METHODS AND MODELS FOR ANALYSIS OF THE ORGANIZATIONAL ECONOMICS ACTIVITY USED FOR DEVELOPMENT OF INFORMATICS SYSTEMS

TEODORA VĂTUIU, MIOARA UDRICĂ*

ABSTRACT: *Study of organizational activity and highlighting problem situations that require specific solutions, require a detailed analysis of the models defined for the real system of the economic companies, regarded not as a sum of assets, but as organizations in which there are activities related into processes. In addition to the usual approach of using modeling languages in the development of information systems, in this paper we intend to present some examples that demonstrate the usefulness of a standard modeling language (UML) to analyze organizational activities and to report problem situations that may occur in data management registered on primary documents or in processes that bring together activities. Examples that have been focused on a travel agency can be extrapolated to any other organization, and the diagrams can be used in different contexts, depending on the complexity of the activities identified..*

KEY WORDS: *Unified Modelling Language; methods and models; analysis of economic activity, diagrams*.

JEL CLASSIFICATION: *D83, O32.*

1. GENERAL CONSIDERATIONS

Progress in the field of computer technology along with the major changes in the acquisition, processing and transmission of economic information requires a systemic vision of the complex activities carried out in the economic companies. Economic systems are considered as particular cases of open systems that achieve a dynamic equilibrium with the environment, while their main features remain constant. Accordingly, the companies are no longer viewed as an amount of assets, but become organizations with activities related into processes.

* *Assoc. Prof., Ph.D.,"Titu Maiorescu" University of Bucharest, Romania, vatuiu_teodora@yahoo.com*
Ph.D., "Titu Maiorescu" University of Bucharest, Romania, udrica@gmail.com

The analysis of the activity performed in an organization can highlight cases recorded in the management of data from the primary documents or in process that groups activities. Accordingly, to solve them, we can use methods that treat differently data and processing them, prioritizing data, or methods managing entity encompassing data and data processing them, following their behaviour along the scenarios defined according to user requirements.

The choice of method is based on the structure of managed data and on the nature of the information provided:

- informatics systems within operational departments process large volumes of data, with similar characteristics, determined by the structure of primary documents. In these cases we use the systemic methods, and define separate models for data and data processing, at the conceptual and logical level. These models come together at the physical level and lead to implementable solutions with relational database managed by database management systems.
- for decisional support systems and for management information systems we choose a method that can access data structures and operations on them encompassed in the same entity. The resulting models are based on the reuse of existing components and emphasize their transformation over time, their changes to events triggered by the system or users.

Activity analysis from an organization and highlighting problem situations that require solutions involves a detailed study of existing models and defining new models, uses modelling languages according to specified processes.

The methodological approach complying development method adapts existing information system and the issues raised by users through the development and integration of new components in order to improve his performance and functionality (Vatuiu &Udrica, 2010).

In addition to the usual approach of using modelling languages in the development of information systems, described in detail by professor Ion Lungu, "*Information systems development methods*", (Lungu, 2006) we propose in this paper to present some examples to demonstrate the usefulness of UML in analysing the activities from an organization, in the reporting of problems arises in an organization. Examples were taken from a travel agency and can be developed for any organization, and the diagrams can be used in a different context, determined by the complexity of activities.

2. UML DIAGRAMS DESIGNED FOR ANALYZING THE ACTIVITY OF AN ORGANIZATION

Using UML in analysis of activity carried out in an organization is determined by the complexity of the existing information system and the level of detail desired by the users.

In this article, the authors have chosen a travel agency as organization and design diagrams with a diagram generator, StarUML. More information on this topic

are provided in detail in the book "*Informatics systems. Problems and solutions*" (Udrică, Vatuiu et al, 2014).

