International Research Journal of Computer Science (IRJCS)
Issue 09, Volume 5 (September 2018)
ISSN: 2393-9842
www.irjcs.com

DESIGN PATTERNS IN THE WORKFLOW IMPLEMENTATION OF MARINE RESEARCH GENERAL INFORMATION PLATFORM

Baomin Shao*
Department of Computer Science and Technology,
Shandong University of Technology, China
bmshao@sdut.edu.cn

Qiuling Li
Department of Computer Science and Technology,
Shandong University of Technology, China
18560999025@163.com

Xue Jiang
Department of Computer Science and Technology,
Shandong University of Technology, China
xajx07@163.com

Manuscript History
Number: **IRJCS/RS/Vol.05/Issue09/SPCS10080**
Received: 08, September 2018
Final Correction: 18, September 2018
Final Accepted: 25, September 2018
Published: September 2018
Citation: Baomin, Qiuling & Xue (2018). DESIGN PATTERNS IN THE WORKFLOW IMPLEMENTATION OF MARINE RESEARCH GENERAL INFORMATION PLATFORM. IRJCS:: International Research Journal of Computer Science, Volume V, 453-458.**doi://10.26562/IRJCS.2017.SPCS10080**
Editor: Dr.A.Arul L.S, Chief Editor, IRJCS, AM Publications, India
Copyright: **©2018 This is an open access article distributed under the terms of the Creative Commons Attribution License, Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited**

Abstract— This paper proposes the use of design patterns in a marine research general information platform. The development of the platform refers to a design of complicated system architecture. Creation and execution of the research workflow nodes and designing of visualization library suited for marine users play an important role in the whole software architecture. This paper studies the requirements characteristic in marine research fields and has implemented a series of framework to solve these problems based on object-oriented and design patterns techniques. These frameworks make clear the relationship in all directions between modules and layers of software, which communicate through unified abstract interface and reduce the coupling between modules and layers. The building of these frameworks is importantly significant in advancing the reusability of software and strengthening extensibility and maintainability of the system.

Keywords—Marine Information Integrating; Object-Oriented; Design Patterns; Workflow; Marine Data Visualization

I. INTRODUCTION

With the development of cloud computing services, one tendency in the marine research domain is the sharing of data and methods over the last 20 years[1-4]. The information service system of marine domain requires customizable, flexible and web-based software solutions to build a user-oriented model. One of the challenges to develop such a software is how to manipulate the operation of multiple simulation programs automatically[5,6]. The implementation of this function can reduce the workload to program and execute various codes, and enable scientists to devote more energy to professional research.

© 2014- 18, IRJCS- All Rights Reserved

Generally, marine model operations may require several data pre-processing or assimilation fusion modules, model modules, data reprocessing and visualization modules, and at least one result validation module, these subroutines often have complex interdependencies between their interfaces and intermediate data, and in practice, human tracking processes become impractical to perform calculations when certain requirements (start time, cut-off time, total running time) are met. At present, a series of software systems based on workflow[5,7,8] or general GIS theory[9,10] have been developed to solve the customization and automation of marine information services. However, when this kind of software is published to the Internet, a more difficult situation emerged: the server must handle hundreds or thousands of execution of user requests, each user request may contain a complete data processing, analysis, and visualization process of time and space, thus each execution module must be monitored, so that subsequent processing to increase the system throughput and resource utilization efficiency has to follow the execution immediately. These problems are difficult to be realized by the traditional process control method. Based on the reusable object-oriented design pattern technology, this paper explores an automated execution process model for marine scientific research to address the above requirements, and develops the marine geographic information system network platform based on this model.

II. PLATFORM ARCHITECTURE

The web-based marine research general information platform interacts with users through the website interface, it adopts web form to collect user requirements and stores them in a custom process database, whose design is focused on data, analysis method and visualization method marine research users. The background execution system monitors the submission of the user's new requirement web-form, then takes out the form information and puts it in the execution queue to wait for the idle execution server, one order ID is sent to the execution server to run. When the order is executed, the system sends both the order and result to the user by mail. The system structure and execution process are shown in figure 1.

