

Using AMDD method for Database Design in Mobile Cloud Computing Systems

Silviu Claudiu POPA, Mihai-Constantin AVORNICULUI, Vasile Paul BTESFELEAN
„Babeş–Bolyai” University Cluj-Napoca,
Faculty of Economic Science and Business Administration
silviu.popa@econ.ubbcluj.ro, mihai.avornicului@econ.ubbcluj.ro,
paul.bresfelean@econ.ubbcluj.ro

The development of the technologies of wireless telecommunications gave birth of new kinds of e-commerce, the so called Mobile e-Commerce or m-Commerce. Mobile Cloud Computing (MCC) represents a new IT research area that combines mobile computing and cloud computing techniques. Behind a cloud mobile commerce system there is a database containing all necessary information for transactions. By means of Agile Model Driven Development (AMDD) method, we are able to achieve many benefits that smooth the progress of designing the databases utilized for cloud m-Commerce systems.

Keywords: Agile method, Agile Model Driven Development, Mobile Cloud Computing, Database Refactoring, Database Designing, Mobile e-Commerce, UML

1 Introduction

The unprecedented development of the technologies used in IT in the past years, determined by the necessity of storage and fast transmission of information with the lowest costs, revolutionized the global, direct or retail commerce.

Data intensive web based systems, that in the most of their uptime do data processes have become a part of our everyday life. Several systems of this type appeared on the internet: webshops, online ticket shops, banks, etc. These are used everyday by users around the world.

As a result, up to date wireless telecommunications technologies lead to the birth of new types of e-commerce, such as the mobile e-Commerce (m-Commerce). This branch allows commerce to use a large variety of devices and develop anywhere and anytime. There are various mobile business applications, such as: online business administration, shopping, financial services, information services, electronic payments, advertisement, entertainment, etc. One of the areas of mobile transactions is represented by the mobile commerce, where is easily observed a migration from the PC oriented approach towards a multi-device advance (tablet, Smartphone, etc.)

In the Internet era, Cloud Computing has become a significant research topic for both scientific and industrial communities. It represents an abstract computing network, highly scalable, able to support user applications with usability proportional costs.

This network is based on a virtual infrastructure, invisible to the user. It could be located anywhere in the world, and could be generally used in a web browser without the additional install of other software [37].

A definition used by many experts in the field is given by the USA's National Institute of Standards and Technology (NIST) [36], one of the most authorized voices in the field:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics (On-demand self-service, Broad network access, Resource pooling, Rapid elasticity, Measured service), three service models (Software as a Service, Platform as a Service, Infrastructure as a Service), and four deployment models

(Private cloud, Community cloud, Public cloud, Hybrid cloud)".

Experts and researchers from the European Commission, illustrated in one of their reports the essence of a cloud system [16]:

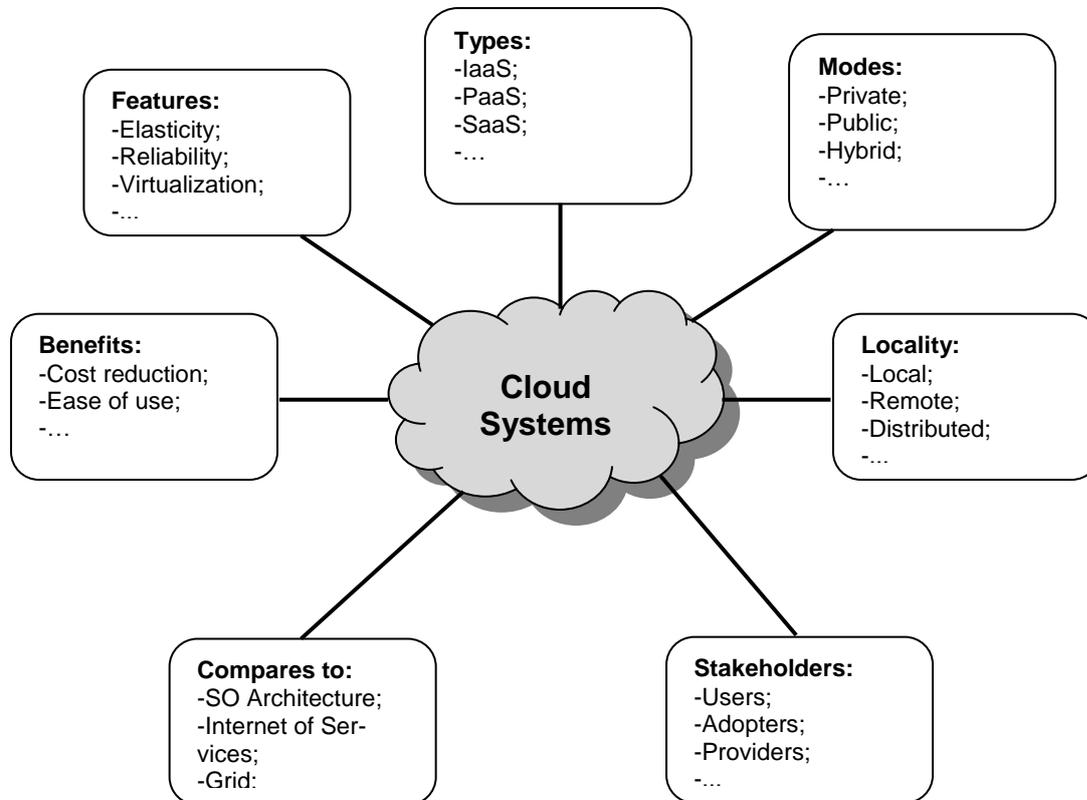


Fig. 1. A view on the main aspects forming a cloud system [35]

