

A Confederation of Patterns for Resource Management

Rosana T. Vaccare Braga

ICMC-Universidade de São Paulo/ Universidade de Franca
C. P. 668, 13560-970 - São Carlos – SP - Brazil
Phone Number: 55 16 273-9679 / 273-9696
e-mail: rtvb@icmc.sc.usp.br

Fernão S. R. Germano

ICMC-Universidade de São Paulo
E-mail: fernao@icmc.sc.usp.br

Paulo Cesar Masiero¹

ICMC-Universidade de São Paulo
E-mail: masiero@icmc.sc.usp.br

Abstract

A confederation of patterns that covers a great number of applications in business systems is proposed. Resource Management is the term used to define these applications. It includes patterns for Resource Location, Resource Trading and Resource Maintenance. They are applied to examples as medical attendance, video rental, real estate rental, library service, show box office, fertilizer retail store and car repair shop. The proposal is based on professional practice, and results from the combination of recurring patterns, already covered in other PLoPs. The practice used mostly procedural languages, but through object oriented reverse engineering of real systems it was possible to explicit the proposed object oriented patterns.

1 - Introduction

We propose a confederation of patterns for designing systems for resource management. This confederation results from more than ten years of practice of systems development for medium and small business. Most of this practice has been done using procedural languages, but after getting acquainted with the concept of software patterns it was possible to recognize patterns as the basis of the reuse that has been done in the construction of more than twenty systems. They cover electric and mechanical car repair shops, television and video cassette repair shops, fertilizer retail stores, motorcycle dealers, electric appliance repair shops, perfume retail stores, video-rental stores, odontological attendance, publicity agencies, etc.

In all these systems, some objects can be easily identified, as well as the methods they involve, their relationships and attributes. For example, customer or party is a common object, with its usual attributes like code, name, address, etc. Another object that appears repeatedly is resource, with transactions to relate it to the parties. If we try to use recurring patterns to each of these situations, we can observe that many patterns are similarly applied.

The confederation of patterns here proposed, called Resource Management, includes three patterns that are formed by recurring patterns originally proposed by Coad [Coad 92], Fowler [Fowler 97] and Boyd [Boyd 98]. Their combination for the applications mentioned earlier have common points that are covered in the proposed set of patterns. The similarities among these systems make it worth to think about how to establish a pattern that could come to the mind of designers when they are developing a system in the domain.

Resource Management is broken down to Resource Location Pattern, Resource Trading Pattern and Resource Maintenance Pattern. These, however, are not mutually exclusive. Resource Location focus mainly on the satisfaction of a certain temporary need of a good, like the use of a physician time for an attendance or the use of a videotape for seeing a motion picture. Resource Trading focus on the transference of property of a

¹ Financial support from CNPq and FAPESP

good. For instance, real estate purchase or sale would be an instance of Resource Trading while real estate rental would be an instance of Resource Location. Resource Maintenance focus on the maintenance of a certain good, using labor and products to perform it, as in an electric appliance repair shop.

The following sections describe in details the three patterns of the proposed pattern set. The notation used to express the patterns is UML (Unified Modeling Language) [Eriksson 98, Fowler 98].

2 – Resource Location Pattern

2.1 - Problem

Many business systems deals with the location of resources for a certain period of time after which they are returned to their owners. A payment is due for this location.

2.2 - Context

Books in a library, doctor time in medical attendance, video tapes in video rental, properties in real estate rental and places in shows are examples of resource objects available for location, in the civil law sense of the term.

Even sharing many points in common, not all these resource location systems are equal. Each has its own peculiarities. They do, however, have similarities that are to be considered if we want to maximize the reuse of analysis, design and code. Borrowing a book in a library is not the same as renting a video in a rental shop. For the book you don't pay a daily rate, but you do pay a fine for delayed return. In both cases you can speak of resource location.

2.3 - Forces

- Even being different, resources to be located frequently have similar attributes.
- The location of the resource is usually done for a certain customer in a certain period of time. The location attributes and the customer attributes are similar in different applications.
- A certain payment is made related to the resource location or to the delay in returning it.
- The resource is usually of a certain type.
- A booking of the resource is usually necessary before its actual location.
- Often more than a copy or exemplar of the resource is available for location.

2.4 - Solution

In Figure 1 we present the **Resource Location Pattern**. The idea behind the pattern is simple. A customer can locate a resource exemplar of a resource. He/she can do this directly, or can book it previously. A booking fee can be collected too. A resource booking can be cancelled, and in this case it will not have a corresponding resource location.

A location of the same resource can be made many times by the same customer, each of which results in some kind of payment, that can be split in installments. Each installment has a due date, so that if this date is in the past, the installment is overdue or paid and if it is in the future it represents a coming installment. The attribute "situation" takes care of acquitting an installment when the customer pays it. The location lasts for some time, which varies from minutes to years, depending on the application. While a resource exemplar is located to a customer, it cannot be located to any other customer, but it can be booked for future location. A resource can often be grouped in types of resource, to provide the user some reports by type. The resource exemplar has the attribute "status" to show if it is or is not currently available.

The Resource Location Pattern embeds other recurring patterns, as will be shown in the "related patterns" section.