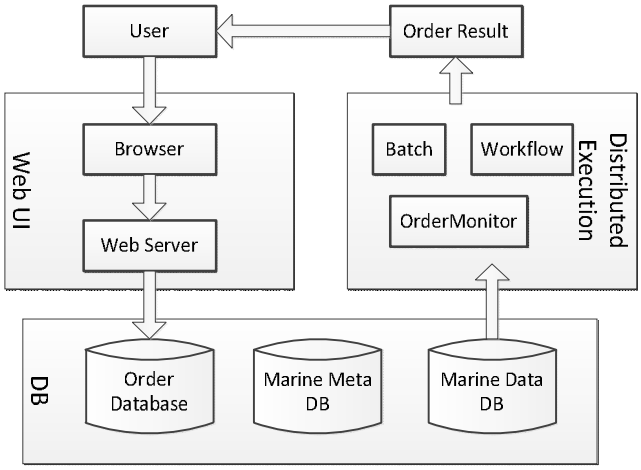

Figure.1 Architecture and Dataflow of Marine Research General Information Platform

I. DESIGN PATTERNS

Design patterns was introduced into computer science by Erich Gamma et al. in the 1990s from the field of architectural design, and its most important idea was the conception of encapsulating changes[11]. Good system design comes from successful development experiences in the past, and the introduction of design pattern enables developers to reuse successful design and architecture more easily. Design pattern can be divided into three types: behavioral, creational, and structural according to the purpose. One difficulty of the application design patterns is identifying applicable patterns in specific application areas, which requires software developers to spend much time in deeply analyzing system requirements and change expectations[12]. This paper aims at designing and developing the general network service platform for marine research, studying and applying design patterns, and discussing how to select and apply appropriate design patterns to improve the scalability and maintainability of the software during designing and developing. It is hard to define the specific scope of software functions due to the frequent change of research user requirements. If the coupling between the software layers is very strong, once the user put forward new requirements or changes of the original version, will likely lead to large-scale software modification, this will not only bring huge waste of manpower and material resources, but also may cause delay of the research.The introduction of design patterns in the system design can greatly improve the scalability of the system and ensure that the system can effectively respond to changes of user requirements. Moreover, it can also enhance the maintainability of the system and reduce the cost of the later maintenance.

© 2014- 18, IRJCS- All Rights Reserved

II. SYSTEM DESIGN

A. Process execution engine and Composite pattern

In the marine research general platform, the user needs the system to run different kinds of working process nodes, such as the data processing, analysis, visualization process. The working process node need have the ability to be executed in parallel, serial, circular mode, so the management node can be regarded as a series of nodes container, each management node adopts its own executive control method for internal nodes and appears as a single task node for process execution. Considering the possible changes of user requirements, when adding, deleting, changing the execution process nodes or adding management nodes, the system structure should not change significantly [13], and the composite pattern is adopted to design process management.

Composite pattern is a simple but profound structural pattern. It defines a class hierarchy of basic objects and composite objects that can be combined into more complex composite objects, which in turn can be combined, and so on. Composite objects can be used anywhere in client code, clients can consistently use composite objects and individual objects. Usually users don't know or care whether they're dealing with a leaf node or a composite component. It is a one-to-one association between the workflow base class instance and the process monitoring service in the system. The monitoring service only cares about the execution of a process, but because there is a list structure in the internal workflow, a scheme is needed to change this association into a one-to-many relationship. Figure 2 shows the basic structure of the Composite pattern in the system. It is a class hierarchy of execution nodes. The base class node has four derived classes: Data Access, Analysis, Visualization and the fourth derivative class, Compositor, a combination that contains a list of multiple node instances. When the Run method of Compositor is called, it delegates the method to every node instance in the list. So an instance of a Compositor is like a single node to the system. You can pass it to any function or object and it behaves like an execution node, even though it's just a proxy for a set of node instances.

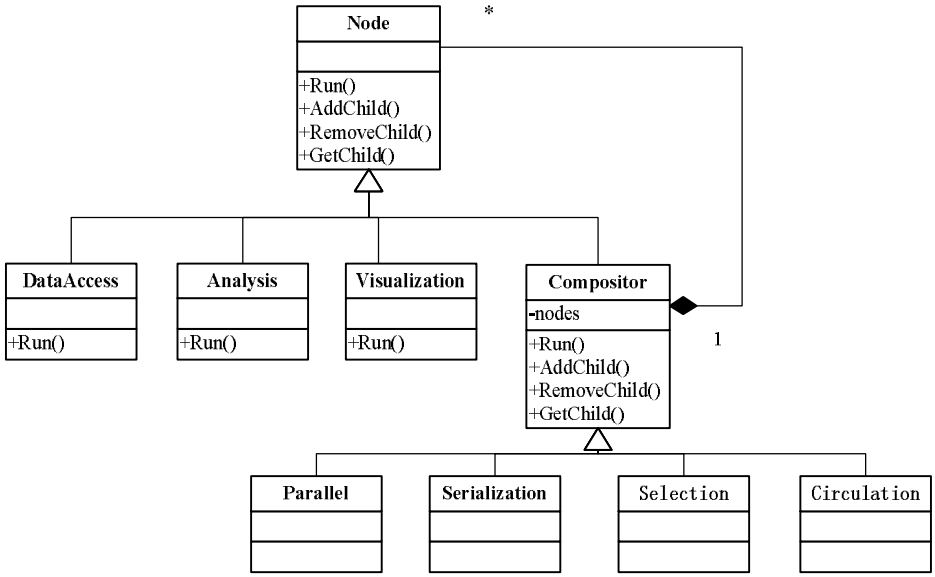

Figure.2 Class Diagram of User-Define Nodes (Composite Pattern)

