Composition Model of Complex Virtual Instrument for Ocean Observing

Keyong Hu, Zhongwen Guo, Zhaodong Che, Feifei Shen
College of Information Science and Engineering, Ocean University of China, Qingdao, China
Email: {hukeyong, guozhw, che, shenfeifei}@ouc.edu.cn

Abstract— **Ocean observing in global and regional scale is an important part of Internet of Things. Traditional virtual instruments have their limitations when applied in ocean observing, resulting in poor data usability and low development efficiency. In this paper, the complex virtual instruments with enhanced capabilities for ocean observing is introduced to realize cost-efficient developments, in which the objects related to ocean observing are managed uniformly, the common functions that can be reused are provided and the configuration that enable code reuse and visual developments are presented. Furthermore, the concept of composition model of complex virtual instrument for ocean observing is proposed, in which complex virtual instruments that act as four roles can be interconnected by use of three operations. With the assistance of the composition model, the observation data of different observing systems can be exchanged and shared in multiple ways and the development efficiency of processing software is improved. The model has been applied to practical developments to validate its feasibility and effectiveness.**

Index Terms— **virtual instrument, complex virtual instrument, composition model, ocean observing**

I. Introduction

THE concept of ”Internet of things” has been proposed for several years. It is envisioned as a huge network that interconnects physical objects by means of a variety of devices with sensing, processing and networking capabilities such as Radio-Frequency IDentification (RFID) tags and sensors. In this vision, obviously, the ocean is a dispensable part that should be incorporated due to its great benefits to the public, including weather and climate prediction, natural hazard mitigation, life and property protection and maritime resource exploitation.

The fundamental approach of connecting oceans with other objects is deploying sensing devices in the sea area monitored, conveying events occurred in ocean in the digital format, modelling ocean phenomena [1] and then making ocean understandable and interpretable, which we call ”Ocean Observing”. To date, there have been many programs initiated for regional, national and global observing, such as Argo for measuring ocean temperature and salinity on a global scale [2], Integrated Ocean Observing System (IOOS) for sustained observing of U.S. coastal waters and great lakes [3]. These observing activities have their own objectives and collaborate to form the global ocean observing system (GOOS), which is considered as a component of the Global Earth Observation System of Systems (GEOSS) similar to the ”Internet of things” in terms of concept [4].

According to [5], an ocean observation is an act which observes a sea area and assigns a result to an ocean phenomenon. The phenomenon is a property of an identifiable sea area, and the result contains the value generated by observing instruments. The deployment of instruments is difficult and expensive due to harsh environment conditions in oceans. Different types of instruments that are equipped with sensors are often fixed in mobile or stationary observing platforms like drifting and moored buoys, ships of opportunity, volunteer observing ships and underwater gliders [6]. Such platforms investigate sea areas periodically or continuously, and transmit the sensed values in wireless or wired communications.

The sensed values need to be displayed and processed to make sea areas monitored understood easily, which is the responsibility of virtual instruments. A virtual instrument (VI) is actually a software program that processes the measurements from a physical instrument (PI) according to specific algorithms and provides friendly graphical interfaces to users [7], [8]. Many previous works focus on design and implementation of VI-based measurement system [9]–[12]. Especially, in [13], a virtual instrument is designed to automate water quality monitoring process, with several internet capabilities such as alarm notifications and remote storage. Most VI-based measurement systems are developed by use of easy-to-use tools, which includes LabVIEW [14], LabWindows/CVI, HP VEE, TestPoint and Measurement Studio [15].

However, traditional VIs have their limitations when applied in ocean observing field. On one hand, only the sampled values of ocean phenomena are displayed while the related metadata that help to improve our ocean knowledge are neglected. For example, the observed object reveals the regions and locations an observation occurs, and the instrument information indicates sampling methods of specific variables and current running states; on the other hand, ocean phenomena values are scalar or vectorial, and multiple sampling geometries result in various data types, including point, time series, profile, trajectory, trajectory-profile and gridded data [16]. The presentation of these data has specialized requirements

Manuscript received August 10, 2011; revised January 2, 2012; accepted April 16, 2012. © 2005 IEEE.

This work was supported in part by Specialized Research Foundation for Ocean Renewable Energy under grant No. GHME2012ZC02 and Specialized Research Foundation for National Public Benefit under grant No. 201105034

doi:10.4304/jsw.9.5.1177-1188

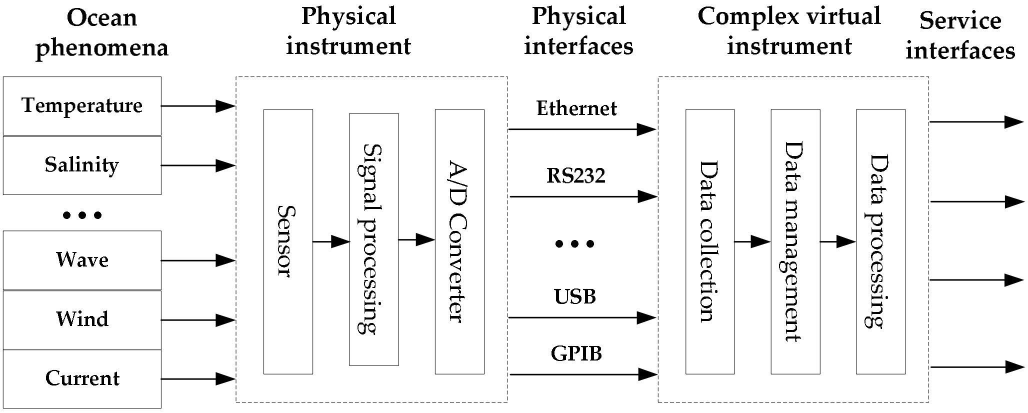


Figure 1. Data flow from physical instruments to complex virtual instruments.

that can't be satisfied in traditional VIs. Many manufacturers provide ocean instruments in accompany with VI software that support data collection, storage and display. However, it is very hard to integrate them into user-oriented observing systems due to lack of Application Programming Interfaces (APIs). The general approach is writing codes to acquire the measurements from instruments and process them according to specific algorithms, which is time-consuming and low cost-efficiency. Facing the aforementioned problems, the need for designing ocean observing oriented complex virtual instrument (CVI) systems that have enhanced functionality with high cost-efficiency arises.

