Extending the Object-Process Methodology to Big Data Systems

Tao WANG, Feng YANG, Yi-fan ZHU, Xiao-bo LI and Xin ZHOU

College of System Engineering, National University of Defense Technology, China

*Corresponding author

Keywords: Data Description, Time Constraints, Concurrency Management, Status Management.

Abstract. There is a general consensus in the software literature that big data systems are difficult to model, specify, and design. It is an important and challenging task to develop an intuitive and easy-to-use, yet coherent and concise method for specifying such systems. The Object-Process Methodology (OPM) graphically specifies systems in an integrated model that describes the static-structural and behavioural-procedural aspects of a system using Object-Process Diagrams (OPD), we present OPM/D as an extension of OPM for specification of big data systems. Focusing on the four phases of the value of big data, we extend OPM in 4 points: data description, time constraints, concurrency management and status management. A detailed example denotes the accuracy, simplicity and expressiveness of OPM/D to express big data applications.

Introduction

Over the latest 20 years, data increased in a large scale in various fields. According to a report from International Data Corporation (IDC), 2011, the overall created and copied data volume in the world was 1.8ZB, which increased by nearly nine times within five years\(^1\), we are awash in data.

With the explosive growth of global data, the term Big Data is used to describe the enormous datasets, and it is applied broadly in many fields, such as marketing and advertising, healthcare, life sciences, financial services, education, etc. Unlike the traditional datasets, big data focuses on semi-structured or unstructured data that need more real-time analysis. Big data brings new opportunities for discovering new values; on the other hand, it also incurs new challenges, especially on the development of big data applications.

Nowadays, organizations may use big data to create value in five ways. First, organizations can use data to develop a better understanding of customers and tailor product and services for narrowly defined segments. Second, organizations can use data to monitor performance of key functions, identifying factors contributing to observed variances and pointing to needed remedial actions or ways to optimize systems. Third, organizations can use data to predict behavior or forecast events, and as a result, take appropriate action. Fourth, organization can use data to meet regulatory compliance or legal discovery requirements. Finally, organization can use data as the building blocks for new products and services\(^2\).

For the reason of the high potential of big data, over the past few years, many government agencies announced major plans to accelerate big data research and applications. In academia, issues on big are often covered in public media, such as The Economist, New York Time. Nature published a special issue on big data in 2008. Science also launched a special issue about the key technologies of "data processing" on big data in 2011. As for industries, nearly all major information technology companies, including IBM, Microsoft, Google, Oracle, EMC, Amazon, and Facebook, etc. have started their own big data projects\(^3\).

Big data, big challenge, although the research community on big data has proposed some solutions from different perspectives, they still present an integrated method to specify big data applications. To solve this problem, we present a unified framework OPM/D—an extension to the Object-Process Methodology (OPM)\(^4\). The paper is organized as follows. Section 2 briefly surveys specification methods for big data systems. Section 3 introduces and demonstrates the basis of OPM and OPM/D. Section 4 concentrates on using OPM/D specifying the four phases of the value chain of big data, i.e.,
data generation, data acquisition, data storage, and data analysis, which demonstrates the accuracy, simplicity and expressiveness of OPM/D.

Specification Methods for Big Data Systems

The sharply increasing data deluge in the big data era brings about so many challenges on data acquisition, storage, management and analysis. Traditional technology on data management and analysis generally is based on the relational database management system (RDBMS). However, the RDBMS generally apply to structured data, other than the semi-structured or unstructured like web pages, videos, audios, etc. We have to face the challenge that the traditional RDBMSs could not meet the needs of big data. Industries and research communities have given some solutions from different perspectives. Such as cloud computing is used to meet the requirements on infrastructure for big data, distributed file systems and NoSQL databases are used for the permanent storage and management of large-scale disordered datasets.

In spite of the special characteristics of big data systems, now the common methods for specifying big data systems are still UML—a comprehensive language served as a general-purpose, standardized modeling language for object-oriented analysis and design. UML uses a set of diagrams to specify a system from multiple views such as requirements view by using use case diagrams, structure view (by using package diagrams, component diagrams, class diagrams etc.), behavior view (by using activity diagrams, interaction diagrams, etc.), and implementation view (by using deployment diagrams). An additional textual language, the Object Constraint Language, is also provided with UML for expressing static consistency constraints on sets of objects and their interrelations.