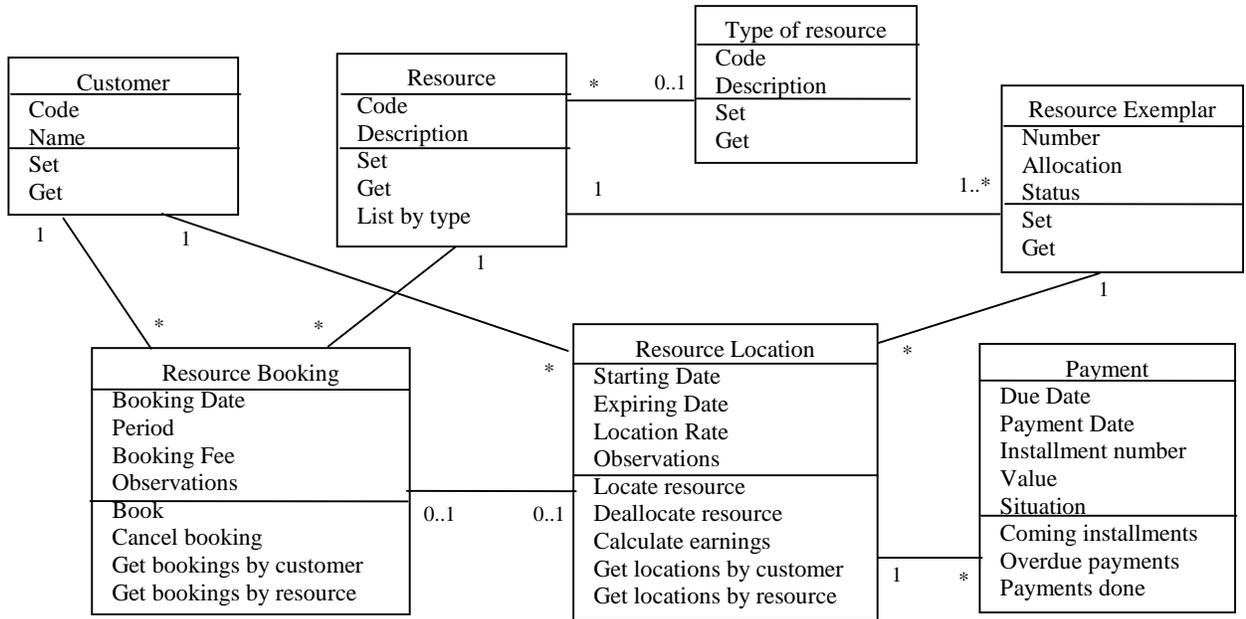


Figure 1 – Resource Location Pattern

2.5 - Examples

2.5.1 - Library Service

Figure 2 shows the instantiation of the Resource Location Pattern for a library service system. The reader can reserve a book before it is actually borrowed. The possible types of books are technical, scientific, fiction, literature, etc. The payment for the location is substituted by the fine for delayed return of the book borrowed. Some other peculiarities have to be noticed as, for example, that a certain book can have several copies. Booking is made for the book but what is actually located is the copy. Probably the fine is paid in cash, so there is no need to have attributes such as due date, installment number and situation.

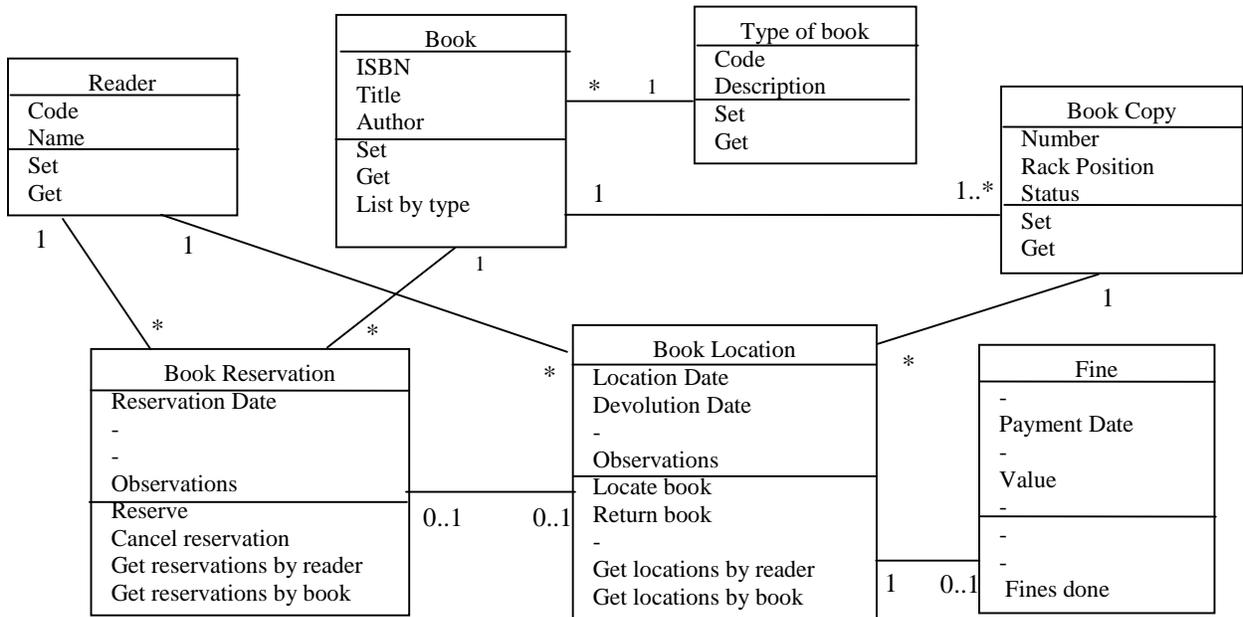


Figure 2 – Instantiation of Resource Location Pattern for a library service

2.5.2 - Medical Attendance

Figure 3 shows the instance of the Resource Location Pattern for a medical attendance system. If the medical attendance is done in a large hospital, for example, the booking is made for a certain specialty, without knowing which doctor will actually do the attendance. For small doctor's offices, however, the class "Resource Exemplar" may disappear, as the doctor leading each specialty is only one. In the attendance, instead of the starting date and the expiring date of the resource location there is only the attendance date and hour. So, the method to deallocate the resource is not necessary. Some specialties may be grouped in medical areas; for example, pediatrics is an area where there are surgeon's experts, neurological experts, newborn child experts, etc. The payment can be made in installments and the date in which they are due is object of control.