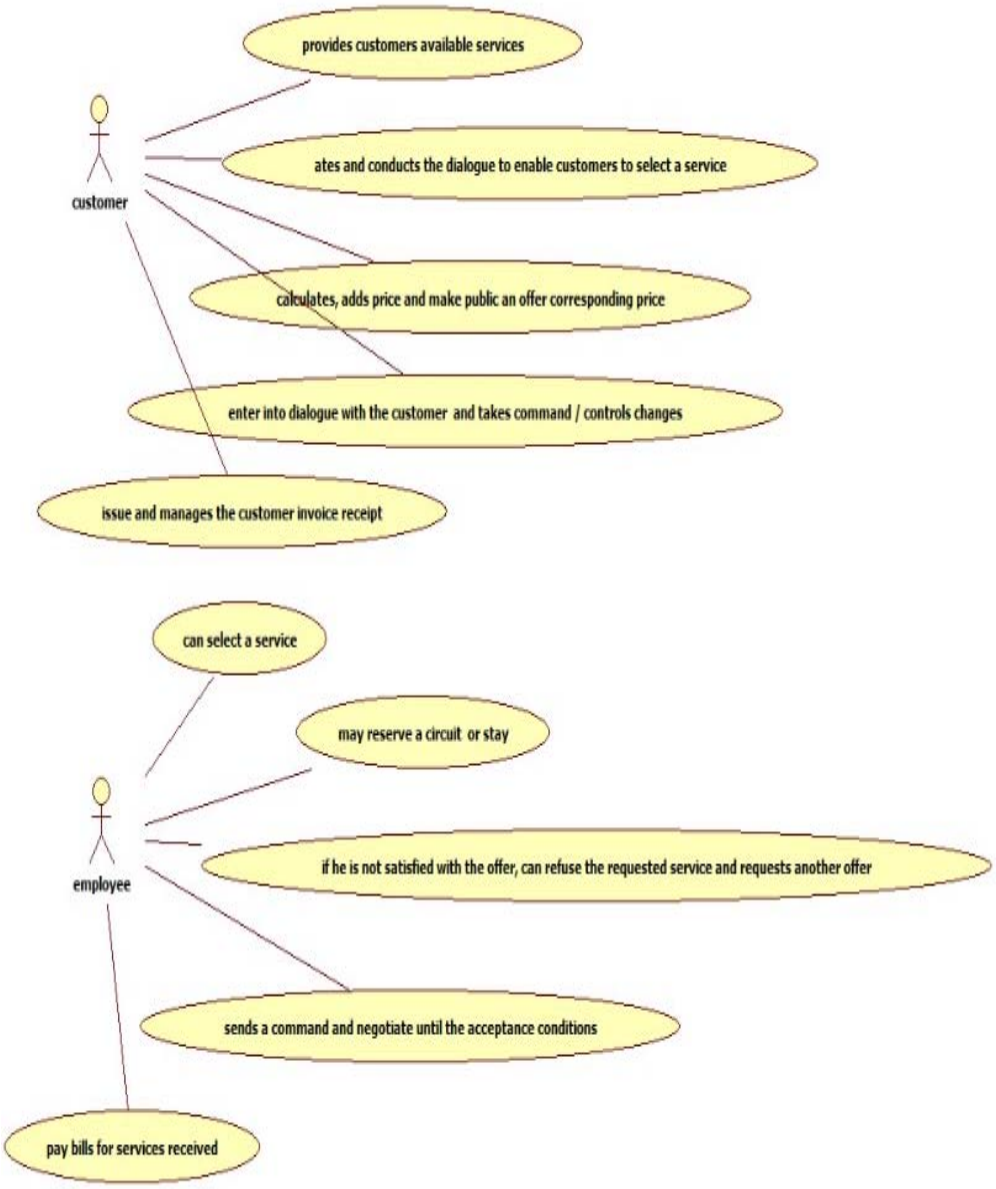

Figure 1. Use case diagram

As a starting point, for the existing informatics system we consider *known travel agency business description, use case diagram and class diagram:*

Business description:

- travel agency offers to customers available services (we take into account circuits or stays);
- depending on the selected offer and the conditions accepted by both parties, the client sends an order, taken and monitored by an employee of the company;
- at successful completion of the service, the customer receives an invoice with details of the services received;
- cashing the invoice is managed by an employee of the company;
- in some cases it is necessary to conclude a contract with the client; an employee of the company manages the agreement.

Use case diagram is shown in the figure 1.