The concept of Complex Virtual Instrument (CVI) is first proposed in [17], in which a standardized CVI-based development model is provided. A CVI is composed of multiple VIs and characterized by enhanced functions and interfaces. As compared with traditional VIs, its virtues are as follows: 1) CVI realizes unified data management of various objects in ocean observing process, mainly including phenomena, instruments and results; 2) CVI is capable of connecting multiple types of instruments through configuring instrument drivers; 3) CVI can provide access to its data via uniform networking interface; 4) CVI codes can be reused and loaded from remote CVI library to reduce redundant developments. In this paper, we provide specialized design for Ocean Complex Virtual Instrument (OCVI) system that meets the requirements in ocean observing field, and then propose the concept of complex virtual instrument network, which integrates all the observing instruments distributed at oceans spatially into a whole.

The structure of this paper is as follows: Section II presents the structure design of the OCVI for ocean observing. In Section III, OCVI implementation strategy is described. In section IV, OCVI composition model is proposed for the purpose of interconnecting OCVIs. Section V validates the efficiency and effectiveness of OCVI and its composition model through practical developments. Section VI concludes this paper.

II. OCVI Design

Fig.1 illustrates ocean observing process from PIs to OCVIs. PIs sense ocean phenomena as the input, convert signals and output digital data via communication interfaces. Similarly, the OCVI receives the digital data as the input, processes them and provide them to other OCVIs via service interfaces. Hence an OCVI C can be represented formally as:

$$C = \langle D, F, I \rangle$$

Where
D = data set produced in ocean observing process D
F = function set based on specific algorithms F
I = service interfaces I

As shown in Fig. 2, the OCVI is composed of data management module, multiple VIs and supportive modules. It is characterized by service interfaces. The data management module is responsible for retrieval, storage and cache of historical and real-time observations. VIs obtain observation data through data management module and process them according to user operations. Supportive modules provide enhanced functions to ease user operations.

A. Data management

This module realizes unified management of observation data which are actually descriptive information of various objects in ocean observing process. The relationship of these objects is shown in Fig. 3. Responsible object that sets up observing projects deploy method object that indicates how an observation is conducted in observed object that represents where an observation occurs, sense sensed object that denotes what phenomena are sampled and produce result object that records when observed values are generated.

All the objects except result object follow the storage structure (ItemTitle, ItemContent). An ItemTitle describes an attribute name of an object and an ItemContent assigns value to an attribute. The advantage of this approach is the object descriptions can extend when necessary, whereas the disadvantage is that the relationships between

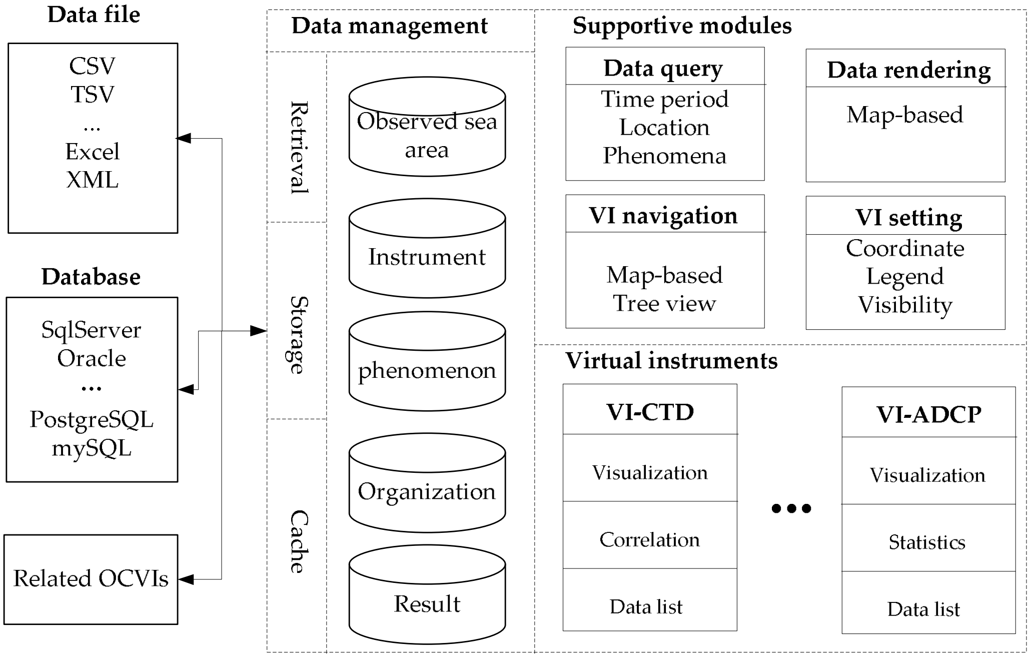


Figure 2. OCVI structure.

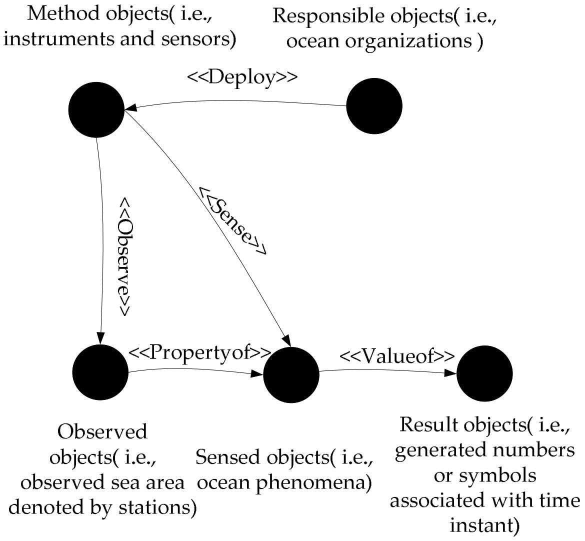

Method objects(i.e., instruments and sensors)
Responsible objects(i.e., ocean organizations)
<<Deploy>>
<<Sense>>
<<Observe>>
<<Propertyof>>
<<Valueof>>
Observed objects(i.e., observed sea area denoted by stations)
Sensed objects(i.e., ocean phenomena)
Result objects(i.e., generated numbers or symbols associated with time instant)

Figure 3. The relationship of objects in ocean observing process.

objects and some key information may be missing. For example, the ”OrganizationID” attribute in the Platform information indicates which organization is responsible for deployment and maintenance; longitude and latitude attributes in the Station information are indispensable for data processing based on geographical information. To this end, the requisite and reference attributes are built in respective object description.

