysis and Programming for a Working Capital Management System[1]

Rosina Weber Lee[*] Lori Viali[*] Roberto Pacheco[*]
Alejandro Martins[*] Ricardo M. Barcia[*] Suresh Khator[**]

Abstract

The main purpose of this paper is to present an Object Oriented Analysis (OOA) of a firm and its accounting and financial environments for the implementation of a working capital management system. The object oriented analysis has been designed so that it can be used by different types and sizes of companies, (e.g., industrial, commercial or service). This versatility is a consequence of an important feature of the object oriented paradigm: the reusability of code. This OOA includes firm's regular operations as well as tools and reports used in the management of working capital such as cash flows and estimated balance sheets. In order to demonstrate the functionality of the OOA, we discuss parts of the analysis that we have implemented successfully in the C++ object oriented programming language.

Keywords: *Working Capital Management; Object Oriented Programming, Accounting, C++.*

Introduction

Working Capital (WC) management involves decisions concerning current accounts affecting the short term management of the firm. These decisions are taken at strategic levels (e.g., target inventory turnover) and at operational levels (e.g., how to invest the excess cash). In order to align these decisions with other policies of the firm, it is necessary to process a large number of variables and evaluate lots of data using expert knowledge in a short period of time. These features represent an ideal task to be performed by a computer; since it can manage a large amount of data very rapidly and can consider a large number of variables in different scenarios in a very short period of time. All these factors lead us to a knowledge based, connectionist or intelligent hybrid system.

The flexibility of the integration between accounting and decision modules is important when the aim is building an intelligent system capable to deal with WC management tasks. This object oriented analysis accomplishes this integration. Building an intelligent system becomes easier by applying the object oriented paradigm since it provides appropriate mechanisms for knowledge acquisition and knowledge representation. Some advantages of the object oriented paradigm to Financial Accounting are [4]: it provides an easy mean to model the mixture of generalization and instance-of relationships of financial systems; it is versatile in modeling the complex structures of financial accounting; and it does not separate financial procedures from financial structures (which is similar to what an accountant does).

After describing the OOA we discuss briefly the implementation of part of this analysis. A simple but successful operation shows the right way of doing things.

Working Capital Management

WC management decisions are taken in order to ensure the goal of maximizing firm's value. One of the most important aspects of WC management is related to the firm's capacity of meeting its liabilities. Every firm has a target liquidity since there are opportunity costs involved and this affects its profitability.

[1] Presented at CIFEr IEEE/IAFE Computational Intelligence on Financial Engineering, Manhattan, New York City, 9 a 11 de abril, 1995.

[*] Federal University of Santa Catarina - Department of Industrial Engineering and Systems - Trindade, Caixa Postal 476, CEP 88049, Florianópolis, SC - (financial support by CNPq) - BRAZIL.

[**] University of South Florida - Department of Industrial and Management Systems Engineering - 4202 East Fowler Av. ENB 118, Tampa, Florida 33620-5350

When facing high levels of inflation or interest rates, WC management can decide the feasibility of a business. Avoidance of idle money would be the first and obvious result of a good WC management. A firm purchasing on credit and selling only in cash can operate without its own money. Then WC becomes a source of funds. On the other hand, if an industry takes long time to manufacture its goods, WC may only garner funds compromising the firm's short term liquidity or long term investments.

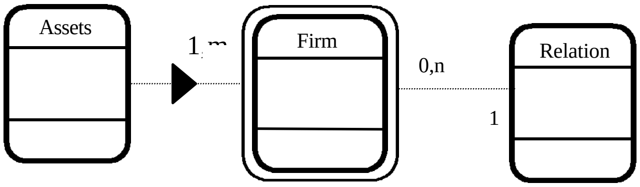
Hence, improving WC management boosts the firm's value and any firm can reach this goal simply by using a computational system that embodies financial information and expert knowledge to manage WC.