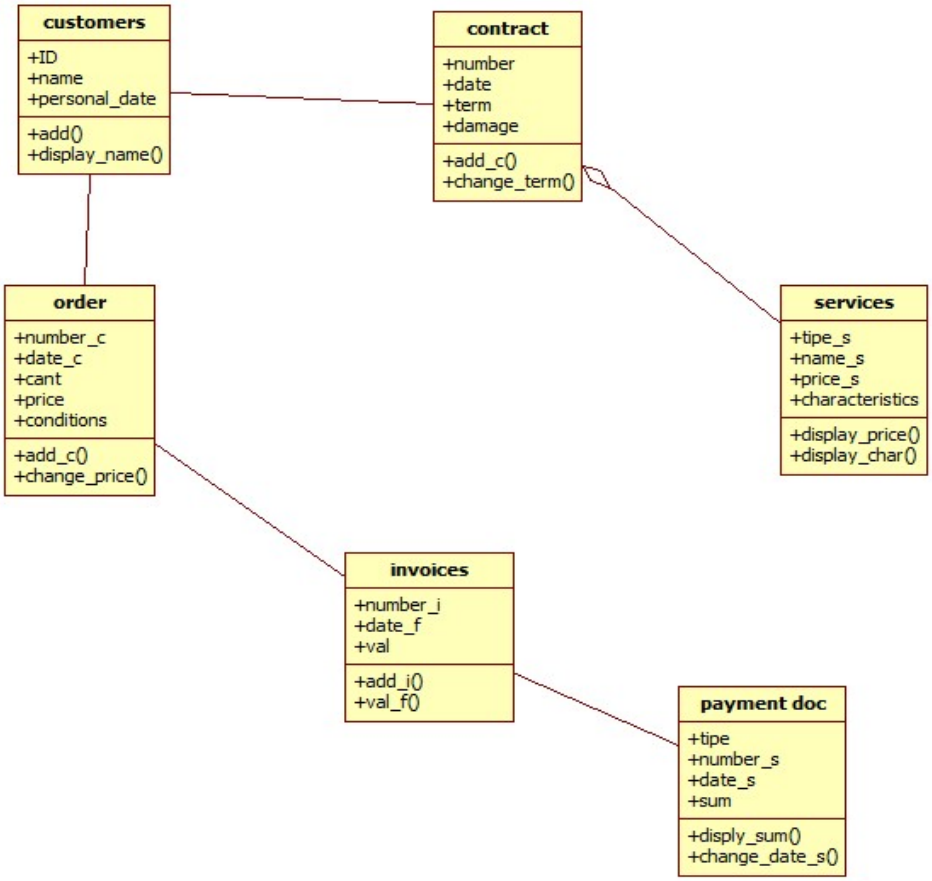

Figure 2. The class diagram

The actors involved in the processes described:

- **Employee**
  - provides customers available services;
  - initiates and conducts the dialogue to enable customers to select a service;
  - calculates, adds price and make public an offer corresponding price;
  - enter into dialogue with the customer and takes order / controls changes;
  - issue and manages the customer invoice receipt.
- **Customer**
  - can select a service;
  - may reserve a circuit or stay;
  - if he is not satisfied with the offer, can refuse the requested service and requests another offer;
  - sends an order and negotiate until the acceptance conditions;
  - pay bills for services received.

The class diagram is showed in figure 2.

3. ISSUES HIGHLIGHTED IN THE DIRECT RELATIONSHIP BETWEEN EMPLOYEE AND CUSTOMER

Problems in the direct relationship between employee and customer can be showed in sequence diagrams designed for different scenarios:

Example: the scenario which describe signing and performance of a contract:

- contractual conditions are established;
- the contract is signed by both parties;
- the contract signed by the client is recorded;
- the invoice is recorded when the contracted service starts;
- depending on the performance of the contract, the employee verifies the conditions and:
  - registers customer acceptance for services rendered;
  - tracks payment of the invoice by the customer;
  - closes and archives the contract.

or

  - draws up minutes for inappropriate services;
  - provides information for reversal invoices;
  - closes and archives the contract with the new conditions.

Corresponding sequence diagram is shown in the figure 3.

Notes: 1. As it shown in the previous scenario, in the performance of the contract there are situations in which the contracted services are not accepted by the client. This is not sufficiently surprised in sequence diagram (fig. 3.), so it is necessary to design a diagram of activities which includes decision blocks (fig. 4).