The data storage of ocean phenomena is relatively complex. First, one phenomena may correspond to one or multiple quantities. For example, the salinity produces one ”salinity” quantity while the current produces the quantities including ”horizontal current velocity”, ”horizontal current direction”, ”vertical current velocity”, etc. Second, some quantities are calculated based on other quantities. For instance, continuous-wind speed is the mean wind speed from the primary anemometer during a respective 10-minute period. Herein, the concept of ”Variable” is introduced to represent a quantity identified by triple (latitude, longitude, depth) at a time instant, which enables simple storage and general processing. The observed phenomena of an OCVI are separated into a collection of variables, each of which stores result data as uniform structure (VariableID, DateTime, Value, Quality-ControlFlag). Various sampling geometries are expressed through grouping variables. For example, a temperature profile comprises temperature variables that are sensed in different depths vertically.

Users pay more attention to station and variable objects since they can reflect ocean environment state of observed sea areas directly. As illustrated in Fig. 4, the variable object can be used to represent the data set since it relates to all the other objects. Let $N_{S,i}$ and $N_{V,i}$ denote the number of stations and variables in an OCVI C_i and Station $S_{i,j}$ has $N_{S_j,i}$ variables observed. The data D_i of OCVI C_i can be identified by its contained variables that relate to the other objects by giving consideration to stations:

$$D_i = \left\{ V_{jk} | j = 1, 2, ..., N_{S,i}; k = 1, 2, ..., N_{S_j,i} \right\}$$

Where

$$\sum_{j=1}^{N_{S,i}} N_{S_j,i} = N_{V,i}$$

B. Virtual instruments

The VIs receive result data of variables and provide them to users in a visual way. A VI is responsible for

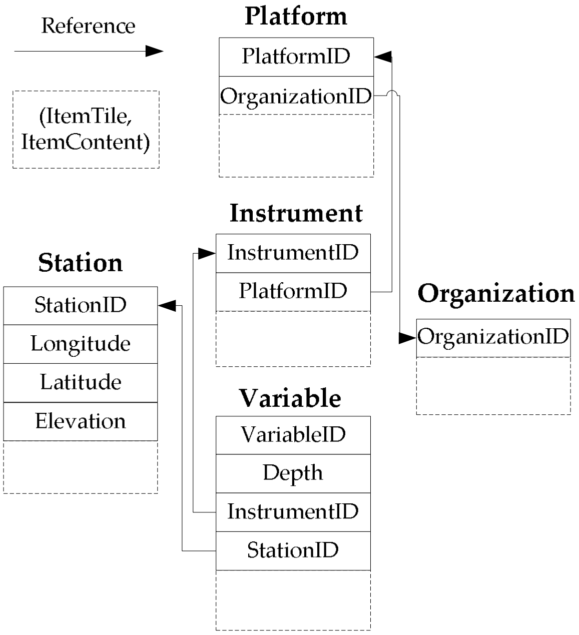


Figure 4. The references between objects in ocean observing process.

processing certain phenomena of an observed object that are sensed by one or multiple PIs of the same type. The types of variables of a VI are aware since the PIs sense particular phenomena.

Let $F_i = \{f_{i,j} | j = 1, 2, ...N_{F_i}\}$ denote all the available functions of OCVI C_i, in which N_{F_i} means the number of functions. Let $VIS_i = \{VI_{i,j} | j = 1, 2, ..., N_{VI,i}\}$ denote the VI set of OCVI C_i, in which $N_{VI,i}$ means the number of VIs.

Each VI comprises a set of function modules that meet specialized processing requirements. Hence the function set F_i is split into two or more subsets $F_{i,1}, , F_{i,N_{VI,i}}$ that corresponds to each VI:

$$\bigcup_{j=1}^{N_{VI,i}} F_{i,j} = F_i$$

Partial functions are commonly used in different VIs, thus two function subsets are overlapping:

$$\forall F_{i,j}, F_{i,k} \in \{F_{i,1}, ...F_{i,N_{VI,i}}\}, j \neq k \Rightarrow F_{i,j} \bigcap F_{i,k} \neq \emptyset$$

Similarly, the data D_i is split into two or more subsets $D_{i,1}, , D_{i,N_{VI,i}}$ that corresponds to each VI:

$$\bigcup_{j=1}^{N_{VI,i}} D_{i,j} = D_i$$

A data subset contains multiple variables, each of which is processed by a unique VI. Thus two data subset are non-overlapping:

$$\forall D_{i,j}, D_{i,k} \in \{D_{i,1}, ...D_{i,N_{VI,i}}\}, j \neq k \Rightarrow \\ D_{i,j} \bigcap D_{i,k} = \emptyset$$

Hence the VI set VIS can be defined as:

$$\begin{aligned} VIS_i &= \{VI_{i,j} | j = 1, ..., N_{VI,i}\} \\ &= \{F_{i,j}(D_{i,j}) | j = 1, ..., N_{VI,i}\} \\ &= \left\{f_{i,k}(D_{i,j}) | j = 1, ..., N_{VI,i}; k = 1, ..., N_{F_{i,j}}\right\} \end{aligned}$$

Where $N_{VI,i}$ represents the number of VIs in an OCVI and $N_{F_{i,j}}$ represents the number of functions in function subset $F_{i,j}$.

The function set are mainly implemented by the following modules:

- The time curve module: The module is responsible for displaying the trend of variable data over a period of time. Each variable is modeled as a time evolving curve.
- The vertical profile module: This module is used to model the vertical trend of profile variables. A curve is plotted according to the values of variables at different depths.
- The variable correlation module: For displaying the variation relationship between two variables, this module plots the points each of which is located with the values of two variables at one timestamp.
- The vector statistics module: It is responsible for displaying statistics results of vector data over a period of time. The statistics is to analyze the distribution situation of direction and speed in multiple directions. In each direction, the frequency and average speed are presented.
- The vector field module: This module is responsible for visualizing vectorial variables in one plane that is divided into multiple grids. In each grid, the direction and speed of variables are displayed.
- The scalar field module: Similar to vector field module, it is responsible for visualizing gridded data for scalar variables. The usual method is plotting the isolines according to the variable values.
- The tabular display module: A human-readable numerical data table is provided to users in this module. It runs in static or dynamic mode, between which users can switch manually. In dynamic mode, realtime can be appended in current page timely; in static mode, data records are exhibited in multiple pages.
- The data export module: Users are able to download the data of interest. This module supports exporting data in common file formats.