Object Oriented Analysis

Developing an object oriented analysis consists in detailing a problem space and abstracting the entities of the domain. Object oriented analysis organizes a domain that will be eventually programmed in an object oriented language. Only in the past few years we could see the emergence of object oriented analysis methodologies [9]. Similar to structured analysis, the goal of OOA is to develop a consistent and accurate representation of the problem domain. The difference is that OOA does not separate data from procedures. Among more than twenty technologies that came out in past few years, we find three of them being broadly used: Coad and Yourdon [6], Booch [3] and Shlaer and Mellor [18]. Here we represent hierarchies using Coad and Yourdan methodology, though many concepts are based on Booch.

According to Coad and Yourdan [6], an object oriented analysis consists of identifying five major layers in the problem: subjects, objects, structures, attributes and services. First we identify subjects by grouping entities. Then, we find the objects generated from classes. Later, we define the structures (such as classification and assembly), attributes, and services. This process is the result of each individual's abstraction of the problem.

The abstraction in the present problem space arises from the idea of a generic firm that can be further specialized into specific ones. There are three basic entities that compose this abstraction: firm, relations and assets. The problem domain is composed by operations that take place in a firm. We have included regular operations like sales, purchases, hiring, tax paying, etc.... Items such as financial and economic indicators that somehow interfere in the firm's business are also included. Other capability of the system is the generation of financial reports.



Our conclusive abstraction in modeling this problem is that every act or operation of the firm represents a relation with another entity. Thus, each relation is the basis of an operation and it has an accounting entry. Since it is the origin of those relations, the firm was abstracted as an entity. That lead us to define the first class **Firm**. The class **Firm** is part of the first layer of abstraction that includes two other classes: Class **Assets** and Class **Relation,** shown in Figure 1.

Classes **Relation** and **Assets** are both roots of hierarchic structures, each one representing a subject (from now on referred to as *subject assets* and *subject relation*). Class **Firm** is linked to objects in the *subject relation* by instance connections [6]. The objects of the *subject assets* and class **Firm** have whole-part connections, representing an assembly structure. In Figure 1, these relations are represented by dotted lines. We use this representation to emphasize that the actual connections happen between an object of **Firm** and objects instantiated within *subject assets* or *relation*. This means that the firm has different kinds of relations (e.g., sales and purchase), each one being an object of a specific class (in the *subject relation*). The same happens in the hierarchy represented by *subject assets*.