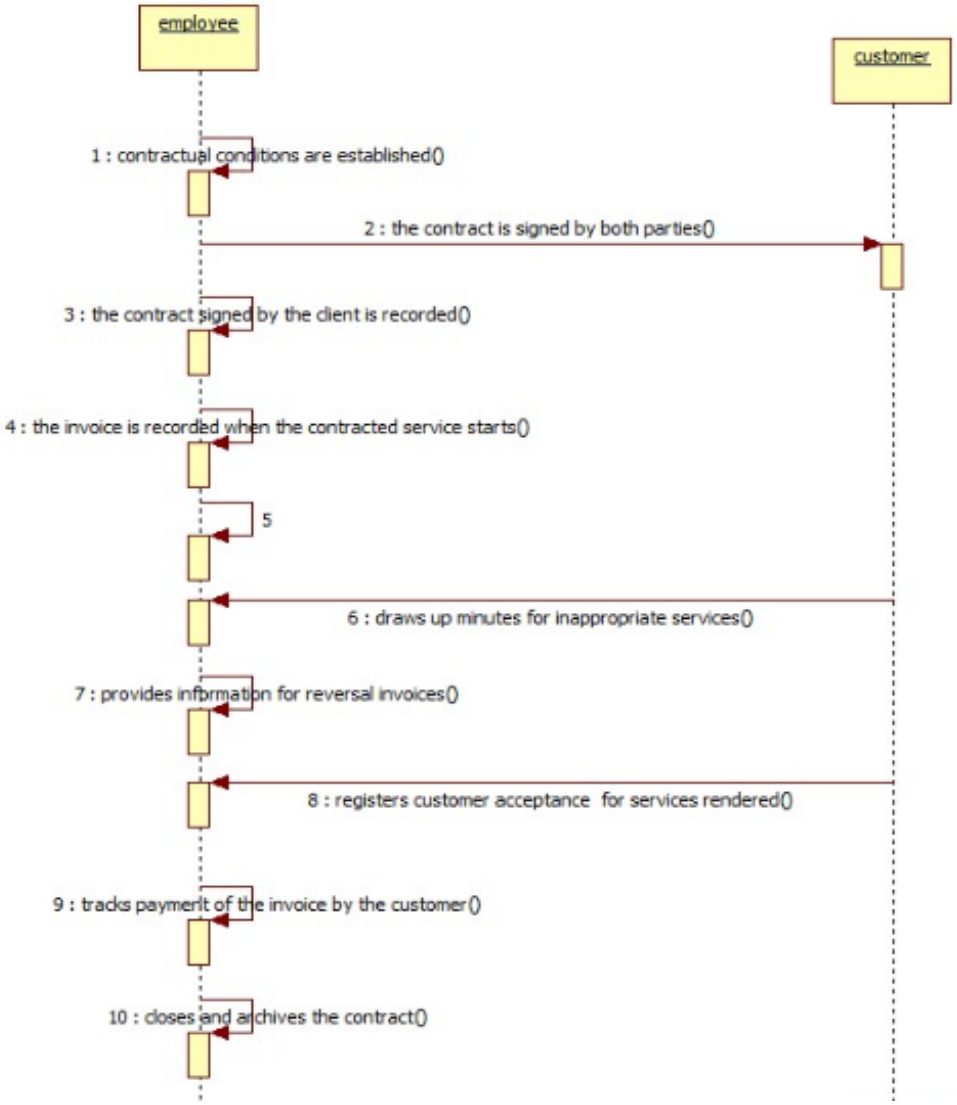

Figure 3. Sequence diagram

2. These diagrams indicate the necessity of adding new data and operations in the class diagram :

- order class: display_conditions ()
- services class: change_caracteristics()
  display_time_service()
  display_red_serv()
- customer class: analysis _cond_contract ()

accept_contract ()
display_accept_ct ()
display _accept PV ()
accept_ct (boolean attribute)
accept_PV (boolean attribute)
ID_ct (class type attribute)

- contract class : change_damage ()

3. By implementation of these attributes and operations, the tourist agency is able to verify how the contractual conditions were made, modified and complied. In this way on can interfere to solve problems in billing and contracting processes.

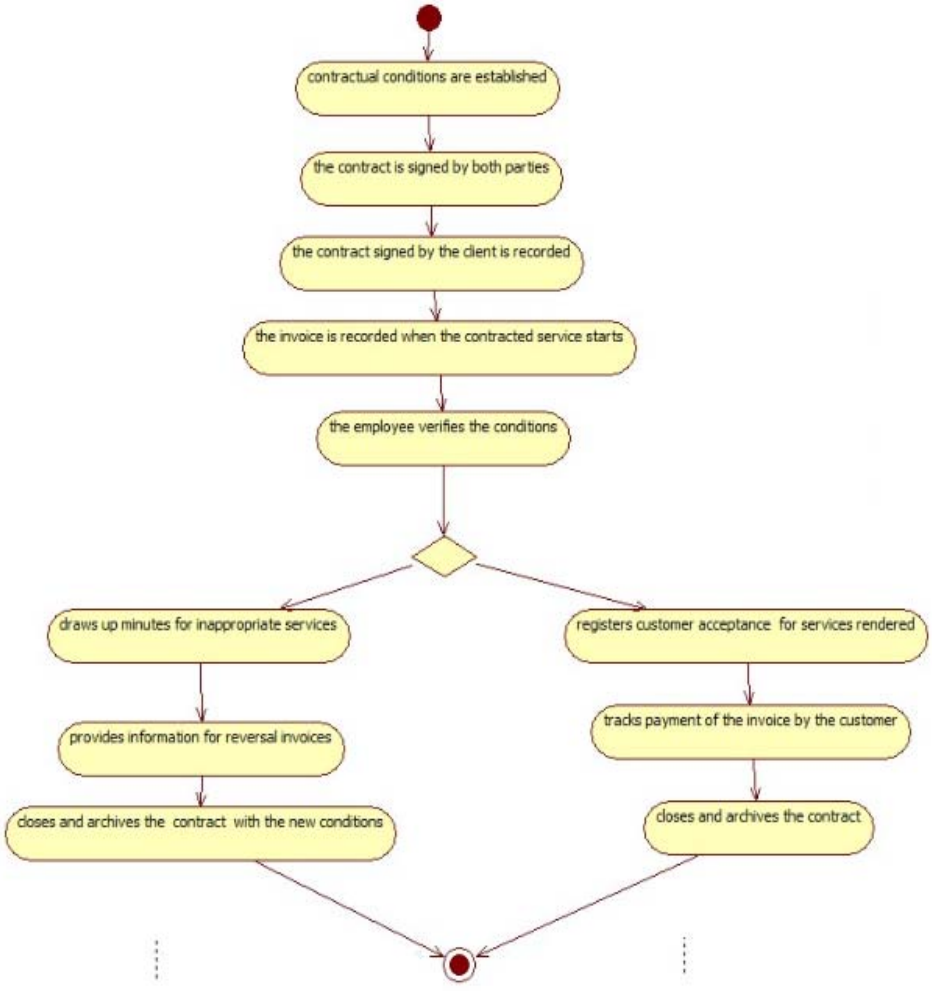

Figure 4. Diagram of activities

To capture the moments when the manager have to take a decision and their influence in activities sequence, on can design activities diagrams corresponding the key processes in the travel agency (Udrica&Vatuiu et al, 2014).