Last year, TechSoup Global and its network of partners performed a study of various NGOs, nonprofits, foundations and charities in the world (over 10 500 participants from 88 countries), with the purpose of identifying their present tech infrastructure and potential adopting the cloud technologies. Some of the study's main results included [35]:

- 90% of respondents worldwide were using at least one cloud computing application.
- 53% report planned to move a "significant portion" of their IT to the cloud within three years.
- 60% affirmed that the lack of knowledge was the supreme barrier to greater use of the cloud.
- 79% said the greatest advantage was easier software or hardware administration.
- 47% said cost-related changes and ease of setup would be the greatest motivators for moving their IT to the cloud.

- NGOs in Egypt, Mexico, India, and South Africa were the most hurried in moving their IT to the cloud.

The combination of mobile network and cloud computing creates a new IT domain, namely Mobile Cloud Computing (MCC). According to Technopedia [34], MCC represents "a technique or model in which mobile applications are built, powered and hosted using cloud computing technology". It is also named MoClo, a combination of MOBILE CLOUD. This advance determines the developers to design applications for mobile users without the restrictions imposed by the mobile operating system, the computing or memory capabilities of the device. Various applications based on mobile cloud computing have been developed and served to users [35], such as Google Gmail, Maps and Navigation systems for Mobile, Voice Search, and some applications on an Android platform,

MobileMe from Apple, Live Mesh from Microsoft, and MotoBlur from Motorola.

The mobile commerce systems in cloud need to use a database with appropriate information. In order to design a database it is required to exactly comprehend the information desired to be memorized and the way in which the information is stored. A proper information organization allows flexibility in the database and extra volume to combine information in different ways with reference to specific events.

The Agile methodology defines a system of methods meant to reduce the cost brought by changes and for a faster development of the system, especially in the following contexts [1] [2]:

- when the most important elements show up later in the project;
- when we are forced to adapt to elements that cannot be controlled, this methodology is being used for a faster development of the system.

It represents the quality of being agile, able to progress quickly and easily, the movement skills – of the software methods development, intending to offer an answer to the business community eager to have a lighter weight and a faster software process [2]. It is the case of software industry's rapid growth in both Internet as well as in mobile applications. The new Agile method arises disputes in literature, even though the attempts in developing systems using the new methodology will continue in the next years.

2 Opportunities of using UML

Using UML while building cloud computing system is opportune from the following views [1][7]:

- UML offers powerful instruments for modeling the aspects of behavior. Classes contain data and their associated processes;
- UML offers a whole view on the groupings of different components in the form of packets and the physical places of these packets;
- the UML-s diagram of collaboration is very important to understand the interac-

tion of different objects of the system and the messages they exchange;

- Cloud computing systems are complex systems that present a more dynamic evolution than other systems. The use of a flexible instrument is necessary for the analysis and design of these systems to permit future extension of the system;
- UML uses mechanisms that permit the easy reuse of systems components (capsules, heritage and polymorphism);
- UML permits an integrating view of the data and their processes;
- analysis and design with UML permits the continuous development of the model;
- development methods that use UML are iterative methods guided by the usage case diagrams. This strategy permits the elimination of several programming errors before the application gets to the client. This fact leads to a series of advantages like: respecting deadlines, reducing exploiting and maintenance costs, reducing the possibility for the system to not face the demands of the user;
- UML unlike other methods permits a continuous developing-designing process and vice versa;
- the user is continuously in contact with the designer team, which was not possible with other methods, where the user saw the system only in the end.
- it is an expressive language;
- it is extensible, it permits the continuous adaption to demands of the domain it is built for;
- despite its flexibility, it is a strict language with well defined protocols;
- UML is easy to use;
- Cloud computing systems work intensely with asynchronous events. UML has instruments of high expressivity that enables the modeling of systems;
- once finished, a usage case can be launched due to object oriented paradigm.

3 Specific Aspects to Be Taken Into Consideration When Designing a Cloud Computing System According to the Aspects of Designing a Classic System

A lot are in common between designing classic and cloud computing systems. Similarities derive from the use of common methodology of building and designing these systems. For example, the diagrams of the usage cases have to be made, classes have to be identified with existing relation, in both cases. Differences induced by the process are due to its special character of a web based data and information exchange.

In case of classic systems, data and information exchange takes place in the same informational system, the respective data and information staying behind the organizational boundaries. In case of an cloud computing system data and information will exceed these boundaries [1] [8].

Components among which the informational flow takes place in a classic system belong to the same entity and are usually designed by the same team of designers. In case of cloud computing systems the components belong to different organizations and are designed by different teams.

The technology of building an cloud computing system presents a high level of homogeneity and is certainly ensured of compatibility and interoperability. It is not probable for the technology used in the case of an cloud computing system to be natively homogenic and less probable is natively compatible and interoperable.

In the designing of a classic system, redesigning comes after a certain period of time, after it can no longer face the tasks of the domain it was built for. In case of an cloud computing system designing is a continuous process. An cloud computing system has a special dynamism unlike classic systems. The cloud computing system evolves and changes with the speed of the web [1] [8].