B. Process Creation and Builder Pattern

No design pattern exists independently in actual system design and development, and there is a relationship between design patterns on the mutual support and dependence. Understanding design patterns and their relationship is useful for solving problems that may arise when a design pattern is applied[14]. There are two process execution methods in the marine research platform: workflow engine and batch execution. Two process representations, the tree data structure defined in the Composite pattern and the linear structure defining the batch execution, which need to be constructed respectively from the process database. The main steps of process construction are the same, including data access, analysis and visualization, but the creation of each sub-step may vary greatly. The builder pattern separates the construction of a complex object from its presentation, allowing the same build process to create different representations, thereby implementing the decoupling of process definition data in the system and the generation of execution process instances, it's primarily used for building a complex object step by step. The builder pattern in the system is shown in figure 3. The base Builder class specifies the abstract interface for each class that creates a service object, while Workflow and Batch implement the Builder's interface to construct and assemble the individual components of the service and provide an interface to retrieve the product.

© 2014- 18, IRJCS- All Rights Reserved

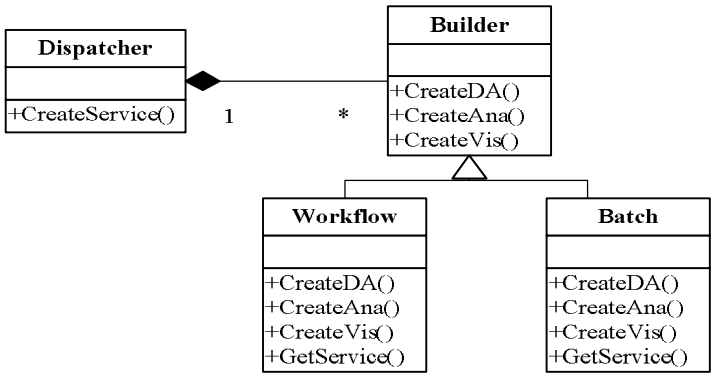

Figure.3 Workflow Creation Class Diagram(Builder Pattern)

The dispatcher notifies the workflow production component, which processes the request from the workflow handler, and adds the component to the product where the user gets the results from.

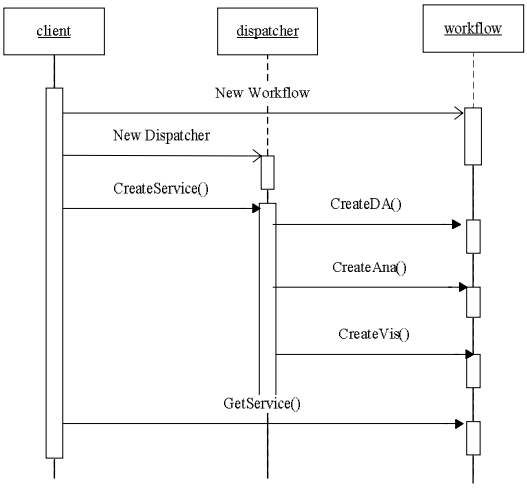

Figure.4 Interaction Diagram of Workflow Nodes Creation

C. Visualization and Decorator Pattern

There are two dimensions of user visualization requirement changing in the system: the extension of thematic diagram and the alteration of the structure of thematic diagram.

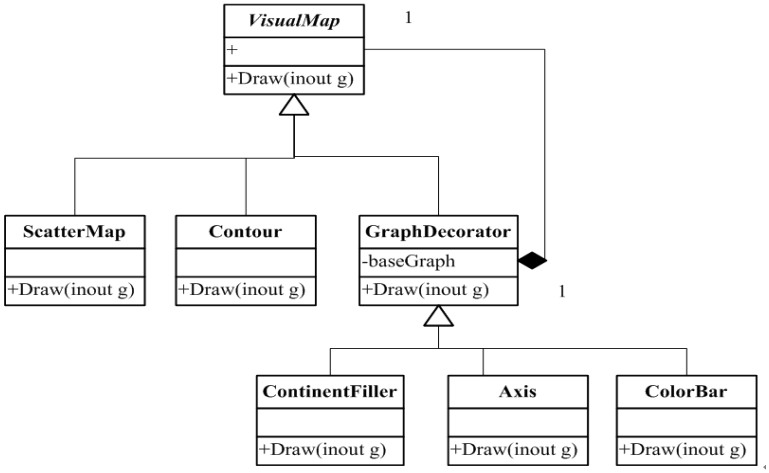

Figure.5 Visualization Library Class Diagram(Decorator Pattern)