OPM is a holistic system approach that integrates the structure and behavior of a system within a single frame of reference. The building blocks of OPM are things (objects and processes), structural relations, and procedural links. A thing is a generalization of an object and a process—the two basic building blocks of any system expressed in OPM. Structural relations (or simply relations) express structural connections (e.g., aggregation, generalization, and characterization) in order to model the static view of a system. Procedural links (or simply links) connect objects and processes to describe the dynamic behavior of a system—how processes transform (generate, consume, or affect) objects.

In OPM, there are two equivalent specification notations jointly express the same OPM model. A visual one through Object-Process Diagrams (OPDs), and a textual one through the Object-Process Language (OPL) to abstraction and refinement are carried out through two mechanisms: unfolding/folding of thing hierarchies, mainly used for managing the complexity of object structure, and zooming-in/zooming-out, largely used for detailing and hiding the internal dynamics and the interface of processes.

Modeling of complex systems should conveniently combine structure and behavior in a single model. Motivated by this observation, OPM is a comprehensive, holistic approach to modeling, study, development, engineering, evolution, and lifecycle support of systems. Employing a combination of graphics and a subset of English, the OPM paradigm integrates the object-oriented, process-oriented, and state transition approaches into a single frame of reference. Structure and behavior coexist in the same OPM model without highlighting one at the expense of suppressing the other to enhance the comprehension of the system as a whole.

Rather than modeling separate views for each of the system’s aspects and integrating the various views mentally, OPM presents an approach that is orthogonal to customary practices. According to this approach, various system aspects can be integrated for better comprehension.

Simulation and Discussion

Although OPM is a promising candidate for modeling big data systems, it is still not fully suitable to satisfy some big data application requirements, such as dynamic architecture, dynamic algorithms,
security and privacy management, etc... OPM/D, which is built on the foundation of OPM and hence inherits all of its capabilities, extends its expressiveness for specifying big data systems, in particular. This section presents the extensions of OPM that are included in OPM/D, and the next section evaluates and compares them to an UML extension for the big data systems.

**Data Description**

The first thing for a big data system is describing the data to be used. In the object-oriented paradigm, data always are modeled as object classes. For example, the Object Constraint Language (OCL), which was designed to accompany UML models, is an expression language that enables describing constraints on object-oriented models and other object modeling artifacts.

In OPM/D, the corresponding data of an object are presented using some sequential panes as Figure 1, which describes both data name and type. At the same time, the data relation among objects can be controlled through a process, which automatically affects the objects that are derived from other objects. As shown in Figure 1, it contains 3 main objects: Commodity, Customer, and Statistics. The object Commodity stores the details of commodity, such as Name, Price, and Producing area. The object customer stores the information of customers: Name, Address, and Phone number etc. The data relation between Commodity and Customer is characterized using an object so-called Shopping-Bag, which stores the commodities that a customer selects, and it is changed by shopping. The object Statistics maintains statistical information about all shopping. Once a customer do some shopping, the statistics updating process will be activated automatically and change the object Statics. This is expressed by an event link between the object Shopping-Bag and the process Statics Updating.

![Figure 1. An OPD featuring the data description.](image)

**Time Constraints**

Time factor plays a very important role in big data systems, and the behavior of big data systems can’t be specified by just giving the inputs and outputs. There is a need to represent triggering events, guarding conditions, temporal constraints and timing exception. Among these time constraints factors, triggering events can be described using OPD directly, guarding conditions are specified using a recorded letter “c” within the circle of a link, and temporal constraints are specified in parentheses as an interval \((x, y)\), where \(x\) and \(y\) determines the lower and upper bounds of the constraints respectively. There three kinds of temporal constraints:

- Process duration constraints, in which the interval \((x, y)\) is recorded inside the constrained process, as shown in Figure 2(a).
- State duration constraints, in which the interval \((x, y)\) is recorded inside the constrained object state, as shown in Figure 2(b).
- Reaction time constraints, in which the interval \((x, y)\) is recorded above the procedural link connecting the triggering object to the triggered process, as shown in Figure 2(c).

\[
\begin{align*}
\text{(a)} & \quad \text{Process 1} \quad (x, y) \\
\text{(b)} & \quad \text{Object 1} \\
\text{(c)} & \quad \text{State 1} \quad (x, y) \quad \text{State 2} \quad \text{Object 1} \\
\end{align*}
\]

Figure 2. Expression of time constraints in OPD.