The information consumers in the case of a classic system are known from the beginning. In case of an cloud computing system these are not known before and even if they are they present a high grade of dynamism.

4 Aspects of the RUP and XP Methods Applicability

From analyzing RUP methods characteristics the following conclusions are brought having in scope cloud computing systems [7]:

- RUP is a method that permits developing systems with a flexible and extendible architecture. Cloud computing systems are these kinds of systems so they comply with these demands;
- RUP puts accent on dealing with aspects of potential risk in time. This characteristic is a plus for every system;
- it doesn't imply a fix set of tasks in the initial stage so they can be refined as the project evolves. From the point of view of an cloud computing system tasks can't be specified in the first stages so this characteristic is in favor of cloud computing systems;
- RUP put accent on the final product and on the conformity of this with the final users' demands. This is clearly an advantage of cloud computing systems; etc;
- RUP takes over the advantages offered by UML. For a lot of IT systems this represents an advantage;
- RUP makes possible to control the quality of the developed system. The systems quality is in close relationship with its reliability. The better the quality the more reliable is the system. This characteristic brings an advantage to developing an cloud computing system;
- the time under which a system is developed using RUP is much less than in case of other methods, this is considered an advantage;
- RUP when it is adopted it becomes a repetitive and predictable process for the developing team. This leads to a high efficiency in case of developing large and reliable software.

From analyzing XP's applicability in the development of an cloud computing system it promotes the following practices [7]:

- *Planning*: involves a tight interaction between the client and the programmers. Programmers estimate the effort necessary

and the client decides the porpoise and the frequency of launching prototypes. This practice advantages smaller cloud computing systems like electronic market or provisioning but disadvantages types like administrating the flow of activities;

- *Delivering versions in short time intervals*: a system is delivered and then quickly updated to better and better versions. This practice is very popular among software companies; a proof for this is the number of patches launched to resolve problems. For an cloud computing system strategy like this can be applied for all presented architectures;
- *Developing systems is made on the base of a single description that explains how the system works*: a practice like this is hard to apply on cloud computing systems; etc.

5 The Agile Method and Mobile Cloud Computing

The future of the informational economy in a company consists in the ability of setting the pace, creating changes, but more important is the involvement of capacities for system development. In a volatile environment engaging continuous changes, severe traditional methods for system development are not sufficient for success. Therefore, the designers must be receptive, able to achieve their goal while in progress – to cut a long story short – to be agile.

The Agile software development is conceived to solve the need for speed and flexibility. The agility describes a global and collaborative activity that is able to create and answer changes, focusing on adjustments over a foreseeable future, person over process [34].

Agile development methodologies and Mobile Cloud Computing harmonize well with each other. Cloud services promptly meet the requirements and the entire development process is transparent to the end user. Feedback is requested in the developmental stages and changes are made accordingly.

Agile methods together with Mobile Cloud Computing offer an extremely interactive environment between developer and customer:

- users can provide immediate feedback through mobile cloud services
- developers will receive rapid feedback and thus react by fulfilling customers' expectations.

The Agile approach handles changes quite as a standard, instead of treating it like unusual issues, and its methods normally manage it, not like additional tasks. Changes' control is a part of everyday's project team work and this does not require a set of additional actions.

Some of the key benefits for using Agile methods in mobile cloud computing are as following:

- Costs reduction;
- Establishing a quality system for client
- Quality feedback;
- Prompt feedback;
- Shortened cycle-time development with 75%;
- Reduces the elapsed time between making a decision to seeing the consequence of that decision;
- Replaces documents with talking in person and whiteboards usage.

6 Database Refactoring

A database refactoring represents a change in the internal structure of a system, in order to make it easier to understand and modify without changing its noticeable behavior. It is mainly characterized by the following elements:

- The programmers restructure the system devoid of altering its performance so as to eliminate overlaps, develop the communications, simplify or append flexibility,
- Design improvement without negative effects on its functionalities,
- Communication improvement, simplicity, flexibility,
- Avoiding duplication,
- Avoiding inefficiency codes,
- Ever changes lead to a under optimum code structure, therefore the complexity will grow and transparency will fall;
- Changes may introduce new deficiencies, therefore database refactoring must be made at small levels and there must be

tests to indicate the places where errors appear;

- Database refactoring does not refer to the introduction of new functionalities - the two activities must be separated;
- Database refactoring improves the structure not the code, and a bad written code must not be re-factorized, but re-written;
- Database refactoring must not introduce new levels of useless complexity;

By means of careful using database refactoring, the behavior of the system will be the same, but with a well-structured design.

A *database refactoring* [2][3] represents a simple transformation brought to a database schema to develop its design while preserving both its behavioral and informational semantics – in other words, you cannot add new functionality or break existing functionality, you cannot add new data, and you cannot change the meaning of existing data.

There are two fundamental reasons behind the wish for adopting database refactoring [3]:

1. *To repair existing legacy databases*: Database refactoring enables you to safely develop your database design in little steps, making it a central method for improving the legacy assets within an organization.
2. *To support evolutionary software progress*: Modern software development processes are all evolutionary through their

nature, including the Rational Unified Process (RUP), Extreme Programming (XP), Agile Unified Process (AUP), Scrum, and Dynamic System Development Method (DSDM).