Some data may need to be presented as a point plot or a contour map; other types of thematic diagrams may be introduced. On the other hand, each plot itself also has a variety of possible changes, such as whether or not it includes the coordinate system, the projection of the coordinate system, the color bar category, and so on. As the number of subclasses grows explosively in response to changes in both directions, the decorator pattern needs to be applied.

© 2014- 18, IRJCS- All Rights Reserved

The use of the Decorator pattern is to dynamically add functionality to an object to make it look like inheritance, which uses child nodes to process data, and data structures can be dynamically created at run time. The following figure shows the structure of the decorator pattern of the system:

The VisualMap class has three derived classes: ScatterMap, Contour and GraphDecorator. GraphDecorator is the base class of every decorator class, its behavior is expressed as a VisualMap, and since it contains an instance of VisualMap, when its Draw method is called, it will delegate to its baseGraph to execute in addition to its own function implementation, this forms a VisualMap function sequence, including the function of a thematic map and the functions of various renderers. The instance is created as follows:

```
ContourMap cm = new Title(
new ContinentFiller(
new Axis(
new ColorBar(
new Contour(dataInfo, inputDataFileName, rec),
recColorBar, dataInfo),
dataInfo, rec, 10),
rec, dataInfo.xMin, dataInfo.xMax, dataInfo.yMin, dataInfo.yMax), rec, title)
```

When user requirement changes, only the thematic or renderer classes need to be changed separately, thus loose coupling of visual implementation can be achieved, and system scalability can be improved.

III. EXPERIMENT AND CONCLUSION

This experiment submits the EOF analysis form of the ERSST data to the system to verify the system library function. The data type is sea surface temperature dataset ERSST with time range of Jan 1 1951 to Dec 31 1986 and spatial range 120°-250° in longitude, 20°-55° in latitude, the selected analysis method is EOF (empirical orthogonal decomposition), and visualization is required to be applied on the analyzed temporal-spatial results. The data preprocessing is performed by inverse distance interpolation and ordinary Kriging interpolation algorithm. The EOF analysis can obtain the principle components of variables, and effectively understand the main space-time variation of the original data[15]. The execution of form involves establishing user processes, creating analysis and visual execution nodes, combining user requirement functions according to filled form, and actual running of different nodes. The execution system has a dedicated service, OrderMonitor, which manages the user-submitted forms, queries the order table on a set cycle, and maintains a task queue with the form data. When the user submits a new form, the OrderMonitor will insert the form information to the queue and update the information in the database to wait for execution, and select one task from the queue according to the scheduling algorithm and starts the new node execution process. The system components and data processing component objects involved in the user submission form are flexibly created by the builder pattern and composite pattern. The system component objects are created by the builder pattern. Composite pattern is responsible for various analysis and visualization functions submitted by the user, and the visualization is implemented by calling the visual class library designed in the decorator pattern.