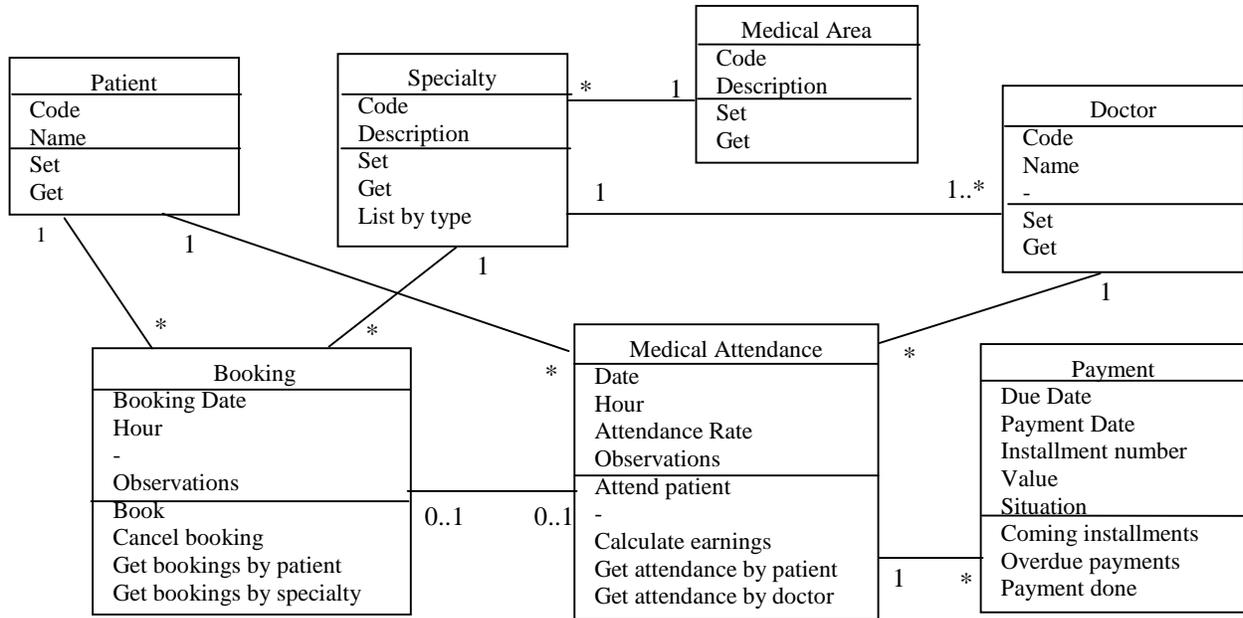


Figure 3 – Instantiation of Resource Location Pattern for a medical attendance system

2.5.3 - Video Rental

Figure 4 shows the instance of the Resource Location Pattern for a video rental shop. This is the most obvious application of this pattern. Here, also, there can be more than one tape for a certain video. Payment includes rental and, in certain cases, a fine for delayed return. This can be solved by adding an attribute "payment type" in the Payment class and assign to it the appropriate value. The possible types of videos are romance, drama, comedy, etc. You can book a video for the weekend, for example, or you can borrow a video tape directly, without a previous booking. The location is usually paid in only one installment, but the customer has the choice to pay it in advance or when he returns the videotape. So the attribute "installment number" can be discarded.

2.5.4 - Real Estate Rental

Figure 5 shows the instance of Resource Location Pattern for a real estate rental system. The resource booking here varies according to the nature of the property being rented. The booking of an apartment, for example, only makes sense while a candidate is being checked for acceptance, but the booking of a store in a shopping center usually forms a queue that can take years to advance. You can have similar real-estate properties to rent in the same building so that you can book either of them, but what is actually located is one real estate property. The possible types of properties are houses, apartments, stores, grounds, etc. This application is the one that better explores the features of the object Payment. Here, it is needed a rigorous control of the installments due dates and planning of coming installments.