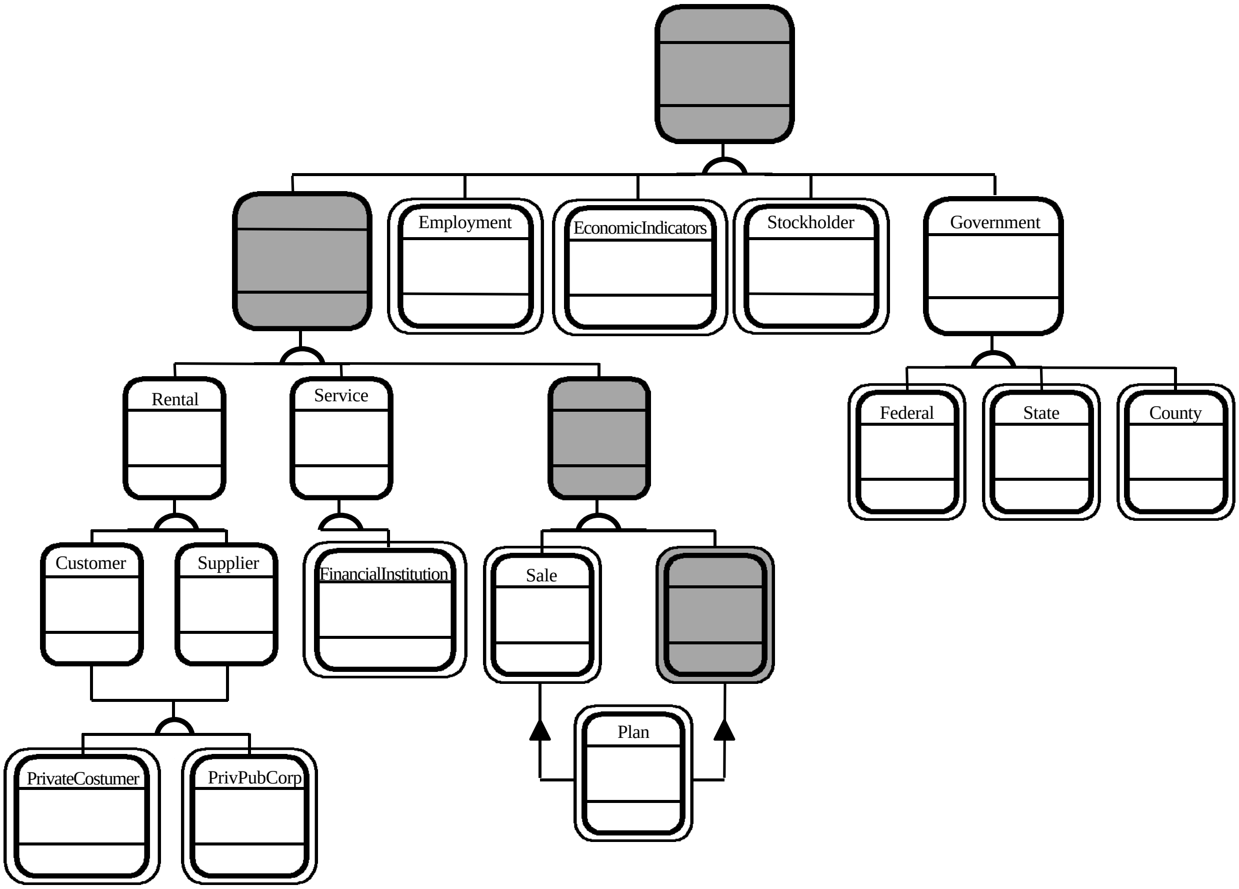


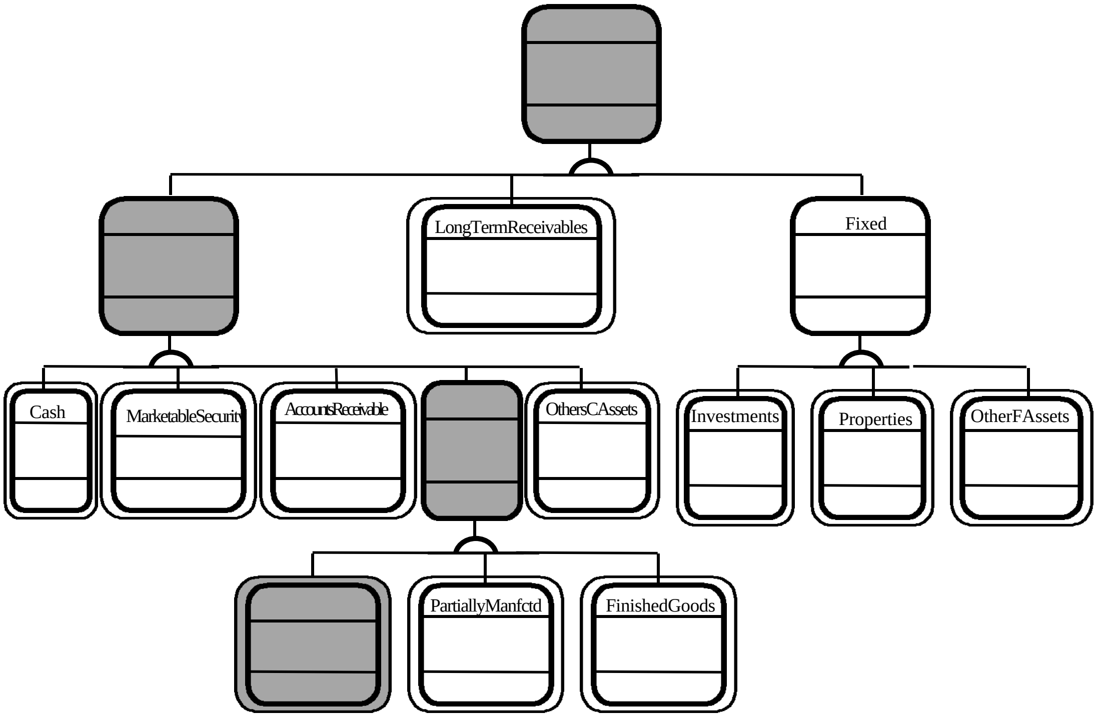
Figure 2 presents the class **Relation** hierarchy. There are five basic relations abstracted in the following classes: **Commercial**, **Employment**, **Economic Indicators**, **Stockholder,** and **Government**. Every operation engaged by the firm is associated to one of the areas modeled by the classes. According to this abstraction, an operation of the firm is represented by an object generated at the end of each branch in the hierarchy (i.e., objects of classes drawn by double-lined boxes). The level where the objects are created depends on the complexity of each area. For instance, in a credit purchase of raw material we identify a commercial relation related to goods. Hence, to generate an object to represent this relation, one has to follow the path **Relation-Commercial-Goods-Purchase** (see Figure 2). Once this operation is completed, two tasks are done: crediting Accounts Payable and debiting Inventories. Crediting Accounts Payable is performed by an entry within *subject relation* while debit Inventories takes place within *subject assets*.