As an outline, we can affirm that database refactoring represents a database implementation technique, just like code refactoring is a system implementation technique.

7 Proposed Methodologies with Agile Model Driven Development (AMDD)

As its title suggests, AMDD is the agile version of the paradigm of MDD (Model Driven Development). Through this MDD method, a new assembly of system models is being built, which generates other models or the source code of the system.

The method is based on abstracting, and allows the accomplishment of the conceptual view of the system, focusing on the system functions without implementing its specifications [4].

When referring to AMDD, the difference consists in creating new agile models, which are fairly good enough to lead the general efforts of development. The Agile approach is different from the traditional approach of development, in which a new model of a project is being created, and then the implementation begins on its ground [22].

The steps that we propose, based on AMDD concept, are illustrated in Figure 2.

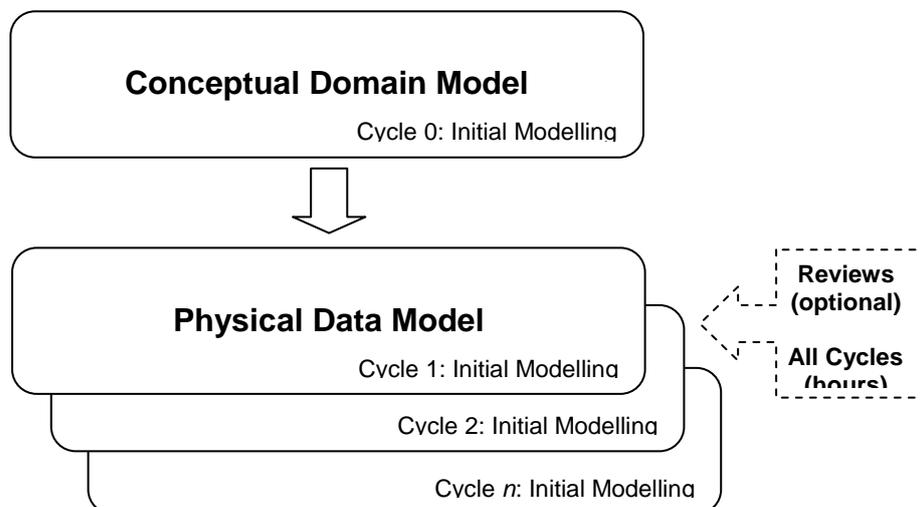


Fig. 2. The proposed steps in designing our database

The first step represents the development of a conceptual domain model. It is a simple one containing the main parts of the business and the relationship between them. The conceptual domain model will be used to make the physical data model.

Some of the AMDD main benefits are as following:

- *Project planning*: if the requests were identified and the architecture was established, you could get enough information to value the costs and to produce a schedule of the ongoing project;
- *The management of the technical risks*: the initial efforts in modeling the architecture help to identify the section with major technical risks;
- *Minimizing the useless products*: a JIT approach on modeling avoids modeling some parts that are useless in the future;
- *Asking more pertinent questions*: the later the request is modeled, the domain could be known better, and the questions asked will be able to clear the confusion;
- *Obtaining precise answers from the stakeholders*: in a similar way, the stakeholders will have a more accurate image on the system, as they get a view on the functional system at regular intervals

It is important to mention that the AMDD method can be used for both small and also large projects.

8 Case Study: Designing a Database for a M-Commerce System in Cloud

In this section we present our view on building a database for a virtual mobile commerce

shop in the cloud. We considered the following scenario: company trading computer components (buy and sell).

A client using his/her smartphone checks the existent offer arranges the products on categories and orders them. To be able to order certain products, the client must sign up on that site, by entering an e-mail address and a password – which will be recorded in the database. One person may order one or more components. The products will be sent via postal or courier services and the payment will be made at delivery.

In addition, the site administrator introduces new articles in the offer and is able to revise the information of the existent products.

The main principles behind the project are as follows:

- Minimizing the human errors;
- Increasing efficiency;
- International development.

The focus is on the quality of the goods and provided services and not on the management of Human Resources departments, bill payments or other actions.

When referring to this methodology, its first step represents the development of a conceptual domain model. It is a simple one, containing the main parts of the business and their relationship. The conceptual domain model will be used to create the physical data model. Figure 3 presents the domain model for our application's database.