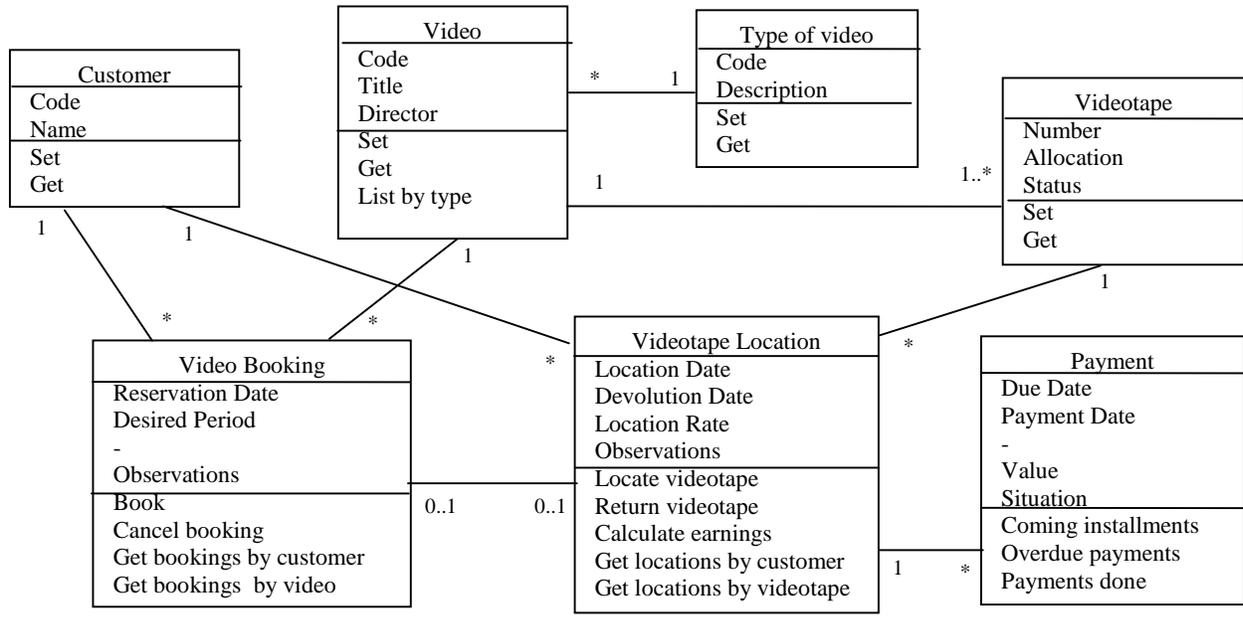


Figure 4 – Instantiation of Resource Location Pattern for a video rental

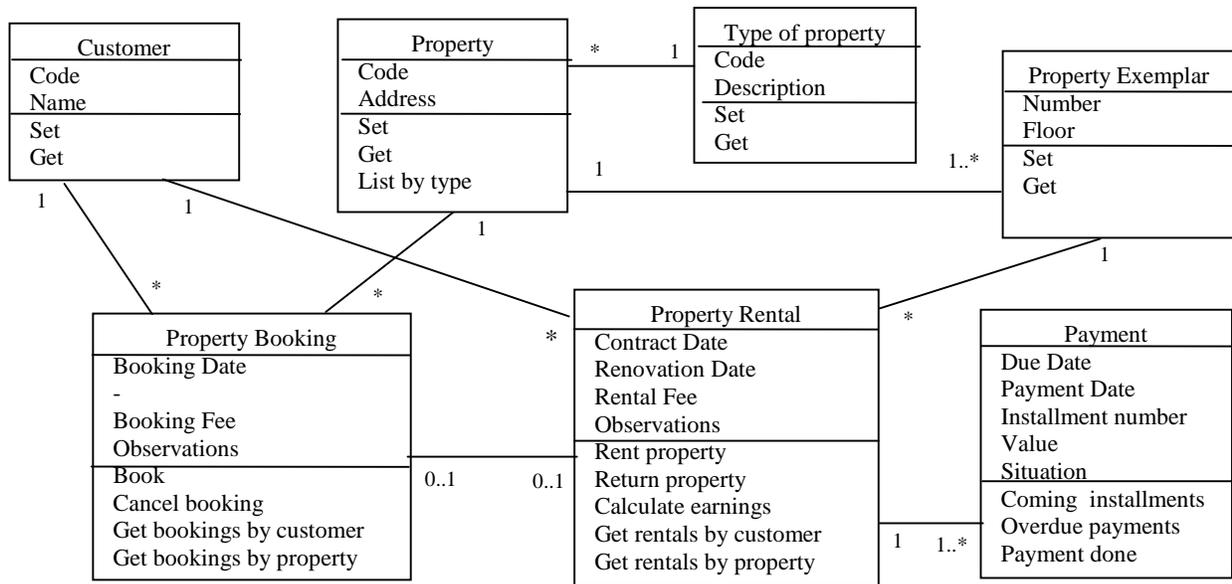


Figure 5 – Instantiation of Resource Location Pattern for a real estate rental

2.5.5 - Show Box Office

Figure 6 shows the instance of the Resource Location Pattern for a Show Box Office. Peculiarities here refer mainly to place booking and payment. Actually, the booking is only good if the customer goes to the ticket shop in a certain period of time and the payment can be reduced to an attribute of the ticket sale. The possible types of shows are play, opera, concert, motion picture, etc.

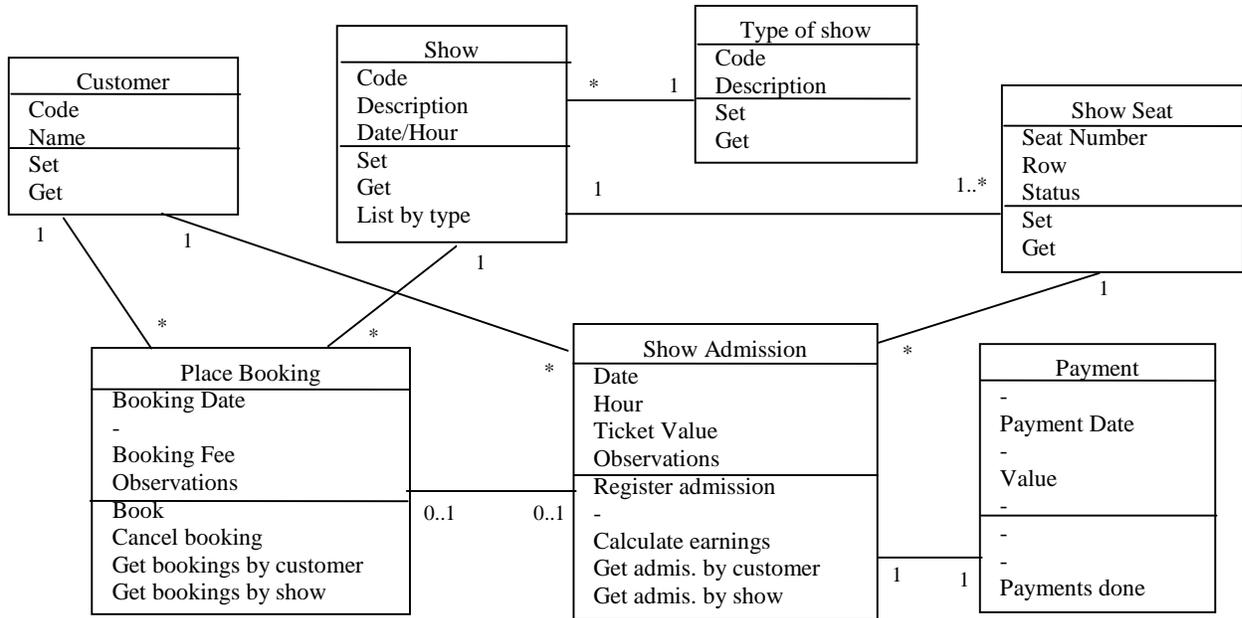


Figure 6 – Instantiation of Resource Location Pattern for Show Box Office

2.6 - Known Uses

Besides the cases exemplified, these patterns have many uses in many of the systems developed by our group. To be able to present these object oriented software patterns, we had to convert our thinking process of developing systems, from procedural oriented to object oriented. For this, we have done reverse engineering in a mechanical car repair shop system with more than twenty thousand lines of Clipper code [Penteadó 98], using Fusion/RE, an overall process to do object oriented reverse engineering [Penteadó 96]. This experiment, else than being educative, made possible deeper understanding of similar experiences [Gall 96].

In parallel, many of the instantiated patterns have been implemented in Delphi, some in the course of a modernization effort of the systems mentioned in the introduction. The systems functionality has been preserved but the user interface has been updated for Windows use as many of the applications have been originally developed under DOS.

2.7 - Related Patterns

Focusing only on the objects Customer, Resource and Resource Booking of Figure 2, we can see an application of the *Time-Association Pattern* [Coad 92]. It is also an application of the *Association Object Pattern* [Boyd 98], shown in Figure 7. Resource Booking is the association between the static objects Customer and Resource. The attributes of the static objects suffer variations but their methods are the same. The attributes and the behavior of the association object can be identified easily within the Resource Booking attributes and methods. Focusing now only on the objects Customer, Resource Exemplar and Resource Location we can see again an application of the Association Object Pattern.

Considering now only the objects Resource and Resource Exemplar we can see that they constitute an application of Coad's *Item-description Pattern* [Coad 92], shown in Figure 8. They can also be seen as a Johnson's *Type-Object* [Johnson 98] or Fowler's *Accountability Type* [Fowler 97]. Johnson, when discussing patterns related to his Type Object, mentions Gamma's Bridge, Strategy, State, Decorator, Flyweight and Prototype patterns [Gamma 95]. Depending on the particular circumstances, one of these design patterns can be used.