**Concurrency Management**

Big data systems are generally distributed, and the processes are always concurrent. In OPM/D, the concurrency and synchronization are expressed quantitatively, explicitly and unambiguously. This way, concurrent processes are located at the same height, while the sequential processes are located one below the other. Concurrency constraints are specified in parentheses as an interval \((x, y)\), where \(x\) and \(y\) determines the lower and upper bounds of the constraints respectively. There four kinds of concurrency constraints as shown in Table 1:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P_1)</td>
<td>Process (P_2) must begin within a time period ((x, y)) after the beginning of process (P_1)</td>
</tr>
<tr>
<td>(P_1)</td>
<td>Process (P_2) must begin within a time period ((x, y)) after the end of process (P_1)</td>
</tr>
<tr>
<td>(P_1)</td>
<td>Process (P_2) must end within a time period ((x, y)) after the beginning of process (P_1)</td>
</tr>
<tr>
<td>(P_1)</td>
<td>Process (P_2) must end within a time period ((x, y)) after the end of process (P_1)</td>
</tr>
</tbody>
</table>

**Status Management**

The objects of big data systems may have several statuses, and the variable status will invoke different processes and result in different function. State management includes the description of states and the transformation of states, as shown in Table 2.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{Object 1}) (\text{State 1})</td>
<td>Object 1’s initial state is State 1</td>
</tr>
<tr>
<td>(\text{Object 1}) (\text{State 1}) ((x, y))</td>
<td>Object 1 once change in State 1, it must transform to State 2 within a time period ((x, y))</td>
</tr>
<tr>
<td>(\text{Object 1}) (\text{State 1}) ((x, y))</td>
<td>After the State 1, Object 1 must change to State 2 within a time period ((x, y))</td>
</tr>
</tbody>
</table>

**The System ABC and Its OPM/D Specification**

We have introduced several key technologies related to big data systems and OPM/D. Next, we will focus on using OPM/D specifying the value chain of big data applications, which may be divided into 4 phases: data generation, data acquisition, data storage, and data analysis. This is very easy to present abstractly in OPM/D as Figure 2.
Figure 3. Top-level 4 phases of the Big Data system ABC.

Among these four phases, the most difficult one is the process “Data Analysis”. Now, we instanced it using an example analyzing users’ behaviors.

Figure 4. OPM/D specification focusing on the process Data Analysis of ABC.

Figure 3 is a simplified OPD of a big data system named “ABC(Analysis of Behaviors of Consumers)”, which uses some of the OPM/D extensions. The Commodity, Consumer, Statistics, Shopping-bag, Customer service staff are four object things, they are all physical objects, and their information are stored in database at the same time. The Shopping, Statistics updating, Serving and Data Analysis(including its sub-process) are four process things.

While Customer buys some Commodity, it will affect the object Shopping-bag, and triggers the process Statistics updating consequently, then process Statistics updating affects the informational object Statistics.

Once a customer finishes shopping, the state of object Customer will transform from “shopping” to “leaving”. The system ABC must begin executing the process Data Analysis among 10 seconds.
The Data Analysis process of the system ABC includes 5 sub processes: Business analysis, Consumer behavior habits analysis, Consumer loyalty analysis, Consumer satisfaction analysis and Rate of return. The process Business analysis is also combined of several sub processes: Products distribution, Consumer retentivity, Consumer loser rate and Updating/crossing consumption analysis. After all the sub processes of Data Analysis executed, the process Serving is invoked. The process Serving requires the object Customer service staff.

Although the complex system ABC includes many processes, Figure 4 shows that once a Consumer leaves, the system must begin applying services in 180 seconds.

Conclusions and Future Work

However, the big data system development is far from mature and recently there has been an increasing interest in finding new methods and establishing standards. There are multiple tools such as MapReduce, Hadoop, EC2, R, WEKA, Gephi, etc. to use in main four phases for big data resolving\textsuperscript{9,10,11}, yet it is still very difficult to model, design and specify an big data system. To meet the challenge, we develop an intuitive and easy-to-use, yet coherent and concise method named OPM/D based on OPM, which describes both the static structural and the dynamic-procedural aspects of a system in unified models. OPM/D’s extension includes data description, time constraints, concurrency management and status management. An example ABC is given to demonstrate how a big data system can be specified explicitly and communicated among analysts, designers, and developers.

Future work is needed to cooperate our findings and experiences with much more big data systems, and to present the details of the 4 phases: data generation, data acquisition, data storage, and data analysis. We also intend to develop a tool for OPM/D which will supports big data system lifecycle evolution\textsuperscript{12}

Acknowledgements

This paper is supported by a grant from the National Social Science Foundation of China (No: 14BTQ034)

References