Figure 3 shows *subject assets* (assets hierarchy). The *subject assets* works similarly to accounting. When an operation causes an entry in assets, the class where the operation was generated sends a message to perform such input. In our example, the debit entry in inventories is accomplished by generating an object in the path **Assets-Current-Inventory-RawMaterial** (see Figure 3). Since every firm's operation is a relation, the classes in *subject assets* never generate any entry by itself. Even in an asset operation like sale of ownership certificates, there is a message coming from the relation with stockholders responsible for the entries.

Liabilities are modeled within *subject relation***.** Classes in this subject have methods that send messages to proper classes in *subject assets***,** whenever an entry is needed. On the other hand, when the entry needed is not related to an asset, the information is kept as an attribute of the object of the class that represents the relation. Considering again the credit purchase of raw material, the entry for crediting accounts payable remains as an attribute of the object generated in the path **Relation-Commercial-Goods-Purchase** (see Figure 2). Whenever entries are needed as to create reports like a weekly balance sheet, the total of all entries (e.g., accumulated accounts payable) is then used.

Therefore, all asset entries are kept within the *subject assets* while liabilities entries remain within the *subject relation*.

The object oriented analysis presented shows the primary subjects of our abstraction. They are the essential part of the object oriented system developed. The abstraction pursues an easy and fast way for dealing with the WC and accounting environments under the object oriented paradigm. Despite not being detailed here, there is a quite long list of attributes and methods of the classes defined. Also, the system implements several relationships between classes that are not shown here due to space limitation.



Programming

We have implemented a part of the object oriented analysis using Borland C++ in order to demonstrate the functionality of the abstraction chosen. The system offers a sample firm for demonstration purposes. The implementation is focused on sales operations. The user may choose one of the operations offered by the system (e.g., purchase, sales, service, reports, etc....) and the operation chosen is sales (*Sales*). The program generates an object sales and calls three services: *begin*, *show* and *enter*.

At this point we are able to demonstrate the usefulness of a significant concept of the object oriented paradigm. Polymorphism is the technique used to define the three services above. When the user chooses the operation *Sales*, an object of the class **Sale** is created. Then, the system calls service *enter* which asks the user about details and informs the system. The second service named *show* acts as a bureau officer numbering the sales process and trying to confirm this operation. When the process is confirmed by the user, the system calls service *enter*. This service performs two steps, one related to the amount of inventories and other representing their value. Thus, first the amount of items sold is taken off inventories. Then, the entries of cash, bank or accounts receivable and inventories are made.

Conclusions

We have presented an OOA of a firm and its accounting environment. This OOA can be used in a system to accomplish the task of managing WC. This model has been focused generically in order to keep its portability to any kind of firm. The implementation of the OOA demonstrates the feasibility of the use of this paradigm in a WC management system.

The system presented here can be the basis for future implementations on several companies. The adaptability to specific cases is simple due to the flexibility and versatility of object oriented systems. The changes are basically deletions and additions of classes describing peculiarities of the firm, but the main structure of the system remains the same. Another advantage of this paradigm is easy maintenance of the system.

In this paper we showed the use of the object oriented paradigm by proposing an abstraction. This abstraction models the accounting and WC environments of a firm, representing an appropriate basis to build decision support systems. We hope this abstraction helps defining the best way of modeling this domain for future applications.