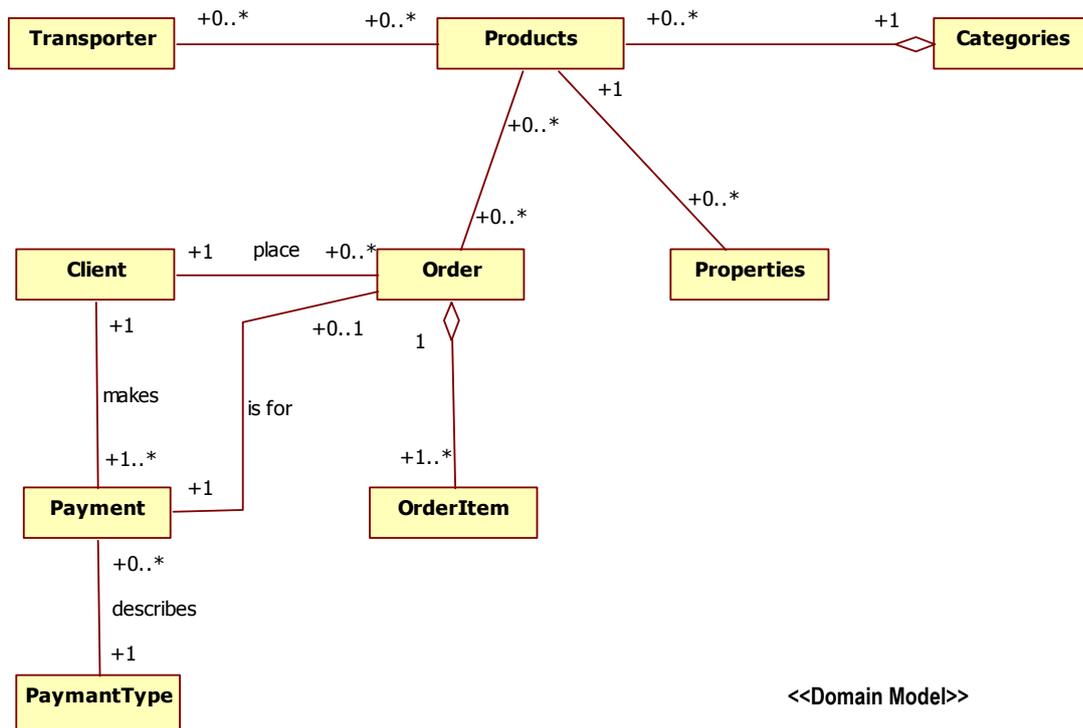


Fig. 3. The Initial domain model

The first iteration of the physical model will contain the most important functionalities of the system, in our case the customer manag-

ing, the type of payment possible in the system, the products and their order. Figure 4 graphically represents this iteration.

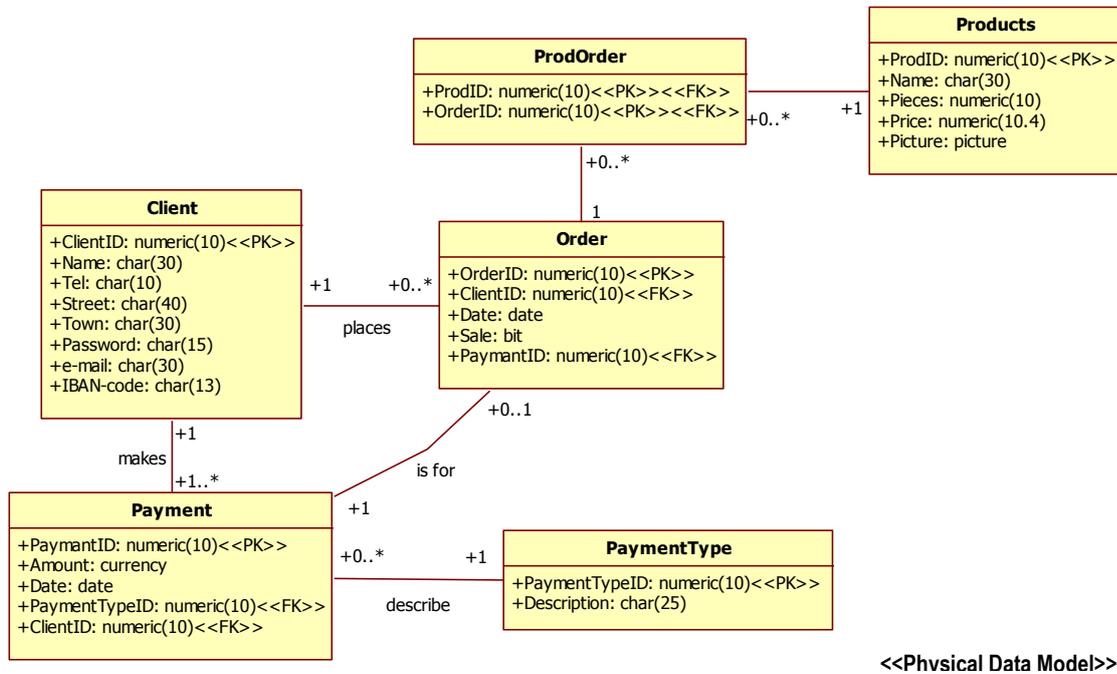


Fig. 4. The first iteration of the database

The second iteration has little developments over the existing database, and, as we can see

in Figure 5, the changes were relatively minor.