Supportive modules provide global view to coordinate and ease VI operations, which are as follows:

- The VI navigation module: The module is used for quickly finding the desired VI. The VI can be located based on geographical map, or be browsed in the tree view.
- The VI setting module: It is responsible for controlling common parts of multiple VIs, which mainly include the upper limit, lower limit of coordinates, the legend and the visibility of variable curves.
- The data query module: This module realizes uniform query based on conditions like time period, location and variables. The query result contains a list of VIs that satisfy the query conditions.
- The data rendering module: Because the observed data are geographic related, it is intuitive to load the visualized results of VIs on a 2D/3D map. Especially, loading the animations that display the variation of

variables to the map make the observations understood easily.

C. Service interfaces

The OCVIs communicate with each other through interfaces that are designed based on SOA (Service-Oriented Architecture) [18]. Each interface is implemented as an operation with request and response parameters. The interface set of OCVI C_i can be defined as:

$$I_i = \{I_{i,j} | j = 1, ..., N_{I_i}\}$$

Where N_{I_i} means the number of interfaces of OCVI C_i

The interfaces are divided into query and access interfaces for respective purpose. The query interfaces are used for searching and retrieving observations of interest from OCVIs. The search operation receives the search conditions including time, location and variable, looks up the stations that satisfy the search conditions, and returns them with brief description of associated objects. Distributed search can be realized by use of the query interfaces. The values of search conditions are assigned at the search client and forwarded to OCVIs that run over different internet sites. The data from distributed OCVIs mainly include identifiers and names of various objects that can be used in the following requests. Currently, we have designed the GetOrganizationById, GetInstrumentById, GetVariableById and GetResultByVariable operations for returning the data of responsible objects, method objects, sensed objects and result objects according to their object identifiers respectively.

The access interface is used for providing observations in the form of object. Each operation corresponds to each type of object. Hence the access interface includes GetStations, GetVariables, GetInstruments, GetOrganizations and GetResults operations. Leveraging the access interface, the observations from distributed OCVIs can be integrated together. The relationships of all the object data need to be remapped since identifications of various objects after integration have changed.

The difference between two interfaces is that the operations filter the data of corresponding objects according to the request parameters in the query interface, whereas the operations of the access interface return all the data of corresponding objects. They are used for different purposes, but both provide standard retrieval of observations. Additional interfaces can be contained to satisfy new communication requirements.

III. OCVI Implementation

OCVIs need to be implemented before they are applied to practical systems. The OCVI implementation involves code development, service management and configuration process. Next we describe them sequentially.

A. Code development

Following the OCVI structure, the functions of data management module and supportive modules are relatively fixed. They are rarely modified since the description model of objects stay unchanged. On the contrary, the VIs need to be extended due to dynamic user requirements. Especially, the rapid development of ocean observing technology leads to many new type of instruments. It is necessary to develop new VIs to process the results from these instruments. Therefore, component-based technology is adopted in the VI developments. Each VI is capsulated as a component that can be easily loaded into an OCVI and unloaded from OCVI with little impact.

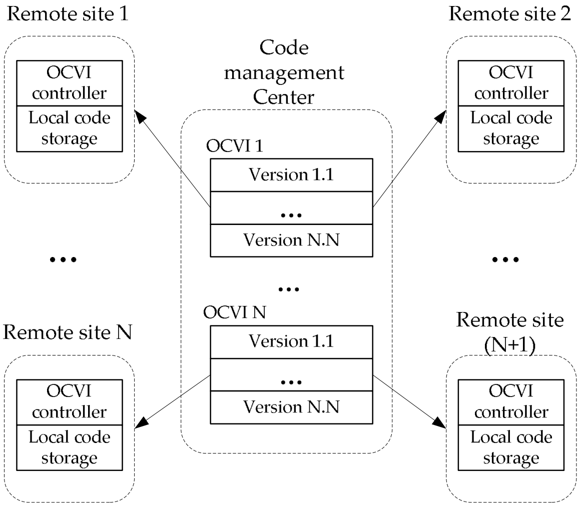


Figure 5. Code management architecture.

We have designed VIs for PIs that are commonly used in ocean observing field. The OCVIs can be built by selecting proper VIs from VI library. The number and type of VIs in an OCVI determines whether the OCVIs are differentiated. The differences between similar OCVIs are identified with version number.

The OCVI codes are managed in the centralized way. The tool that is called OCVI controller is provided to the designer that is familiar with ocean observing but not software expert. The designer can browse all the available OCVIs and select proper ones from them. The configuration is required before the OCVIs are applied. The OCVI codes are loaded from code management center at the first run time and then stored locally. The OCVI controller is responsible for initiating each module according to the configuration. The centralized code management brings two advantages: on one hand, the codes are easy to update. The client is notified when there are updated versions for OCVIs in use; on the other hand, the designer can compare OCVIs according to their descriptions and select the one that is most appropriate for practical observing system.

B. Service management

The service interfaces are designed to facilitate OCVI communications and compositions. The OCVI services have the same interface definitions. They are deployed along with the OCVI deployment. Here web service architecture [16] is referenced, which has similarities and also differences with OCVI service management. Similarly,

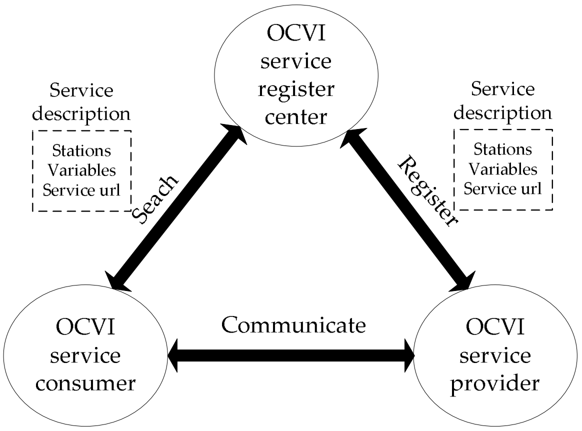


Figure 6. OCVI service management architecture.

the publishing and search process are all based on the Universal Description, Discovery, and Integration (UDDI) standard. As illustrated in Fig. 5, OCVI service providers publishes its service instance to OCVI service registry center with service description; OCVI service consumer finds the desired services in the registry center, invokes them directly and obtains the required data.

