



Object database for your prototypes

Published date	April 2006
Version	1.0
Author	Paul Perez (Pymma)

Pymma group 22 Chestnut Avenue, Rickmansworth Herts WD3 4HB United Kingdom

Contact UK +44 79 44 36 04 65

Contact FR +33 6 99 36 07 10

Contact NL + 31 621 273 973

Email contact@pymma.com

Web site www.pymma.com

Table of content

Introduction.....	3
Set up of a prototype: challenges.....	3
Preoccupations with an installation.....	3
My kingdom for one DBA!!!.....	4
The headache of mapping O/R.....	4
Solutions for the installation of a prototype.....	4
Use of a object database for a prototype.....	4
Benefit of using a database.....	5
How to make use of an ODB?.....	5
Small additional work.....	5
Lack of experience.....	6
Conclusion.....	6
References.....	7
Editors of Java databases	7
Recommended sites.....	7
About the author.....	7

Introduction

Before starting a data-processing project, or in the course of a project, it is frequently required to write mini applications which one calls prototypes. A prototype is an application constructed in a limited time, by a small team. The goal of this work is, for example, to test "experimentally" new technologies (technical prototypes), to evaluate the rise in load of a product (benchmark), to test new concepts of programming (evolutionary prototype). The prototypes are generally classified in the following way:

	Evolutionary	Disposable
Technical		
Application		

In short, the goal of this approach is to familiarise oneself with something new and to obtain experience with this innovation. On the management level, a prototype is regarded as a tool making it possible to evaluate a technology theoretically and thus to minimize the risks taken. But alas in reality, prototypes are constructed with a very limited budget because it is rare that managers appreciate the true value of the experience feedback obtained by the realisation of prototypes. Consequently, the time assigned for these studies is always very course.

Set up of a prototype: challenges

Preoccupations with an installation

Even before working concretely on the prototype object, many additional activities must be finished:

- installation of the environments for development and testing
- installation of the databases
- definition of the mapping between the object languages (Java, C++, C#) and the database.

To install means to create and parameterize a database or to access an already existing database. The majority of our customers use Oracle, IBM or Microsoft databases managed by teams of administrators. It is often necessary to call upon a DBA (Date Base Administrator) to install, configure and optimize the database. If the results of your prototype are disappointing and you did not call upon a DBA, you will always be entitled to the reflection: "Your database was not optimized, that is why it does not work". By past experience, I can affirm to you that even if your team is full of very experienced people having a perfect command of the relational database you will expose yourselves to this type of criticism.

My kingdom for one DBA!!!

This fact would not be serious if it were easy to meet an available DBA in the information services departments quickly. I knew several companies where systematically, it took 10 days to obtain an appointment with one DBA. For them, the problems of setting in production and maintenance are much more important than your prototype. It is something which I understand easily. But when you have 15 days or 3 weeks to produce your prototype, and it takes you one week to have access authorizations and a base of correctly configured data, that soon becomes an impossible mission.

The headache of mapping O/R

Another related activity to relational databases is summarised with a figure - "30%". It is the average time which a project passes by to solve the problems of mapping when it uses an object language and a relational database. This figure varies according to the complexity of your trade model (market room, computer-integrated manufacturing, air transport) and of the tools which you use to define your mapping (TEENAGER, JDO, EJB3...). To lose between 5 and 10 days out of 15 to configure and use a database correctly and then to define the mapping to a relational object whereas, one actually seeks to test the performances of Web services exchanges with a partner, is a pity.

Solutions for the installation of a prototype

Use of a object database for a prototype

I do not seek to include in this paper, the an argument for the old debate "relational database VS object relation database". Here the goal is to optimise the time assigned with a prototype and to devote this only to the objective laid down by management.

To reduce the problems involved in persistence and to keep to schedules, we use, with the agreement of our customers, object databases for the management of persistence in the prototypes. Of course that has uses only when persistence is not the subject of the prototype or database legacy does not intervene with the prototype. Then there are no problems of integration with the information system of the company preventing the use of an object database.