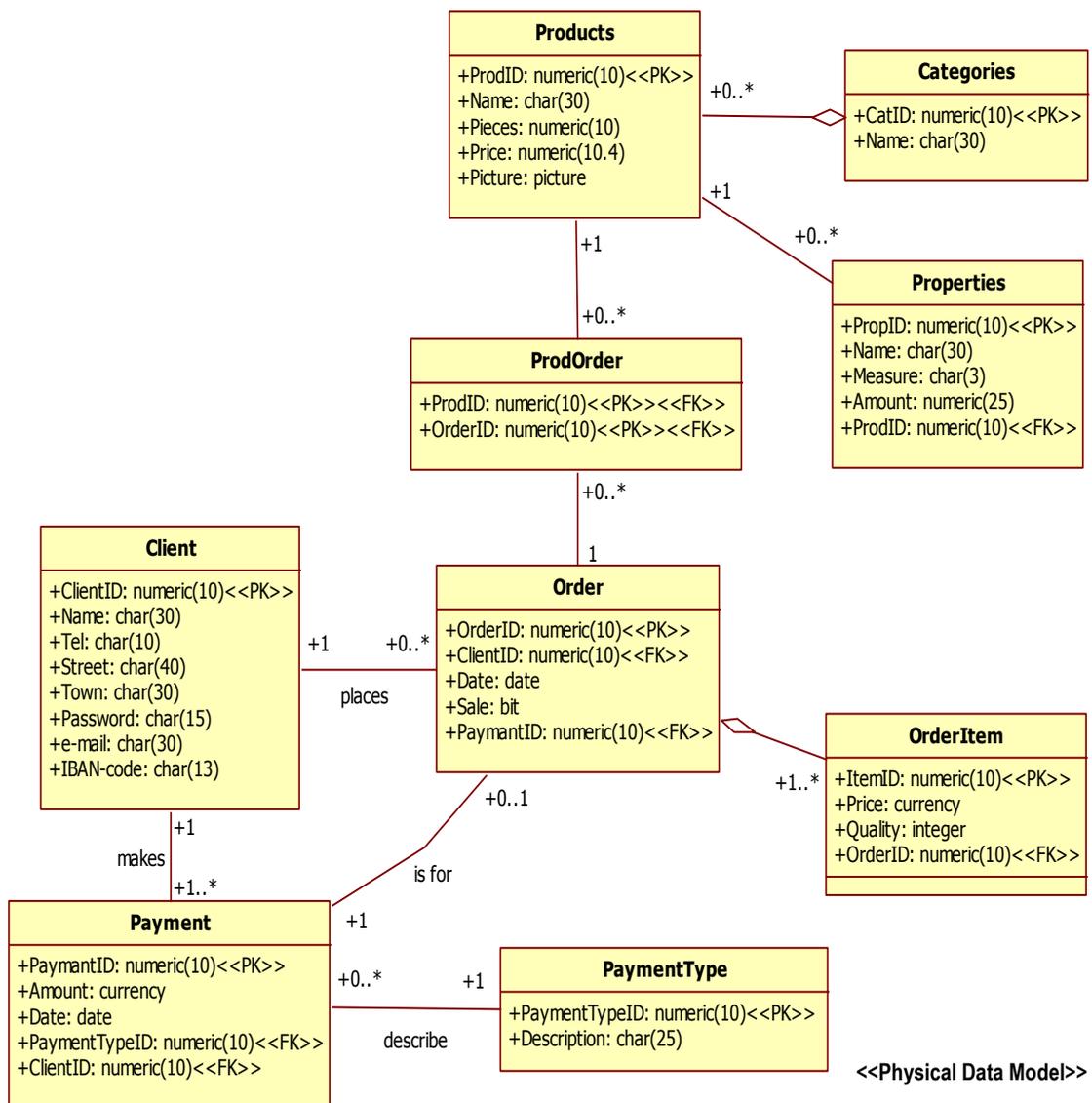


Fig. 5. The second iteration of the database

For the last iteration we also took into consideration the Transporter, who will transfer the required products - this table was added right to the database scheme. The final physical model of the database can be observed in Figure 6.

Using these iterations represents first-class basis for the development team, because they offer good opportunities for a real feedback (if necessary) based on the utilization of the system while in progress. If required, the database could be further modified using new iterations.