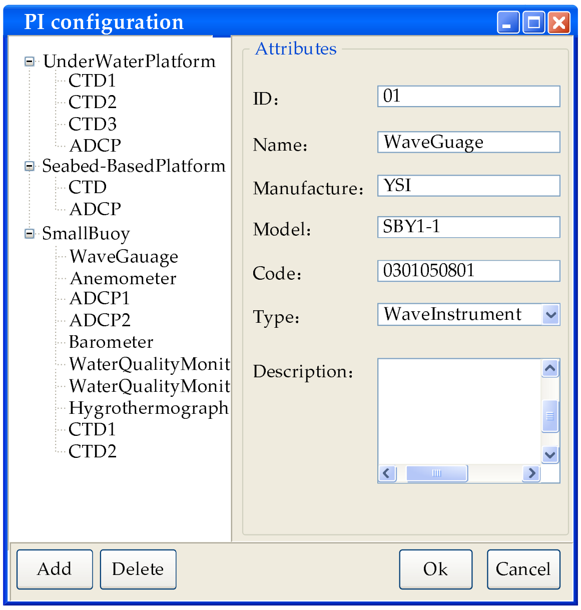
The format of OCVI service descriptions is different from traditional web service architecture. The service definitions are not involved in the publishing and search process since the interface names and parameters are defined in advance for all the OCVIs. The service descriptions mainly include observed objects and sensed objects associated with time period. The OCVI providers submit their own descriptions to the registry center, and the OCVI consumer can be aware of what phenomena in which sea area are observed in an OCVI. Additionally, users are not required to participate in the register process. The codes for registering are built in the OCVI controller and are executed automatically at the first run time.

C. OCVI configuration

Observed objects are specified in this step. The configured information includes locations (i.e. longitude and latitude), name and explanatory descriptions. The properties of the observed objects are specified in the next step. This step can be conducted in the form of tree view and map that seems to be visual and intuitive.

The descriptive information of objects vary with different OCVIs. In an OCVI, the result object data come from PIs directly, whereas other object descriptions are obtained through the configuration process. The configuration process is completed through visual dialogue boxes, and the configuration content is in accordance with actual observing environment. The designer is relieved from complex programming and OCVI code reusability is improved. The configuration mainly falls into three steps, which are configurations for stations, PIs and VIs respectively.

1) Configuration for stations: Observed objects are specified in this step. The configured information includes locations (i.e. longitude and latitude), name and explanatory descriptions.
2) Configuration for PIs: This step allows users to specify observing platforms, instruments and variables. An observing platform comprises different types of instruments, and each instrument produces variables that come from sensed phenomena. Hence the configuration is hierarchical. First, the observing platforms are specified with name, code, type; then the instruments are added in a selected observing platform with name, code, type, manufacture, model and frequency; at last, users add the variables to describe the sensed phenomena. The configuration interface is illustrated in Fig. 7.
3) Configuration for VIs: The VI configuration is



- UnderWaterPlatform
  - CTD1
  - CTD2
  - CTD3
  - ADCP
- Seabed-BasedPlatform
  - CTD
  - ADCP
- SmallBuoy
  - WaveGauage
  - Anemometer
  - ADCP1
  - ADCP2
  - Barometer
  - WaterQualityMonit
  - WaterQualityMonit
  - Hygrothermograph
  - CTD1
  - CTD2

ID: 01

Name: WaveGuage

Manufacture: YSI

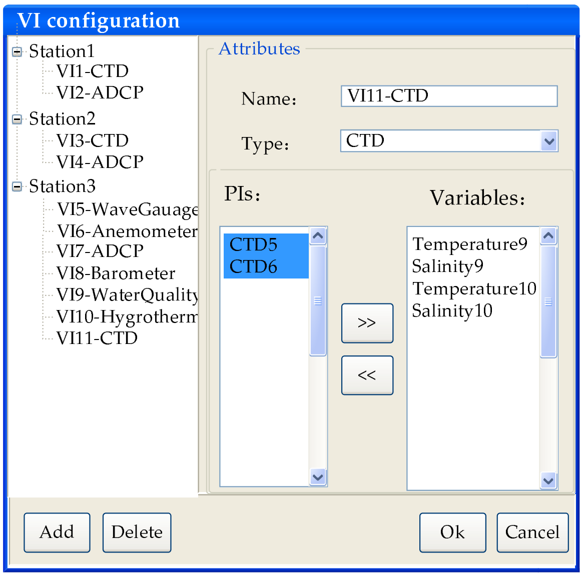
Model: SBY1-1

Code: 0301050801

Type: WaveInstrument

Description:

Figure 7. PI configuration interface.



- Station1
  - VI1-CTD
  - VI2-ADCP
- Station2
  - VI3-CTD
  - VI4-ADCP
- Station3
  - VI5-WaveGauage
  - VI6-Anemometer
  - VI7-ADCP
  - VI8-Barometer
  - VI9-WaterQuality
  - VI10-Hygrotherm
  - VI11-CTD

Name: VI11-CTD

Type: CTD

PIs: CTD5, CTD6

Variables: Temperature9, Salinity9, Temperature10, Salinity10

Figure 8. VI configuration interface.

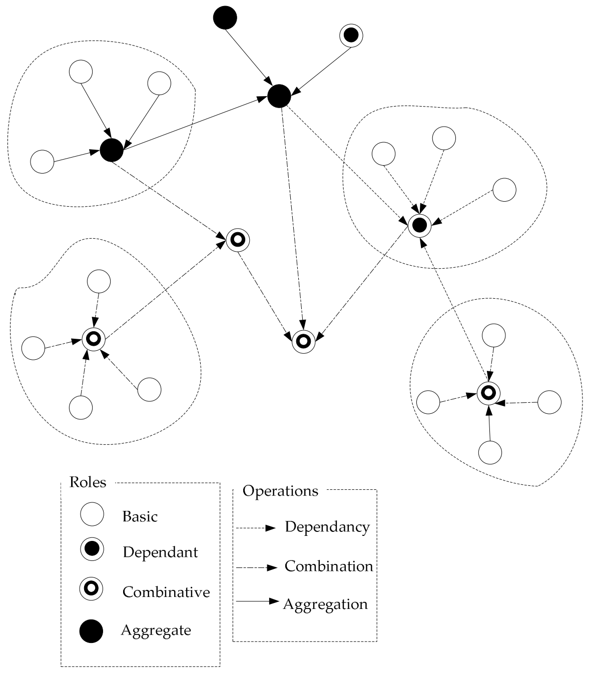


Figure 9. OCVI composition model.

conducted according to observed objects. A VI corresponds to an observed object whereas an observed object corresponds to multiple VIs. The VIs are added in the station nodes, and each VI can process variables from multiple PIs of the same type. The configuration interface is shown in Fig. 8.