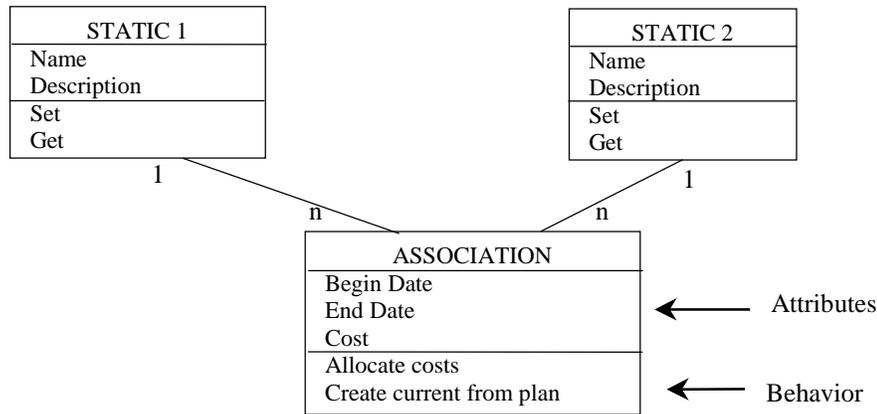


Figure 7 – Association Object Pattern [Boyd 98]

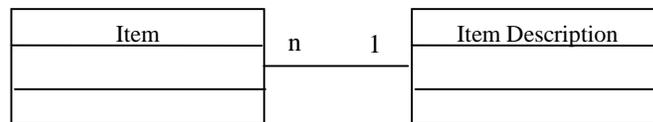


Figure 8 – Item-Description Pattern [Coad 92]

3 – Resource Trading Pattern

3.1 - Problem

A common situation in many business systems is the transference of property of a resource. This involves sales and purchases where a resource trading is done and someone has to pay for it.

3.2 - Context

A customer who buys a product in a supermarket or shop, a customer that requests a special book, a supplier that sells products after receiving a request and a customer that sells his/her house through a real estate dealer are examples of resources traded by two parties. Of course there are some differences among them, but the use of a pattern would simplify system analysis, maintenance and reuse.

3.3 - Forces

- The resource being traded has some common attributes and behavior, whatever is the kind of trade transaction.
- The trading of the resource is usually done for a certain party in a certain date. The trading attributes and the party attributes and methods are similar in different applications.
- A certain payment is made related to the resource trading.
- The resource is usually of a certain type.
- A resource request is sometimes necessary before its actual trade

3.4 - Solution

In Figure 9 we present the **Resource Trading Pattern**. A party can trade a certain quantity of a resource. He/she can do it directly or can make a previous request. A resource request can be cancelled, and in this case it will not have a corresponding trade transaction. Each trade transaction results in some kind of payment, as occurred with the Resource Location. In the same Resource Request several items can be requested, each of which related to a certain quantity of a different resource. Similarly, in the same Trade Transaction several items can be traded. The resource has the attribute “Stock Quantity” to show if it is or not

available in stock. As in the Resource Location Pattern, the resource can often be grouped in types of resource, to provide the user some reports by type. We use the term “party” here in the same sense as Fowler [Fowler 97], meaning both individual persons and organizations.