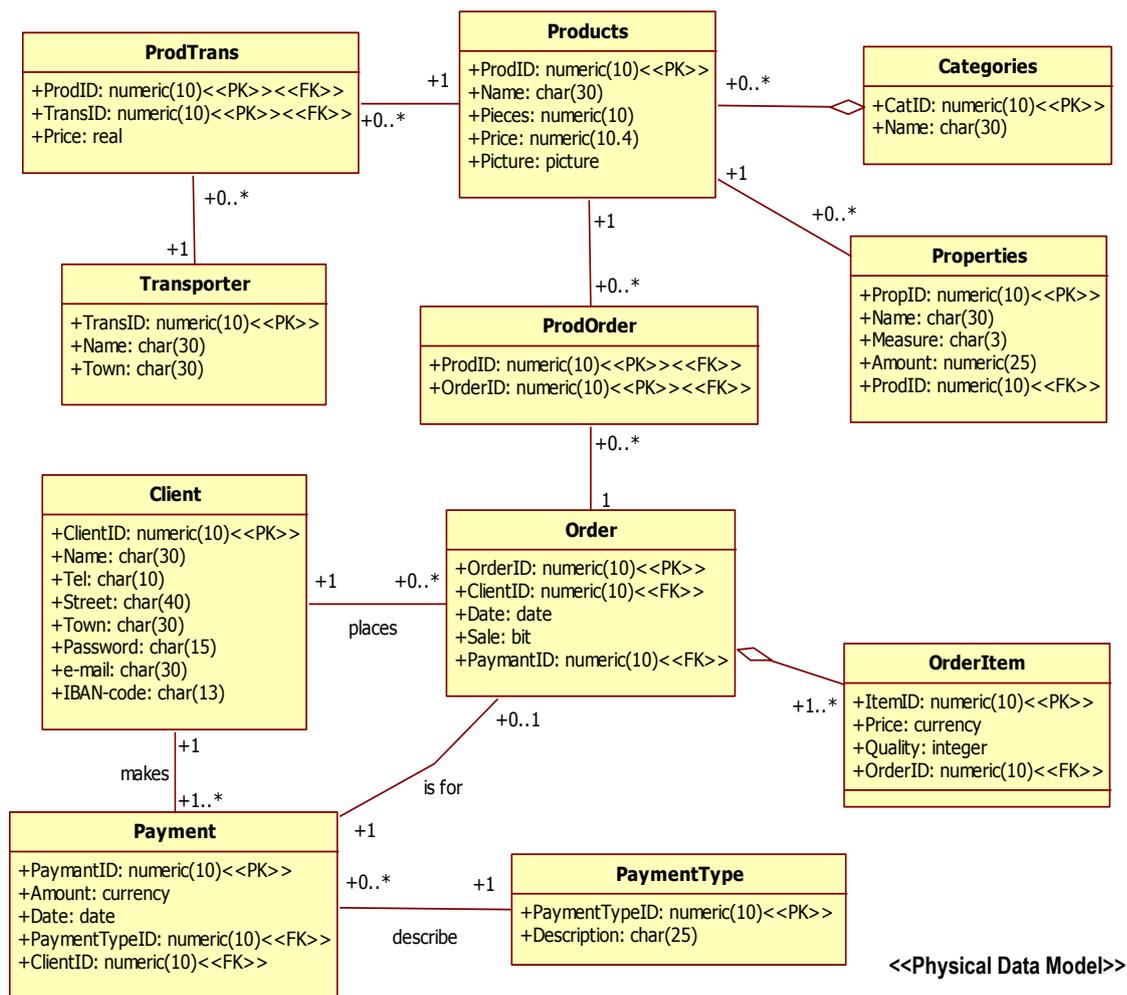


Fig. 6. The final physical model of the database

8 Conclusions

Mobile cloud computing, as a development and extension of mobile computing and cloud computing, has inherited the high mobility and scalability, and became a new research topic in recent years.

The Agile approach is used for a variety of systems, including but not restricted to: web database application, mobile database applications in cloud, customer-server database applications, Business Intelligence systems, integrated software, life critical systems or even mainframe database applications. By means of utilizing the Agile method for our database design, we are able to develop the physical model in order to answer any future requests, we can focus on the domain model and we can better comprehend the project's objectives just from the beginning.

When referring to mobile applications in cloud computing, a further development of the system is essential as well as developing as many fields as necessary. This method allows a continuous development of the database as expected by the developing team.

Acknowledgements. This work was partly supported by the CNCS TE_316 grant, manager Vasile Paul Bresfelean.

References

- [1] S. W. Ambler, "Agile software development at scale." In B. Meyer, J. R. Nawrocki, & B. Walter (Eds.), *In Proceedings of CEE-SET 2007*, LNCS 5082, IFIP International Federation for Information Processing, 2007, pp. 1-12.

- [2] S. W. Ambler, "The Object Primer 3rd Edition: Agile Model Driven Development with UML 2." *New York: Cambridge University Press*, 2004
- [3] S. W. Ambler, and P. J. Sadalage, "Refactoring Databases: Evolutionary Database Design." Boston: *Addison Wesley*, 2006
- [4] A. Anand Rao, K. Madhavi, "Framework for Visualizing Model-Driven Software Evolution and its Application", *International Journal of Computer Science Issues*, Vol. 7, Issue 1, No. 3, January 2009
- [5] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, et al., "Above the Clouds: A Berkeley View of Cloud Computing." *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28*, 2009
- [6] C. Avornicului, M. Avornicului, "Aspects of using MDA paradigm in sistem development", *Proceeding of International Conference on Business Information Systems*, Iași, 2006
- [7] C. Avornicului, M. Avornicului, "Managementul și proiectarea sistemelor informatice de gestiune", *Editura Risoprint*, Cluj-Napoca, 2010
- [8] M. Avornicului, „Proiectarea și managementul sistemelor informatice”, *Editura Abel*, 2010
- [9] M. A. Babar and P. Abrahamsson, "Architecture-centric methods and agile approaches." In P. Abrahamsson, M. Marchesi, & F. Maurer (Eds.), *XP 2009, Lecture notes in business information processing (LNBIP) 31*, Springer-Verlag, Berlin Heidelberg, 2009, pp. 232-233.
- [10] C. Batini and et. al., "Conceptual Database Design", Redwood City, CA: *Benjamin/Cummings Publishing*, 1992
- [11] B. Bernd, D. Allen, "Object-Oriented Software Engineering: Using UML, Patterns and Java", 2nd Edition, *Prentice Hall*, 2004
- [12] M. Bouzeghoub et. al. "Object Technology – Concepts and Methods", *International Thomson Computer Press*, Boston, 1997
- [13] D. Brown, "An Introduction to Object-Oriented Analysis: Objects in Plain English", *John Wiley&Sons, Inc.*, New York, 1997
- [14] T. Dingsoyr, T. Dyba and P. Abrahamsson, "A preliminary roadmap for empirical research on agile software development." *Proceedings of Agile 2008*, Washington, DC, USA, IEEE Computer Society, Los Alamitos, 2008, pp. 83-94.
- [15] G. Dragan, D. Dragan and D. Vladan D. "Model Driven Architecture and Ontology Development", *Springer*, 2006
- [16] European Commission – The Future of Cloud Computing report, [Online] (last viewed 2013, February 21) <http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf>
- [17] D. S. Frankel, "Applying MDA to Enterprise Computing", *Wiley Publishing Inc.*, Indianapolis, 2003
- [18] C. Gane, "Computer Aided Software Engineering", Rapid System Development Inc., Englewood Cliffs, *Prentice Hall*, New Jersey, 1990
- [19] E. Hadar and G. M. Silberman, "Agile architecture methodology: Long term strategy interleaved with short term tactics." *Proceedings of Object oriented systems languages and applications (OOPSLA'08)*, Nashville, Tennessee, USA, ACM publication, 2008, pp.641-651.
- [20] T. K. Hazra, "MDA brings standards-based developing Modeling to EAI Teams", *Application Development Trends*, [Online] (last viewed 2013, February 21) <http://www.techopedia.com/definition/26679/mobile-cloud-computing-mcc>, 2002
- [21] A. Kennedy, "Model Driven Architecture and Executable UML – The Next Evaluatiopnaryx Steps in System Development?", *Szoftvertechnológiai Fórum*, Budapest, 2004
- [22] K. Korhonen, "Migrating defect management from Waterfall to Agile software development in a large scale multi-