Example: the process of taking orders for tourist accommodation in the circuit can be detailed as follows:

1) confirming the customer;
2) take order from the client;
3) negotiating start date of service;
4) determining the length of service;
5) renegotiate until confirming the details of the order;
6) record order.

Activities diagram including objects is shown in the figure 5. Including objects is a common practice used in business modelling and focuses on status changes during information flow. In particular, it can track the flow of primary documents that contained data entered into the informatics system.

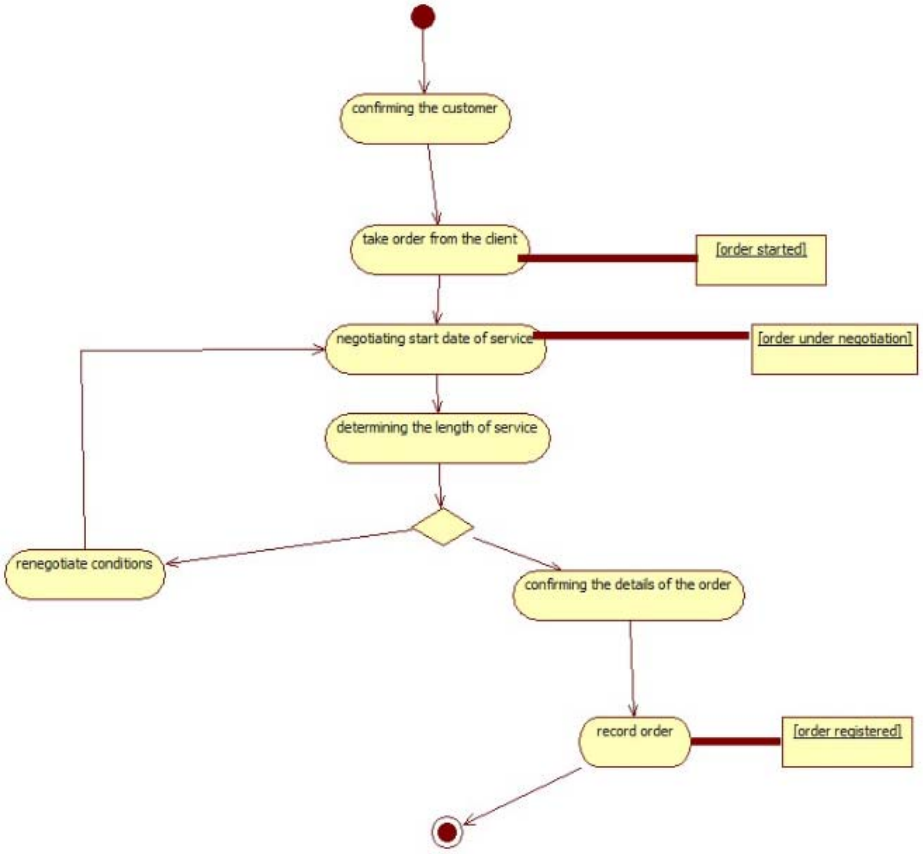

Figure 5. Activities diagram with objects

4. ISSUES RAISED IN THE INFORMATION FLOW OF PRIMARY DOCUMENTS

To highlight how the flow of information from tourist agency influences the state of primary documents that record the activities, we can design state diagrams for classes corresponding to these documents.

Example: The state diagram for the order class is shown in the figure 6.

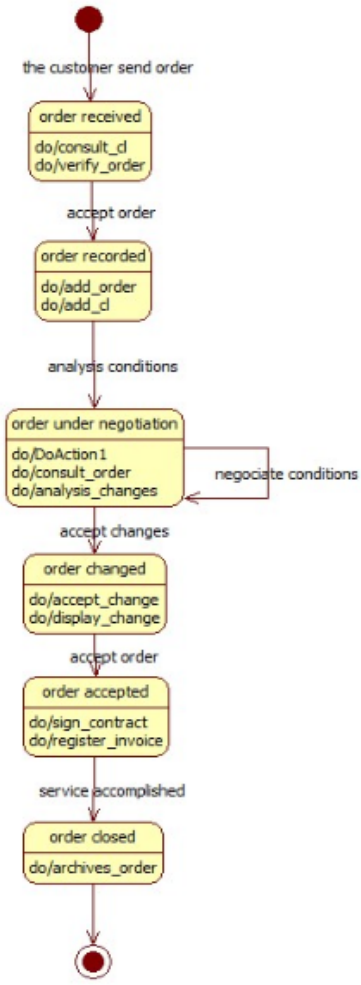

Figure 6. State diagram for the order class

Detailing the status changes depending on the events causing them requires adding attributes and operations in class diagram. Implementation helps to track how to supervise an order made by a customer. For this example on add:

- state_cd attribute - which can highlight any time place and status of orders
- operations:
  consult_cd ()
  change_cd ()
  archive_cd ()
  set_state_cd ()
  display_state_cd ()

Taking into account all these changes, the class diagram become: (fig. 7)

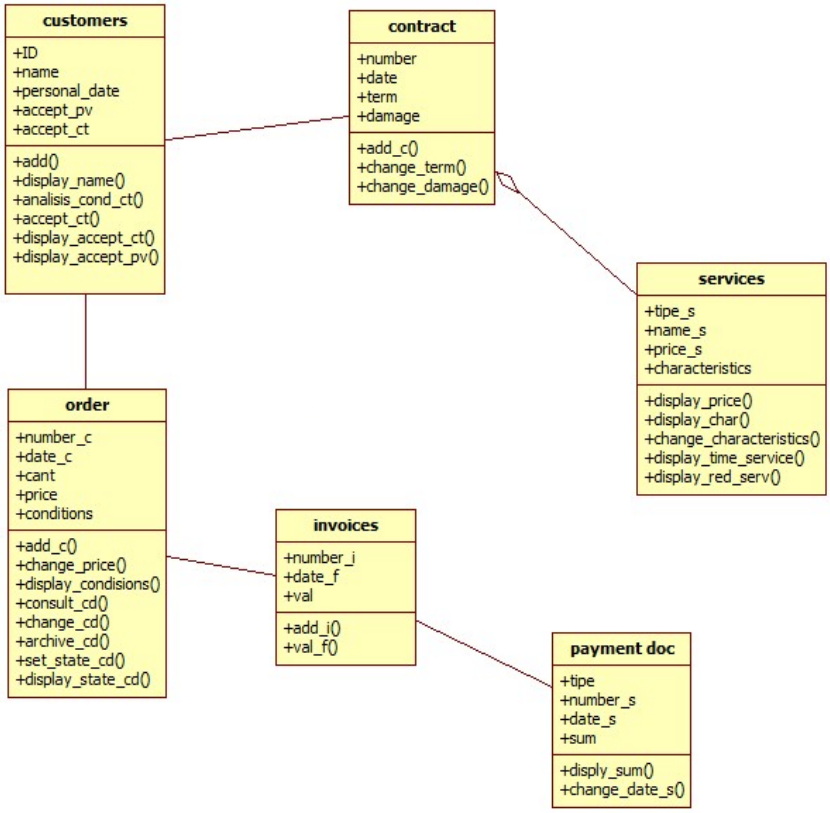

Figure 7. Class diagram

5. SOLUTIONS TO IMPLEMENT MODELS DEFINED

A. The class diagram from the figure 7 may lead to a relational database. Querying this database we can provide solutions to issues raised by users. Specified

operations can be implemented with a relational Database Management System (DBMS).

Notes:

1. For the definition of the relational database, from the full description of the activity in the travel agency we select only contracting and invoicing processes of accommodation.

2. We rewrite user requirements so as to reveal information necessary to establish relationships between tables:

- the travel agency offers accommodation services to customers;
- the travel agency concludes contracts with customers; a client may conclude several contracts; a contract may refer a single service;
- the invoices issued to customers are conform with the contract; several invoices correspond to a contract ; an invoice can record more services;
- the invoices are paid using payment document; an invoice can be paid with a single payment document; a payment document can pay more bills.