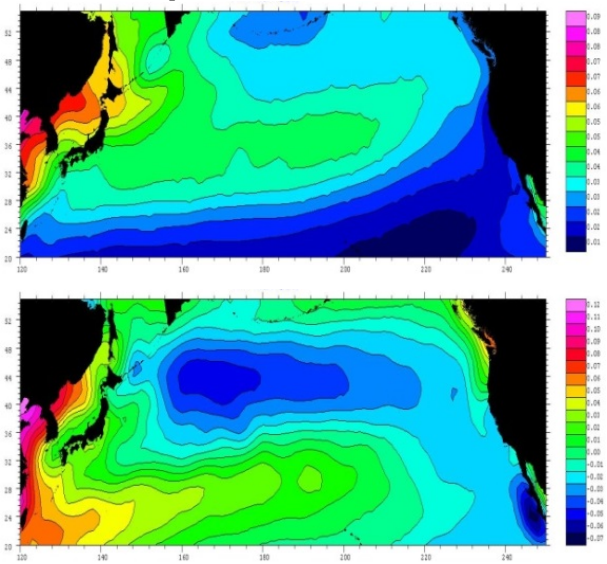

Figure.6 Contour Map of Spatial Function of EOF Analysis Result

© 2014- 18, IRJCS- All Rights Reserved

The results sent to the user in the system contain form information requested by the user, marine data, analysis results, and visual plots, which come from component objects created by design patterns classes. Figure 6 shows the contour map of the spatial function of the first two modes of EOF analysis in the North Pacific Ocean. By this result, the variance contribution rate of the first mode is 97%. The spatial pattern shows that the regional uniformity, the amplitude of variation increases with the increase of latitude, and achieve the maximum in 35 to 45 ° N of the north western Pacific Ocean. This illustrates the Kuroshio annual change has a significant impact in this area. The second mode still shows seasonal variation, the spatial pattern demonstrates a north-south dipole structure in the middle of the north Pacific. The Kuroshio region and its extension body changes a lot, indicating that it is driven by annual cycle of solar radiation and Kuroshio. Design patterns represent excellent reusable design experiences from successful system design. The experiment proves that when the design patterns are introduced into the design of marine research general platform, the developing efficiency of the platform software is effectively improved, and the high reusability of the code and the scalability of the system are ensured.In this paper, the application of composite, builder, decorator patterns in the design of processing node creation, management and visual class library has been studied. Based on this technology, a general platform of marine research has been developed. At present, the software system has been put into application and received high evaluation from users.

REFERENCES

1. M. Kulawiak, A. Prospathopoulos, L. Perivoliotis. Interactive visualization of marine pollution monitoring and forecasting data via a Web-based GIS[J]. Computers & Geosciences, 2010，36：1069–1080.
2. Jacek Dąbrowski, Marcin Kulawiak, Marek Moszyński. Real-time Web-based GIS for Analysis, Visualization and Integration of Marine Environment Data[J]. Information Fusion and Geographic Information Systems, Lecture Notes in Geoinformation and Cartography, 2009, 6：277-288.
3. JiachenYang, ChangWang, QimingZhao. Marine surveying and mapping system based on Cloud Computing and Internet of Things [J]. Future Generation Computer Systems, Volume 85, August 2018, Pages 39-50.
4. Marcin Kulawiak, Andrzej Chybicki, Marek Moszynski. Web-Based GIS as a Tool for Supporting Marine Research[J]. Marine Geodesy, 2010, 33:135–153.
5. Christopher W Harrop, Ligia Bernardet, Mark Govett1. A Workflow Management System for Automating Weather and Climate Simulations[C]. CIRES' Annual, Institute-wide Symposium, 2008.
6. Vittorio Barale. A supporting marine information system for maritime spatial planning: The European Atlas of the Seas. Ocean & Coastal Management，https://doi.org/10.1016/j.ocecoaman.2018.03.026.
7. Stein Sandven, Torill Harare, Robin Berglund. IWICOS: Integrated Weather, Sea Ice and Ocean Service System[J]. Elsevier Oceanography Series, 2003, 69:621-626.
8. Benjamin D. Best, Patrick N. Halpin, Ei Fujioka. Geospatial web services within a scientific workflow: Predicting marine mammal habitats in a dynamic environment[J]. Ecological Informatics, 2007, 2(3): 210-223.
9. ZHAO Qifeng, FAN Hong, LAI Jianfei. Application of GIS to Supporting Atmospheric and Oceanographic Data Management and Visualization[J]. Geo-spatial Information Science, 2009, 12(1):50-55.
10. Joe Breman, Dawn Wright, Patrick N. Halpin. The Inception of the Arcgis Marine Data Model in Marine geography: GIS for the oceans and seas[M]. Redlands, California: ESRI Press, 2002:3–9.
11. Gamma E, Helm R, Johnson R, Vlissides J. Design patterns: elements of reusable object-oriented software[M]. Reading, Massachusetts: Addison-Wesley, 1995.
12. B.C.P. Heng, R.I. Mackie. Using design patterns in object-oriented finite element programming[J]. Computers and Structures, 2008.
13. Taylor I.J. Deelman, E. Gannon, D.B. Shields M. (Eds.): Workflows for e-Science Scientific Workflows for Grids[J]. Springer-Verlag, 2007.
14. Huseyin Ergin, EugeneSyriani, JeffGray. Design pattern oriented development of model transformations [J]. Computer Languages, Systems & Structures . Volume 46, November 2016, Pages 106-139.
15. H. Bjornsson, A. Venegas. A manual for EOF and SVD analyses of climate data[M]. CCGCR Report No. 97-1, Montréal, Québec, 1997(52).

© 2014- 18, IRJCS- All Rights Reserved