The next step is to improve the system by incorporating expertise in WC management providing to the firm a tool to support the financial decisions related to its short term. The authors are currently engaged in developing such a tool showing promising results (e.g., [16]). Also, other further development concerns to the appropriate database to be linked to the present system.

Bibliography

[1] Arcelus, F.J. & Srinivasan, G., "Integrating working capital decisions," *Engineering Economist*; Vol 39, No. 1, pp. 1-15, Fall - 1993.

[2] Bansal, Arun; Kauffman, Robert J. ; Mark, Robert M. & Peters, Edward," Financial Risk and Financial Risk Management Technology (RMT)," *Information & Management*; Vol. 24, No. 5, p267(15), May 1993.

[3] Booch,Grady, *Object Oriented Design with Applications,* The Benjamin Cummings Pub. Firm., Inc., 1991.

[4] Chu, Pai-Cheng, "Applying Object-Oriented Concepts to Developing Financial Systems," *Journal of Systems Management*. USA, Vol. 43, No. 5, pp. 28, 33-4, May 1992.

[5] Cloud, Phil ," Understand Working-Capital Needs," *The Business Journal*; Vol. 16,No. 21, pp. 18., March 1994.

[6] Coad, Peter. Yourdon, Edward , *Object-oriented Analysis,* Prentice-Hall, Inc. New Jersey, 1990.

[7] Dempster, M. A. H. & Ireland, A. M.," Object-oriented model integration in a financial decision support system," *Decision Support Systems* - Vol. 7, No. 4, pp. 329-340, Nov ember - 1991.

[8] Fazzari, Steven M. & Petersen, Bruce C.," Working capital and fixed investment: new evidence on financing constraints," *RAND Journal of Economics*; Vol. 24, No. 3, pp. 328(15), Autumn 1993.

[9] Fichman, Robert G., Kemerer, Chris F.,"Object-Oriented and Conventional Analysis and Design Methodologies," Computer. USA, v. 25, n. 10, pp. 22-39, Oct. 1992.

[10] Gallinger, G. W. & Healey P. B. ,*Liquidity Analysis and Management*, 2 nd. Edition - Addison - Wesley Publishing Company, Inc., 1991

[11] Gianotti, Claudio, Pau, Louis F., *Economic and Financial Knowledge-Based Processing,* Springer-Verlag. Berlin, Heidelberg, 1990.

[12] Kaindl, Hermann,"Object-oriented approaches in software engineering and artificial intelligence", *Journal of Object-Oriented Programming*, Vol.6, No.8, p.38-45, January, 1994.

[13] Kolay, M. K ," Managing working capital crises: a system dynamics approach ," *Management Decision*;, v29, n5, p46(7), September 1991.

[14] Metcalf, Jerry Gross margin return on working capital: a project management technique, "*Industrial Management*; Vol. 32, No. 4, pp.27(3), July-August 1990.

[15] Montenegro, F. & Pacheco, R., *Orientação a Objetos em C++*, Editora Campus, Brasil - 1994.

[16] Pacheco, R.; Martins, A., Weber Lee, R.; Barcia, R. and Khator, S., "A Hybrid Intelligent System to Diagnose and Indicate Solutions to Financial Problems", Technical Report.

[17] Parkinson, K. L. & Kallberg, Jarl G.,*Corporate Liquidity, A Guide to Working Capital Management*" Homewood, IL, Richard D. Irwin, 1993.

[18] Shlaer, Sally, Mellor, Stephen J., *Análise de Sistemas Orientada para Objetos*, Editora McGraw-Hill do Brasil Ltda., 1990.

[19] Snyder, Alan,"The Essence of Objects: Concepts and Terms," *IEEE Software*. USA, Vol. 10. No. 1, pp. 31-42, Jan. 1993.

[20] Stefik, Mark, Bobrow, Daniel G.,"Object-Oriented Programming: Themes and Variations," *The AI Magazine*. USA, Vol. 6, No. 4, pp. 40-62, Winter 1986.

[21] Stroustrup, Bjarne,"What is Object-Oriented Programming?," *IEEE Software*. USA, Vol. 5, No. 2, p. 10-20, May 1988.