3. Class diagram corresponding to these processes, with the specified multiplicity, is shown in the figure 8.

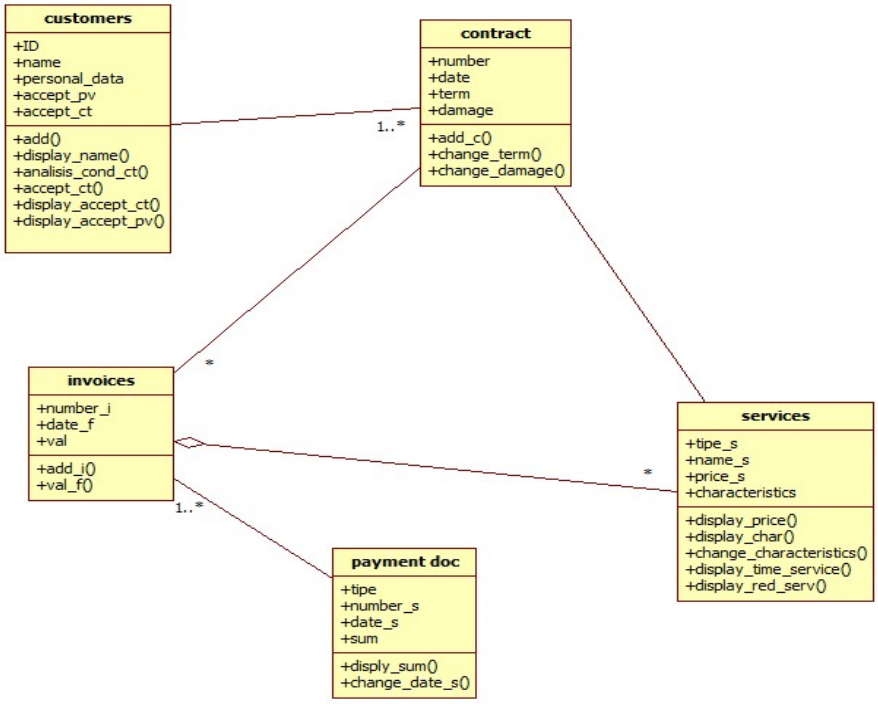

Figure 8. Class diagram

Depending on the multiplicities shown in the class diagram from figure 8, we can define a relational database. It comprises the following tables:

Customer=(id_cl, name_cl, personal_data)
Contract=(nr_ct, date_ct, termen, id_cl, id_services)
Invoices=(nr_invoice, date_invoice, nr_ct)
Servicies=(id_serv, den_serv, pret_unit)
Invoice_Services=(nr_invoice, id_services, durata)
PayDoc=(nr_doc, date_doc)
Invoice_Doc=(nr_invoice, nr_doc, sum)

Note: To query this database can add new attributes and operations in class diagram, as we show in the fig. 9.

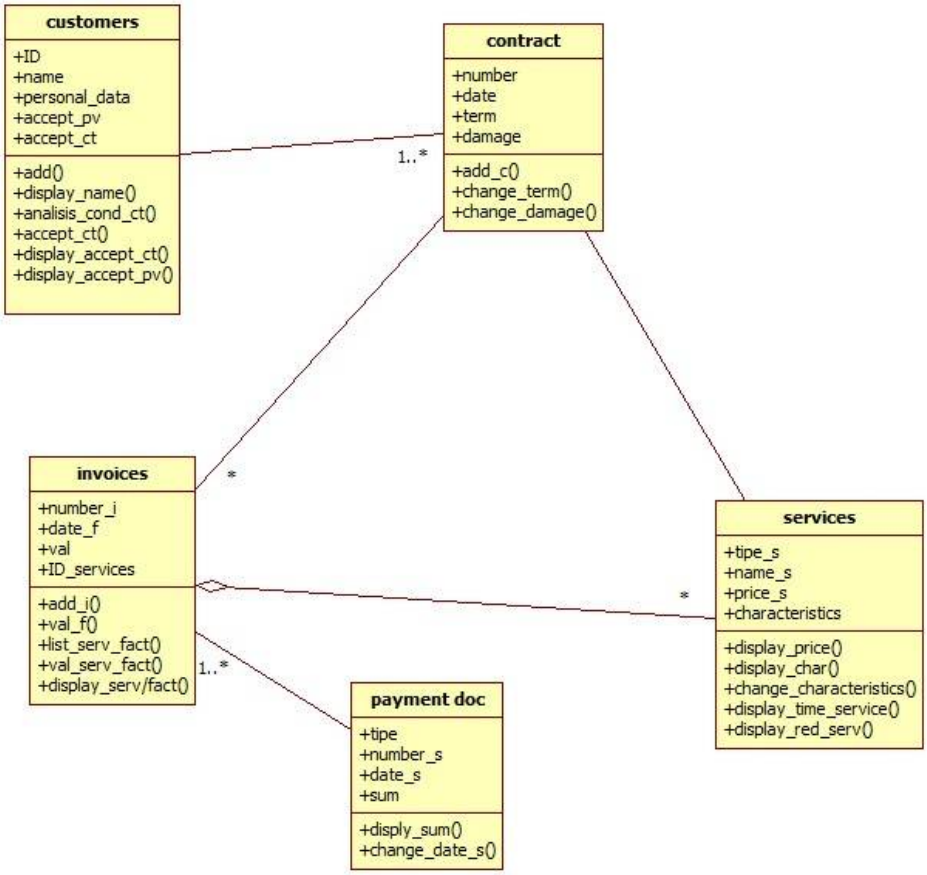

Figure 9. Class diagram

B. The diagram from figure 7 can be implemented in an object oriented language. We present the code generated by the option Generate Code (Java) offered by StarUML for classes invoices and services:

```
public class invoices {
        public Object number_i;
        public Object date_f;
        public Object val;
        public Object ID_services;
        public void add_i() {}
        public void val_f() {}
        public void list_serv_fact() {}
        public void val_serv_fact() {}
        public void display_serv/fact() {          }
}
public class services {
        public Object tipe_s;
        public Object name_s;
        public Object price_s;
        public Object characteristics;
        public void display_price() {}
        public void display_char() {}
        public void change_characteristics() {}
        public void display_time_service() {}
        public void display_red_serv() {}
}
```

6. CONCLUSIONS

Modelling language used in this paper helps define a model of the activity carried out in organization, highlights errors in the solutions revealed in already designed systems, contributes to provide information from a relational database and provides detailed specifications for model synchronization with source code.