IV. OCVI Composition Model

Currently there have been many ocean observing systems initiated in global, national and regional scales. The goal of GOOS is interconnecting existing systems to form a whole system that covers hydrology, meteorology, physical, chemical and other observing for all sea areas. In this situation, more and more observing platforms are deployed at different stations and form global ocean observing network. The observations produced by these observing platforms need to be managed and processed with the corresponding OCVIs. In this section, we introduce the concept of OCVI composition model, which is dedicated to interconnect OCVIs through their service interfaces. Its goal is sharing, exchanging and integrating the observations that react with each other to study the states of ocean environment in different scales.

The OCVI composition model is presented in Fig. 9, in which the OCVI interconnections are based on a hierarchical structure. The OCVIs act as four roles that are basic, dependant, combinative and aggregative respectively. The basic OCVIs interact with PIs directly and process the collected measurements locally. The combinative OCVIs are composed of multiple OCVIs and they can further be assembled into high-level combinative OCVIs. The aggregative OCVIs are merged with multiple OCVIs, and they can further be assembled into high-level aggregative OCVIs. The dependant OCVIs obtain the data from other related OCVIs and use them in the computation process.

Next we discuss three operations in great detail. In the following discussion, the symbols D_{old}, D_{new}, VIS_{old}, VIS_{new}, I_{old} and I_{new} are used to denote data set, VI set and interface set of the destination OCVI before and after the operation.

A. Dependency

The dependency operation reveals use relationship between OCVIs. The dependant OCVIs obtains the observations as the input of its internal computation. The obtained data are not displayed to users (i.e., users are not aware of the obtainment process). These data are often part of the data of source OCVIs that may be basic, dependant, combinative and aggregative. The available source OCVIs can be found in the registry center through the conditional search. The dependant OCVI requests the desired data by specifying stations, variables and time periods in the query interfaces. The source OCVIs act as data providers and return the retrieved data to the dependant OCVI. A typical example of the dependant OCVI is ocean numerical modeling system that is used for forecasting particular phenomena of sea areas based on existing observation data.

Let $\{C_1, ..., C_n\}$ be the source OCVI set of the dependant OCVI C_D. We assume the dependant OCVI C_D obtains the data set $\{d_1, ..., d_n\}$ from source OCVI set. After the dependency operation, the data of the dependant OCVI Dnew is given as follows:

$$D_{new} = \left\{V_{jk} | j = 1, ..., N_{S,new}; k = 1, ..., N_{s_j,new}\right\}$$

For each source OCVI C_i, there exist d_i which are subset of D_i:

$$\forall C_i \in \{C_1, ...C_n\}\ C_i = \langle D_i, F_i, I_i \rangle$$
$$\exists d_i = \left\{V_{i,jk} | i = 1, ..., n; j = 1, ..., P_{S,i}; k = 1, ..., P_{S_j,i}\right\}$$
$$P_{S,i} < N_{s,i} and P_{S_j,i} < N_{S_j,i} \Rightarrow d_i \subset D_i$$

Where

$P_{S,i}$ and $P_{S_j,i}$ means the number of stations and variables of specific station S_j in data set d_i.

In the dependency operation, the data set of C_D is composed of the original data set and the data set from source OCVIs:

$$\begin{aligned} D_{new} &= D_{old} \cup \{d_i | i = 1, ...n\} \\ &= \left\{V_{old,jk} | j = 1, ..., N_{s,old}; k = 1, ..., N_{s_j,old}\right\} \\ &\quad \cup \{d_i | i = 1, ...n\} \end{aligned}$$

The stations in D_{new} is the sum of stations in D_{old} and stations in $\{d_1, ..., d_n\}$:

$$N_{S,new} = N_{S,old} + \sum_{i=1}^{n} P_{S,i}$$

The variables in stations stay unchanged after the dependency operation. When $1 \leq j \leq N_{S,old}$, the variables can be represented by $N_{S_j,old}$:

$$N_{s_j,new} = N_{S_j,old}, \quad 1 \leq j \leq N_{s,old}$$

when $N_{S,old} \leq j \leq N_{S,old} + \sum_{i=1}^{n} P_{S,i}$, we first find which d_m the variables come from. The parameter m satisfy the equation $N_{S,old} + \sum_{i=1}^{m-1} P_{S,i} < j \leq N_{S,old} + \sum_{i=1}^{m} P_{S,i}$, and the station identifier in d_m is calculated as $j - N_{S,old} - \sum_{i=1}^{m-1} P_{S,i}$:

$$\begin{aligned} N_{s_j,new} = P_{S_{\left(j-N_{S,old}-\sum\limits_{i=1}^{m-1} P_{S,i}\right)},m}, N_{S,old} \leq j \leq N_{S,old} \\ + \sum_{i=1}^{n} P_{S,i} \end{aligned}$$

There are no new OCVIs generated and no interface bindings involved in the dependency operation. The obtained data are processed in the background according to specialized algorithms. The source OCVI set of the dependant OCVI is not static. They can be adjusted according to the requirements. For example, disabled OCVIs can be removed and new OCVIs that are available can be contained.

B. Aggregation

The aggregation operation merges multiple OCVIs to the aggregative OCVI. The aggregated OCVIs stay unchanged after the operation while the aggregative OCVI manages and processes the data from source OCVIs in a unified way. Merging the source OCVIs to the aggregative OCVI is actually a problem of data integration. It is relatively simple since the formats of various objects are consistent. Compared to the dependency operation, the aggregation operation merges all the object data of each source OCVI into one data repository.

Let $\{C_1, ..., C_n\}$ be the source OCVI set of the aggregative OCVI C_A. The data of C_A contains the data from all the source OCVIs, which can be given as follows:

$$\begin{aligned} D_{new} &= \bigcup_{i=1}^{n} D_i \\ &= \bigcup_{i=1}^{n} \left\{ V_{jk} | j = 1, 2, ..., N_{S,i}; k = 1, 2, ..., N_{S_j,i} \right\} \end{aligned}$$

The stations in D_{new} is the sum of stations in $\{C_1, ..., C_n\}$: $\sum_{i=1}^{n} N_{S,i}$. When the station identifier is in the range [1, $\sum_{i=1}^{n} N_{S,i}$], the corresponding OCVI identifier m satisfy the equation $\sum_{i=1}^{m-1} N_{S,i} < j \leq \sum_{i=1}^{m} N_{S,i}$. Hence the variable in station j of D_{new} is actually that in station $(j - \sum_{i=1}^{m-1} N_{S,i})$ of D_m.