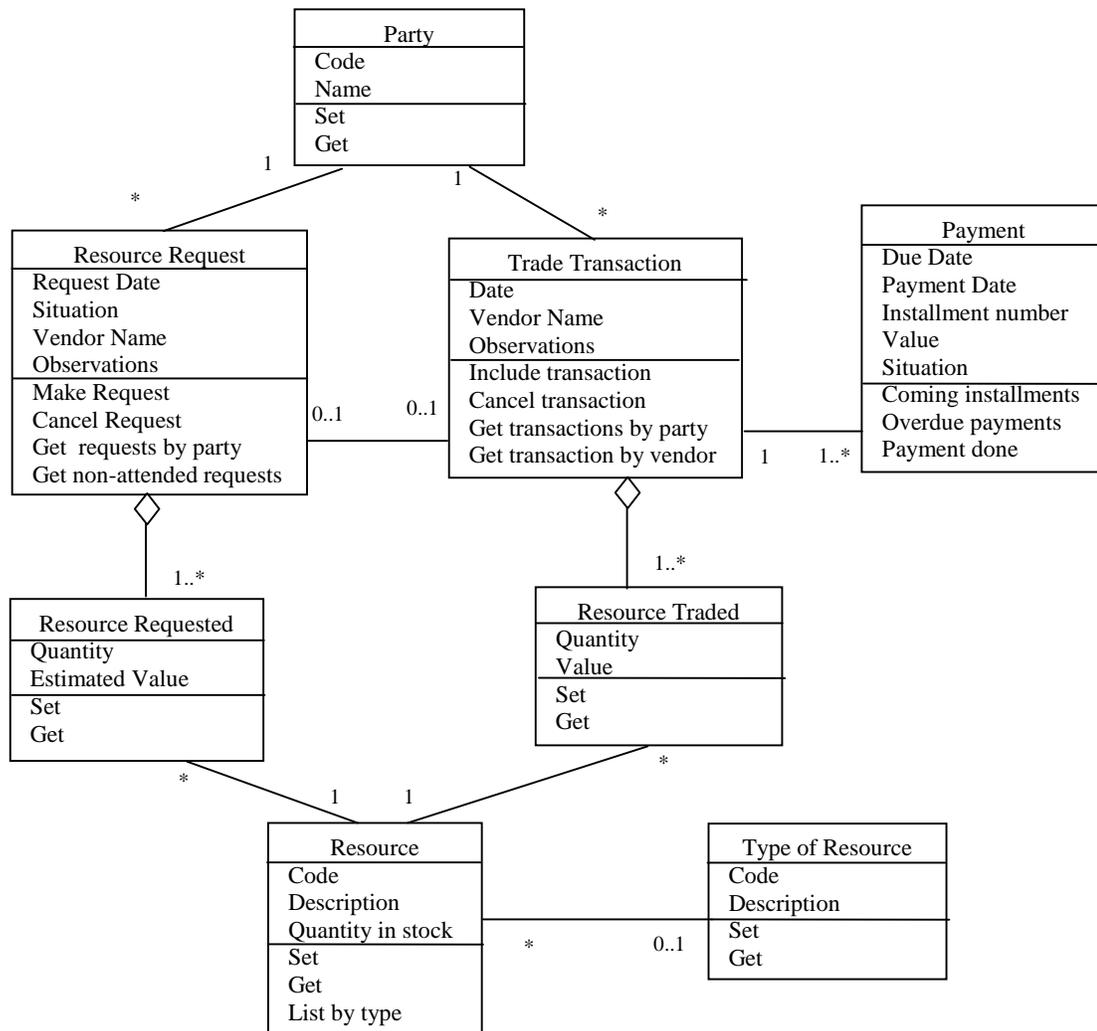


Figure 9 – Resource Trading Pattern

3.5 - Examples

Figure 10 shows the instantiation of the Resource Trading Pattern included in an inventory control system, for example for a fertilizer retail shop. In fact, it is a valid solution for a great variety of systems where the customer can buy a product in stock or make a previous request for it. The quantity sold is subtracted from the quantity in stock of the product. The possible types of product in a fertilizer retail shop are insecticides, ant-killers, acarus-killers, bacterium-killers, etc. In systems where there is no possibility of placing orders, as in retail supermarkets, for example, the classes Product Request and Product Requested are omitted and the customer buys the product directly.

Another subsystem of the inventory control system that uses the Resource Trading Pattern is the product purchase. In this case, the Customer class is substituted by the Supplier class and the Sale Transaction class by the Purchase Transaction class. The main difference, here, is that the quantity purchased is added to the quantity in stock of the product and the payments are accounts payable, instead of accounts receivable.

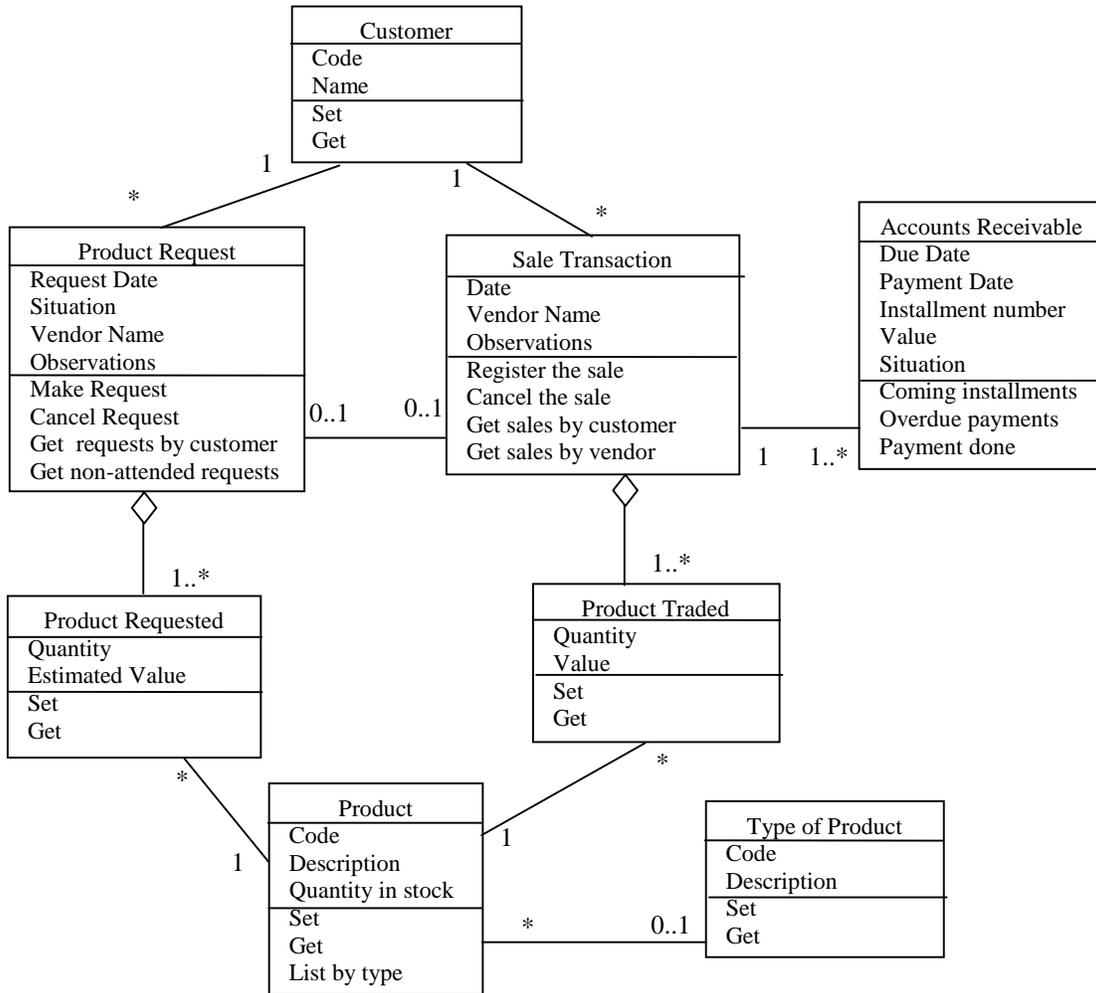


Figure 10 – Instantiation of Resource Trading Pattern for a Fertilizer Retail Shop

3.6 - Known Uses

This pattern is applied to many information systems developed by our group. It is present in three subsystems of the car repair shop system mentioned in section 2.6 and appears two times in each of the following systems mentioned in the introduction: fertilizer retail shop, perfume retail store and motorcycle dealer.

3.7 - Related Patterns

Focusing only on the objects Resource Request and Resource Requested of Figure 9, we can see that they can be considered as an application of the *State across a collection* pattern [Coad 92]. The same occurs with the objects Trade Transaction and Resource Traded. Other patterns occur that were already cited in section 2.7.

Boyd's instantiation of her *3-Level Order Pattern* [Boyd 98] for a purchase order has some things in common with the Resource Trading Pattern. The main differences are that she does not distinguish the Resource Request and the Trade Transaction, and that she does not include the object Payment.

4 – Resource Maintenance Pattern

4.1 - Problem

The repair or maintenance of resources is the goal of many business systems. The resource can have a problem or defect that needs repairing.

4.2 - Context

Cars, televisions, electric appliances and computers are examples of resources that have to be repaired when their owners perceive a fault. This repair may include both labor service and changed parts. The similarities among them make worth the definition of a pattern that could be applied when developing such systems.