UML diagrams designed have proposed to help understanding the existing information system, to highlight errors in the solutions already provided, to bring more information to resolve issues.

Defining a database from class diagram brings significant advantages in providing information to users. Complete list of information from the database, it highlights the beneficiaries of this information and the time to be provided, it highlights moments to be queried / updated database so as to avoid the occurrence of problem situations.

Implementing using an object oriented language takes advantage of defining reusable classes, ever integrated into a new system. Meantime, create a better distribution of tasks/ information on beneficiaries; show the information flow of primary documents, the correlation between different primary documents, those responsible for their management.

Solutions can be developed or may be an example to other issues raised in an organization.

REFERENCES:

[1]. Booch, G.; Jacobson, I.; Rumbaugh, J. (2010) *OMG Unified Modeling Language Specification*, Version 1.3 First Edition: March Retrieved 12 August

[2]. Gabay, J. (2002) *Merise vers OMT et UML*. InterEdition, Paris.

[3]. Ghencea, A.; Vătuiu, T.; Țarcă, N. (2011) *Models and solutions for the implementation of Distributed Systems*, the International Conference "European Integration – New Challenges" 7Th edition, may, Annals of the University of Oradea, Economic Science, University of Oradea Publishing House, pp. 801-807

[4]. Győrödi, R.; Lungu, I.; Győrödi, C. (2012) *Sisteme avansate de descoperire a cunostiintelor din baze de date*, Editura Universitătii din Oradea, Oradea

[5]. Ionita, D. (2010) *UML in Business Administration*; Journal of Knowledge Management, Economics and Information Technology; Issue 1; December

[6]. Lungu, I. (2005) *Metode de dezvoltare a sistemelor informatice*, Editura Universitas, Petrosani

[7]. Lungu, I. (2005) *Sisteme informatice executive*, Editura ASE, Bucuresti

[8]. Țarcă, N. (2005) *Informatică economică*, Editura Universității din Oradea, Oradea

[9]. Țarcă, N.; Vătuiu, T.; Țarcă, I. (2010) *A Study Regarding the Use of the Information Technology and Communications in the Logistic Activity of Small and Medium Enterprises*, Annals of the University of Petroşani, Economics, vol.10(4), pp. 347-354

[10]. Udrică, M.; Vătuiu, T.; Ghencea, A. (2011) *Issues Concerning the Use of UML Diagrams to Define the Underlying Process Model Simulation*, Annals of the University of Petroşani, Economics, vol.11(4), pp.305-314

[11]. Udrică, M., et al (2009) *UML prin aplicatii- Studii de caz privind dezvoltarea sistemelor informatice*. Editura Renaissance, Bucureşti

[12]. Udrică, M. (2000) *Modelare orientată obiect*, Editura Cison, Bucureşti

[13]. Udrică, M., Vătuiu, T. et al. (2014) *Informatics systems. Problems and solutions*. Editura Hamangiu, Bucuresti

[14]. Vasilciuc, B. (2011) *Baze de date Access*, Editura Renaissance, Bucureşti

[15]. Vătuiu, T.; Udrică, M. (2010) *Sisteme informatice. Eficiență prin analiză, proiectare, implementare*. Editura Renaissance, Bucureşti

[16]. Vătuiu, T. (2007) *Proiectarea sistemelor informatice*, vol. 2, curs universitar ID, Editura Academica Brâncuşi, Targu-Jiu

[17]. Vătuiu, T. (2006) *Proiectarea sistemelor informatice: aspecte conceptuale şi manageriale*, Editura Universității din Sibiu

[18]. Vătuiu, T. (2001) *Procesul de dezvoltare Software UML*, Sesiunea națională de comunicări ştiințifice, ediția a-IX-a, Universitatea „Constantin Brâncuşi" , Facultatea de Ştiințe Economice, Tg-Jiu, ,Vol.4, Editura „Studii Economice", pag. 160

[19]. Vătuiu, T. (2001) *Analiza orientată pe obiecte în cazul metodologiei UML*, Sesiunea națională de comunicări ştiințifice, ediția a-IX-a, Universitatea „Constantin Brâncuşi", Facultatea de Ştiințe Economice, Tg-Jiu, ,Vol.4, Editura „Studii Economice", pag. 164

[20]. Vătuiu, T.; Țarcă, N.; Udrică, M.; Popeangă, V.N. (2010) *The Use of the EMCS Business Web Application for Monitoring the Movement of Excise Goods within the European Community*, Annals of the University of Petroşani, Economics, vol.10(4), pp. 363-374