After the data integration is completed, the VIs that are responsible for processing them also need to be integrated. The VI set in the aggregative OCVI can be represented as:

$$\begin{aligned} VIS_{new} &= \bigcup_{i=1}^{n} VIS_i \\ &= \bigcup_{i=1}^{n} \{F_{i,j}(D_{i,j}) | j = 1, ..., N_{VI,i}\} \\ &= \{F_{new,j}(D_{new,j}) | j = 1, ..., N_{VI,new}\} \end{aligned}$$

The VIs of the same type can be merged together. Two VIs $VI_{new,j}$ and $VI_{new,k}$ are of the same type if their function set $F_{new,j}$ and $F_{new,k}$ are the same. The merging process is implemented through combining the data processed by the functions.

$$\begin{aligned} VI_{new,j} \bigcup VI_{new,k} &= F_{new,j}(D_{new,j}) \bigcup F_{new,k}(D_{new,k}) \\ &= F_{new,j}(D_{new,j} \bigcup D_{new,k}) \end{aligned}$$

The aggregation operation is complex compared to the dependency operation. The aggregative OCVIs can be further aggregated into higher-level ones, or composited into the composite ones. The observations of the aggregated OCVIs are related. It is meaningless to aggregate the OCVIs that have no relationships between each other. A common example is that several OCVI systems are aggregated since they observe similar phenomena of adjacent sea areas that react to each other.

C. Combination

The OCVIs can be connected together to provide a comprehensive view to users by use of the combination operation. The generated OCVI can be regarded as the proxy of the OCVIs that are combined. This operation is essentially the composition of web services of the source OCVIs. Its advantage is that the source OCVIs need no modifications and the observation data are still maintained in their respective database. Users operate on the combinative OCVI, which forwards the requests to source OCVIs through service interfaces. The source OCVIs return the data that satisfy the request conditions. The data from multiple source OCVIs are combined and then provided to users.

The combinative OCVI and the basic OCVIs are similar from the point view of users. The difference is that users of the combinative OCVI concentrate on several ocean variables in large scale, whereas users of the basic OCVIs pay more attention to the variables of wide range in small scale.

The OCVI obtains the data of all the objects except result object from source OCVIs at startup time. The volume of these description data is relatively small so that the performance of the composite OCVI is not reduced. At run time, users have been aware of observing state (i.e., what phenomena are sensed at which stations) through the obtained metadata. The result data of interest can be viewed according to users' operations. The data are obtained by forwarding the request to particular source OCVIs. These functions are realized through binding service interfaces of source OCVIs to interfaces of the combinative OCVI.

We consider the combinative OCVI C_c is constructed upon source OCVI set $\{C_1, ..., C_n\}$, and they all have uniform interface definitions.

The new service interface set can be expressed as:

$$\begin{aligned} I_{new} &= \bigoplus_{i=1}^{n} I_i \\ &= \bigoplus_{i=1}^{n} \{I_{i,j} | j = 1, ..., N_{I_i}\} \\ &= \{I_{new,j} | j = 1, ..., N_{I_{new}}\} \end{aligned}$$

Where $N_{I_i} = N_{I_{new}}$

ID	Name	StationID	InstrumentID	Depth	Unit	Range
1	Temperature1	1	1	0	℃	0-30
2	Illumination1	1	2	0	mcd	0-1500
3	Temperature2	2	3	0	℃	0-30
4	Illumination2	2	4	0	mcd	0-1500
5	Temperature3	3	5	0	℃	0-30
6	Illumination3	3	6	0	mcd	0-1500
7	Temperature4	4	7	0	℃	0-30
8	Illumination4	4	8	0	mcd	0-1500

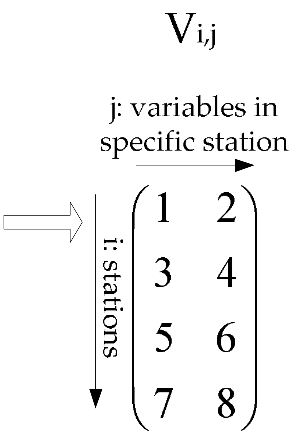


Figure 10. The partial variables in OceanSense OCVI and their representative matrix.

The binding of access interfaces is different from that of query interfaces. For access interfaces, an interface in C_c binds the corresponding interfaces of all the source OCVIs:

$$I_{new,j} = \overset{n}{\underset{i=1}{\oplus}} I_{i,j}$$

For query interfaces, the bindings are conditional. An interface is bound to the corresponding interfaces of several OCVIs $\{C_{n_1}, ..., C_{n_2}\}$ that are the subset of $\{C_1, ..., C_n\}$:

$$I_{new,j} = \{I_{n_1,j} \oplus ... \oplus I_{n_2,j} | 1 \leq n_1 \leq n_2 \leq n\}$$

The combinative OCVI obtains the observations from source OCVIs. The corresponding VIs for processing these observations need to be combined in the way similar with that of the aggregation operation. The combinative OCVI do not store the obtained data persistent. Hence the following hold true:

$$D_{old} = D_{new} = \emptyset$$

In theory, the combinative OCVIs can be constructed iteratively into higher-level ones. However, the iterative combination is not suitable for practical systems since the performance is reduced greatly. On one hand, the requests have to be forwarded many times before they reach the observations; on the other hand, the observation data from source OCVIs will be encapsulated repeatedly in the format of eXtensible Markup Language (XML), which prolongs the response time. Hence the source OCVIs are advised to be basic, dependant and aggregative in the combination operation, which both integrate the observations and ensure the composition performance.

V. Application Example

We have applied the OCVI method to practical developments of different ocean observing systems, including OceanSense, ocean sensor observing system and numerical modeling system. The developments validate the effectiveness and efficiency of the OCVI. Furthermore, three OCVI systems are interconnected via OCVI composition operations.

OceanSense (http://osn.ouc.edu.cn) deploys TelosB motes off the seashore to monitor the illumination and temperature phenomena at the sea area of approximate 300m*100m. Twenty sensor nodes are deployed, and each sensor node is fixed on a float that moves over the sea surface. Three of them are used for forwarding the data from the others to the base station in multi-hop way. The OCVI installed in the regional server 1 receives the data from the base station and provides them to users in a visual way. The representation of variables of OCVI OceanSense is shown in Fig. 10, in which only eight variables are listed due to the paper length.