4.3 - Forces

- In all these applications, the resource being repaired has some common attributes and behavior.
- The maintenance of the resource is usually done for a certain customer in a certain quantity of hours. The maintenance attributes and the customer attributes and methods are similar in different applications.
- The maintenance can be done through several tasks, each of which made by a different executor.
- The maintenance has a price to be paid by the customer.
- The maintenance usually needs to change some parts of the resource.
- The resource is usually of a certain type.
- The parts have common attributes and methods.

4.4 - Solution

In Figure 11 we present the **Resource Maintenance Pattern**. A customer can have many resources, which can be repaired when they have problems. The resources may be grouped according to their type. The resource maintenance involves tasks to be done in the resource, mainly labor tasks, and parts to be substituted in order to make the resource good again. The customer has to pay for the resource maintenance. To execute the labor tasks we have the maintenance executor that may be an individual person or a team. They have a commission for their work that has to be treated by the system.

4.5 - Examples

Figure 12 shows the instantiation of the Resource Maintenance Pattern to an electric and mechanical car repair shop. A customer has one or more cars that need to be repaired when they have a fault. The car repair shop has employees to do the labor service and has also many parts in stock, which are necessary to a car repair.

4.6 - Known Uses

The Resource Maintenance Pattern occurs in three systems developed by our group: the car repair shop, the television and videocassette repair shop and the electric appliance repair shop.

Some considerations have to be made about this pattern. Resource Maintenance can be considered as a combination of the Resource Location Pattern and the Resource Trading Pattern. We can think of maintenance as a location of a certain type of resource, labor in this case, and a trade of another resource, in this case the products used for executing the maintenance. This enforces the idea that the confederation of patterns may offer more than one solution to a problem, leaving to the analyst the choice of which of them to use.

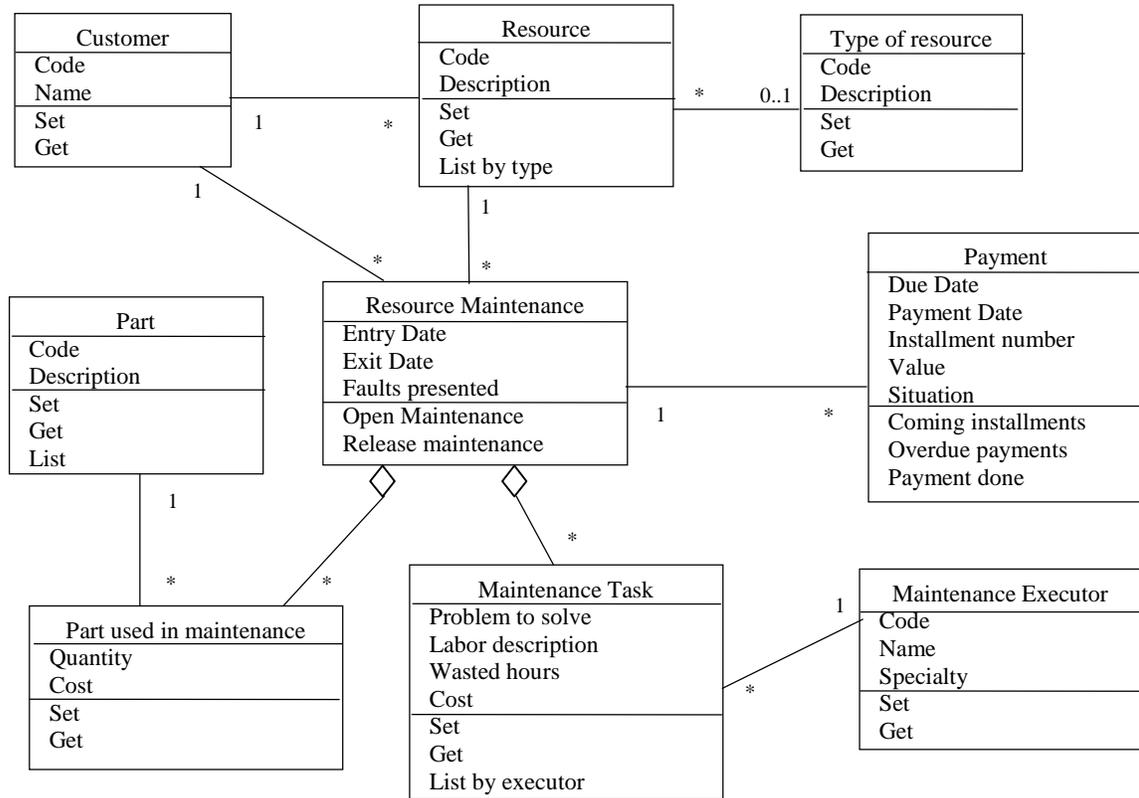


Figure 11 – Resource Maintenance Pattern

4.7 - Related Patterns

Focusing only on the objects Customer, Resource and Resource Maintenance of Figure 11, we can see that they can be considered as an application of the *Association Object Pattern* [Boyd 98], shown in Figure 7. Other patterns occur that were already cited in section 2.7.

Boyd’s instantiation of her *3-Level Order Pattern* [Boyd 98] for a work service order has some things in common with the Resource Maintenance Pattern. The main differences are that she distinguishes the Work Order Task and the Work Assignment. The first defines “the work to be done in different rooms at a point in time” and the second defines “planned or completed work by an employee on a specific Work Order Task”.

5 – Conclusions and Future Work

The proposed patterns may suffer some variations. Consider, for example, the Resource Location Pattern. We could apply it to an Airline Service, considering that a seat is a resource that can be located to a customer during the flight. The customer can also make a place booking. But there are some problems here. First, a flight usually has intermediate stops, so during the whole flight, several people may occupy the same seat. Second, some flights are seat free, so you do not know which seat is actually located to the customer. Third, some people would prefer to think about a place sale instead of a seat location.

Consider now the Resource Trading Pattern. It could be refined to deal with non-stock resources. The customer could request a product and buy it without the intervention of inventory control. The same could occur in the Resource Maintenance Pattern. The repair could use parts specially acquired for this purpose, or the labor service could be ordered to a subcontractor. These and other variations would lead to an improvement in our Confederation of Patterns so that it could apply to a wider domain and possibly become a pattern language.