Benefit of using a database

The installation of the object databases (ODB) in the majority of the cases is very simple and the installation programs make this operation accessible to all developers. An object database (ODB) natively stores the objects of a Java application (C++ C#..). Here an example: when an invoice is being recorded in the database, it is not necessary to create tables to store invoices, customers and articles, nor to create additional tables representing relations N-M. To store an invoice in an ODB one only command "save" is enough and the tree structure of objects related to the invoice (customer, article...) is safeguarded. In the same way, a command "retrieve" is enough to recover the complete tree structure of the invoice. There is no mapping of either the management of the relations between entity nor of definition of a foreign key. Nothing of what makes mapping nightmarish is present here.

How to make use of an ODB?

Small additional work

What I describe here is an ideal situation because persistence objects always requires a small amount of additional work on behalf of the programmer. This work makes it possible for Java objects to be persistent. For example, that can be a Post-Compilation. (Process consisting in compiling the Java code a second time to modify Byte Code and to add methods of persistence) or obligatory implementation of an interface or the definition of objects in files of external configurations. At certain times, it is necessary to use a JVM modified by the editor of the ODB. This complementary work is a function of the technological choices of the editors of the ODB and allows a differentiation between the products. But in any event, this work is much lower than that required by the relational databases. Moreover, it is very simple to make this work generic and reusable between all the prototypes. Lastly, this work does not require knowledge trades or applications.

Lack of experience

Even if some consider that databases are the best means of storing Java objects, it should be admitted that this technology is not one of most common with programmers who after years of SQL have difficulties in understanding how to seek data without requests. How can one seek the products bought by a customer last month without SQL. To seek data in an object field, one uses a technique of research called "Navigation". Navigation makes it possible "to sail" from object into object to cross the lists of object very quickly to find desired information. Like all the techniques of programming, to be effective, navigation requires training and experience in the definition of object model. A technological survey and some practical exercises, a training course will help to cross this first bridge.

Conclusion

The realization of a prototype regularly encounters problems involved in the installation of the relational databases and on the mapping of relational objects. As a result, this activity consumes between 20% and 40% of the time assigned with a project. When the prototype is disconnected from the infrastructures of the information processing system of the company, the use of an object database becomes possible. Thus with little means and time, any team of developers can make itself very effective and devote itself entirely to the realization of the prototype.

Many object databases have licences of free use. The others propose initial versions for very accessible prices. A little training will enable you to control the techniques of navigation.

The more complex your trade field will be, the more the ODB will be invaluable for you to save time. The more DBA's are in demand the more you will appreciate the facility of administration of these products.

Last points, once your prototype is finished, once your management is satisfied by the excellent results obtained and the effectiveness of your team, never make the error to propose your ODB for a real application or a production setting, you would then lose any credibility... unless a miracle happens!

References

Editors of Java databases

Gemstone	www.gemtone.com / www.facetsodb.com
Versant	www.versant.com
Objectdb	www.objectdb.com
Db4Object	www.db4o.com
mcObject	www.mcobject.com
Objectivity	www.objectivity.com
Caché	www.intersystems.com

Recommended sites

Object data management group	www.ondg.org
Object database management systems	www.odbms.org
Barry & Associates, Inc	http://www.service-architecture.com/object-oriented-databases/

About the author

Paul Perez is co-founder and Chief Software Architect of Pymma. Paul is an author, software architect consultant and speaker, brings more than 17 years experience helping corporations utilize object technology for mission-critical information systems. Prior to co-found Pymma, Paul was an independent software consultant, working for large financial services companies in France, UK, Benelux, Israel and Germany. Paul's extensive experiences in high-performance architecture design helps the company's clients solve real-world business problems through technology. As chief software architect, Paul assists current and future client engagements and develops best practices based on his domain expertise. Paul holds a post master degree from Paris-Orsay University and is a Sun Certified Enterprise Architect and holds several other certifications