- site organisation: A case study.” In *XP 2009, LNBIP 31*, pp. 73-82, Springer-Verlag Berlin Heidelberg, 2009
- [23] R. Martin, ”Agile software development principles, patterns and practices”, *Prentice Hall*, 2003
- [24] M. Blaha and J. Rumbaugh, „Object-Oriented Modeling and Design with UML”, 2nd Edition, *Prentice Hall*, 2005
- [25] R. Muller, ”Using UML for Data Modeling”, *Morgan Kaufmann Publishers*, 1999
- [26] E. Naiburg and R. Maksimchuk, ”UML for database design”, *Addison-Wesley*, 2001
- [27] D. Oprea, ”Analiza și proiectarea sistemelor informatice economice”, *Editura POLIROM*, Iași, 1999
- [28] D. Parsons, H. Ryu and R. Lal, “The impact of methods and techniques on outcomes from agile software development projects.” *IFIP International Federation for Information Processing*, vol 23, Organizational Dynamics of technology-based innovation: Diversifying the research agenda, (Boston: Springer), 2007, pp.235-249.
- [29] M. Raffai, "Az UML 2 modellező nyelv" Szakkönyv Kiadás alatt, *Palatia Kiadó*, Győr, 2005
- [30] S. Ramakrishnan, “Innovation and scaling up Agile software engineering projects”, *Issues in Informing Science and Technology*, Volume 6. [Online] (last viewed 2013, February 21) Available at <http://iisit.org/Vol6/IISITv6p557-574Ramakrishnan573.pdf>, 2009, pp. 557-574
- [31] T. Saridakis, “A system of patterns for fault tolerance.” In *Proceedings of the 7th European Conference on Pattern Languages of Programs (EuroPLOP)*, 2002 pp. 535-582.
- [32] R.J. Schaaf, ”Agility XL.” *Systems and Software Technology Conference 2007*, Tampa, FL
- [33] Schwalbe, Kathy, *Information Technology Project Management*, Course Technology, 2010
- [34] Technopedia - Janssen C., Mobile Cloud Computing (MCC), (last viewed February 20, 2013)
- [35] TechSoup Global – 2012 Global Cloud Computing Survey Results, [Online] (last viewed 2013, February 21) <http://www.techsoupglobal.org/2012-global-cloud-computing-survey>
- [36] The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology, [Online] (last viewed 2013, February 21) <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [37] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, ”A break in the clouds: towards a cloud definition.” *ACM SIGCOMM Computer Communication Review*, 39(1), 2008, pp. 50-55.
- [38] L. Youseff, M. Butrico and D. Da Silva, ”Toward a Unified Ontology of Cloud Computing. In *2008 Grid Computing Environments Workshop*, 2008, pp. 1-10.



Silviu-Claudiu POPA has graduated Faculty of Economics, Babeș-Bolyai University Cluj-Napoca in September 1997. He has finished his PhD studies in 2003. He works at the Business Information Systems department of Faculty of Economic Science and Business Administration, Babeș-Bolyai University Cluj-Napoca. His current position is PhD Lecturer.



Mihai-Constantin AVORNICULUI is a Lecturer PhD. He has graduated Faculty of Mathematics and Computer Science, Babeș-Bolyai University Cluj-Napoca in 2004. He has finished his PhD studies in 2009. He works at the Business Information Systems department of FSEGA, Babeș-Bolyai

University Cluj-Napoca. His research interests include data mining, cloud computing, information systems, and advantage database systems.



Vasile Paul BRESFELEAN is a Lecturer PhD at Babeş-Bolyai University of Cluj-Napoca, Romania. His educational background includes: PhD in Cybernetics and Economic Statistics, Master of Science in Business Information Systems, Bachelor of Economics, at Babes-Bolyai University. Some of his fields of interest and didactic activities are: Data Mining, ERP, Databases, Decision Support Systems, eLearning. He is author and co-author of 15 articles indexed in ISI Thomson database, and several other international and national publications, and presently the manager of a national PNII_TE research grant.