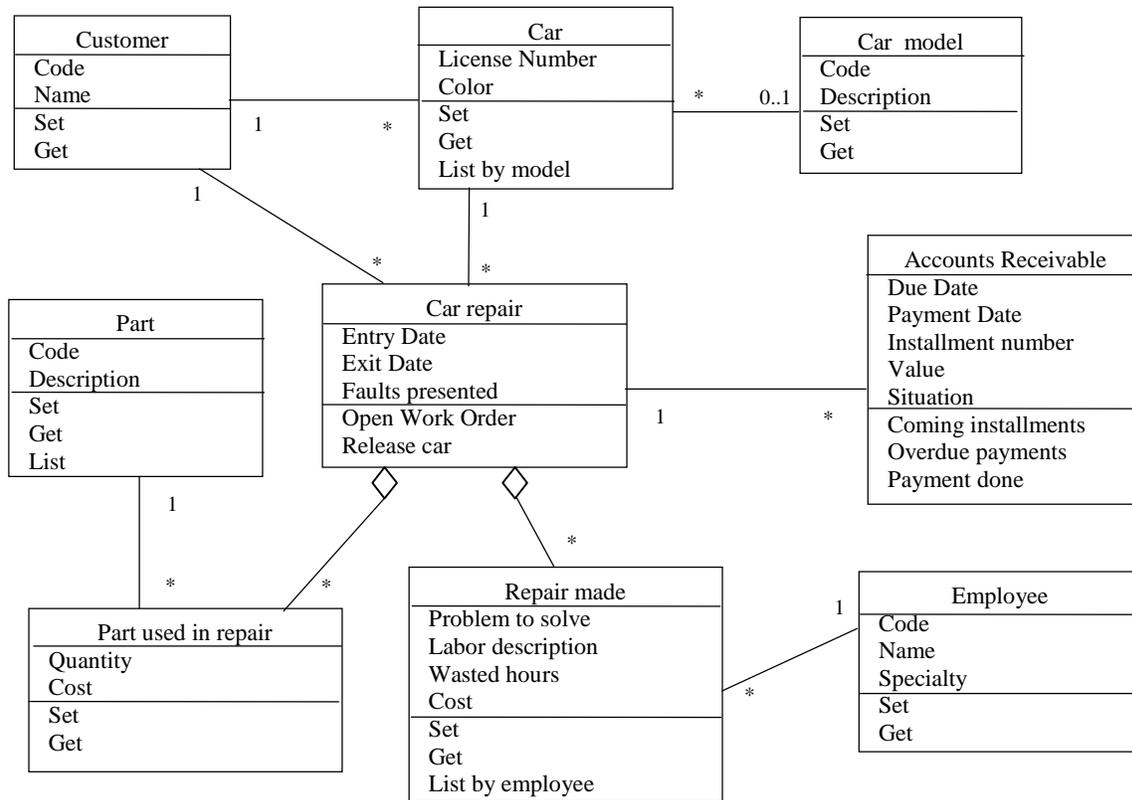


Figure 12 – Instantiation of Resource Maintenance Pattern for a Car Repair Shop

An example of integration with recurring patterns proposed by other authors to the same application domain would be the addition of the Accounting aspects of the transactions covered in the proposal. Fowler has a full chapter “Inventory and Accounting” dedicated to patterns for these aspects [Fowler 97]. Boyd exemplifies the use of the Association Pattern, in which our proposed confederation of patterns bears, in managing details of order attendance like shipment features, room assignment in maintenance work, etc. [Boyd 98]. These also would contribute to a more comprehensive set of patterns.

All these suggestions follow the evolution track suggested by Schmidt when discussing aspects of patterns that will receive considerable attention from the software community [Schmidt 96]. He specifically mentions the “Integration of design patterns to form pattern languages” as one of these aspects, and that “Developing comprehensive pattern languages is challenging and time consuming, but will provide the greatest payoff for pattern-based software development during the next few years”. This contribution represents work in progress that we expect to cast as a pattern language [Kerth 97] in the near future.

References

- [Boyd 98] Boyd, Lorraine – Business Patterns of Association Objects. In “Martin, Robert C. (ed.); Riehle, Dirk (ed.) and Buschmann, Frank (ed.) – Pattern Languages of Program Design 3”, Addison-Wesley, pp. 395-408, 1998.
- [Coad 92] Coad, Peter - Object-Oriented Patterns. Communications of the ACM, V. 35, n°9, pp. 152-159, September 1992.
- [Eriksson 98] Eriksson, Hans-Erik and Penker, Magnus – UML Toolkit, Wiley Computer Publishing, 1998.
- [Fowler 97] Fowler, Martin – Analysis Patterns. Addison-Wesley, 1997.
- [Fowler 98] Fowler, Martin; Scott, Kendall – UML Distilled – Applying the standard object modeling language. Addison-Wesley, 1998.

- [Gall 96] Gall, Harald C., Klösch, René R.; Mittermeir, Roland T. - Application Patterns in Re-Engineering: Identifying and Using Reusable Concepts. In Proceedings of 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, V.III, pp.1099-1106, July 1996.
- [Gamma 95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. - Design Patterns - Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series, 1995.
- [Johnson 98] Johnson, Ralph and Woolf, Bobby – Type Object. In “Martin, Robert C. (ed.); Riehle, Dirk (ed.) and Buschmann, Frank (ed.) – Pattern Languages of Program Design 3”, Addison-Wesley, pp. 47-65, 1998.
- [Kerth 97] Kerth, Norman L. and Cunningham, Ward – Using Patterns to Improve Our Architectural Vision, IEEE Software, pp. 53-59, January, 1997.
- [Pentado 96] Pentado, R.D.; Germano, F. and Masiero, P.C. - An Overall Process Based on Fusion to Reverse Engineer Legacy Code. In Proceedings of the III Working Conference on Reverse Engineering - IEEE. Monterey - California, pp. 179-188, 1996.
- [Pentado 98] Pentado, R.D.; Masiero, P.C.; Prado, A .F. and Braga, R.T.V. – Reengineering of Legacy Systems Based on Transformation Using the Object Oriented Paradigm, In Proceedings of V Working Conference on Reverse Engineering - IEEE. Honolulu - Hawaii, October, 1998, to appear.
- [Schmidt 96] Schmidt, Douglas C.; Fayad, Mohamed and Johnson, Ralph E. (guest editors) – Software Patterns. Communications of the ACM, V. 39, n°10, pp. 36-39, October 1996.