Ocean sensor observing system is designed to monitor the meteorological, hydrological and chemical phenomena of the sea area. The deployment of the observing platforms goes through two stages. At the first stage, the observing platforms include big buoy, wave rider and shore station. Big buoy and wave rider sense the phenomena such as wave and temperature and transmit them in the wireless way. For shore station, human observers mainly sample the meteorological phenomena, record the observations and store them in files. All the result data are stored in regional server 2 and processed by the installed OCVI software. At the second stage, the observing platforms in use include small buoy, seabed-base and underwater experiment platform. Seabed-base and underwater experiment platform work at the sea floor, sense the hydrological phenomena and transmit them through submarine optical fiber. They are recycled every half a year approximately. These platforms are a little apart from that of the first stage and their produced data are stored in regional server 3. Since the observations from regional server 2 and 3 are related and react to each other, they are aggregated in regional data center for uniform management. The observations are synchronous between regional servers and data center. The number of variables of specific stations is less than 64 and 70 for

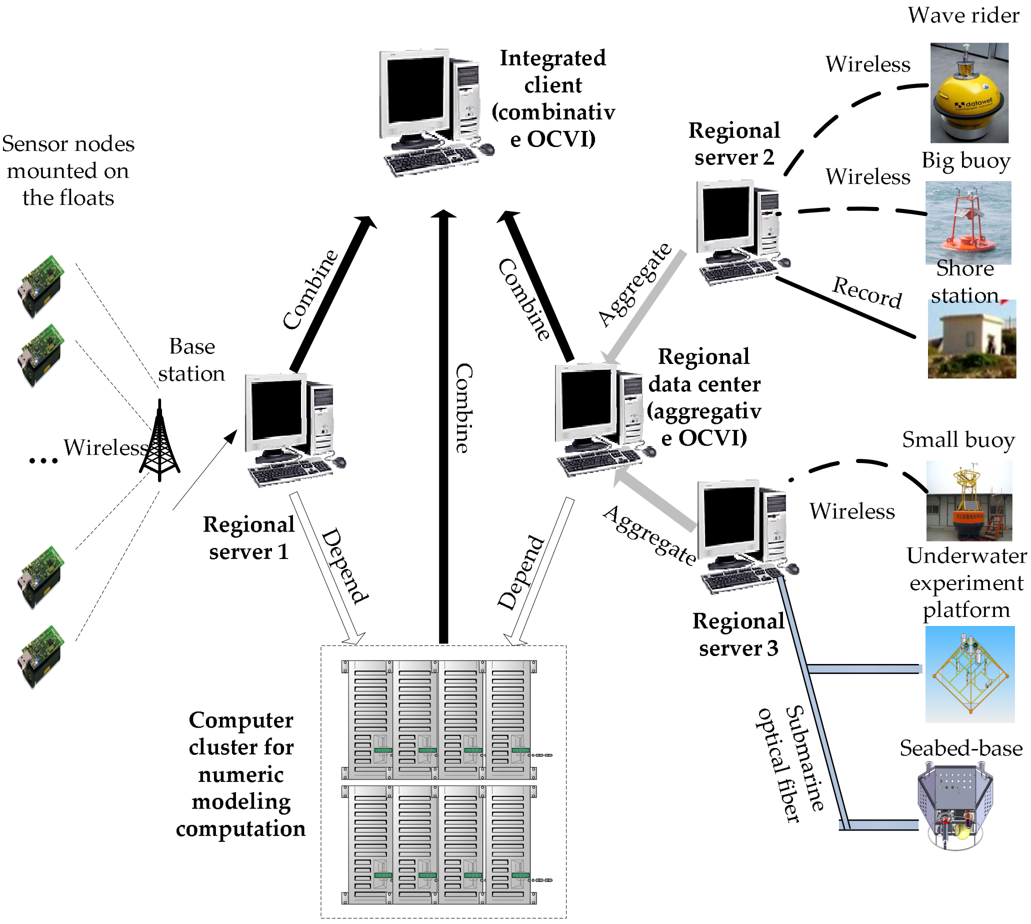


Figure 11. Ocean observing network architecture.

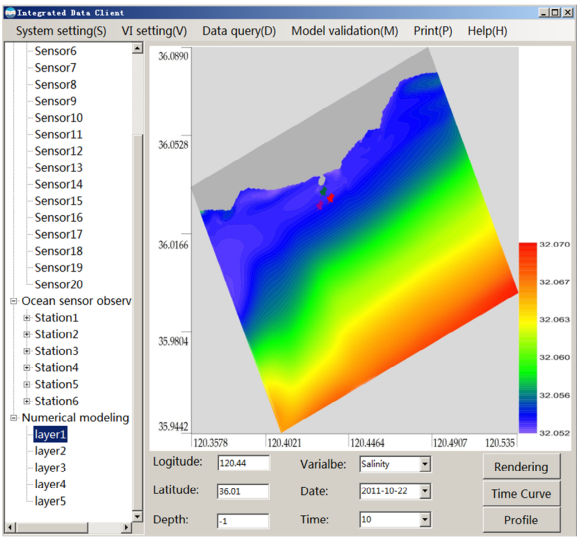
regional server 2 and 3 respectively. The data aggregation process is expressed as the addition of matrix:

$$\begin{pmatrix} V_{1,1} & \cdots & V_{1,64} \\ \vdots & \ddots & \vdots \\ V_{3,1} & \cdots & V_{3,64} \end{pmatrix} + \begin{pmatrix} V_{1,1} & \cdots & V_{1,70} \\ \vdots & \ddots & \vdots \\ V_{3,1} & \cdots & V_{3,70} \end{pmatrix} = \begin{pmatrix} V_{1,1} & \cdots & V_{1,70} \\ \vdots & \ddots & \vdots \\ V_{6,1} & \cdots & V_{6,70} \end{pmatrix}$$

Numerical modeling system is used for periodical forecast of meteorological and hydrological phenomena at the sea area. The meteorological numerical modeling adopts Weather Research Forecast (WRF) model, outputs the forecast results of wind, temperature, air pressure and relative humidity of sea surface and provides sensible heat fluxes, negative heat fluxes and precipitation to the hydrological numerical modeling. The Regional Ocean Modeling System (ROMS) model is used to forecast the hydrological phenomena including water temperature, salinity, wave and current. In both model, the observation data in OceanSense and ocean sensor observing system are assimilated to improve modeling results. All the forecast data are stored in the Network Common Data Form (NetCDF) and processed in the local OCVI. The data set at a time instant contains 182*182*5 values since the sea area is divided into 182*182 grids and five layers vertically.

The entire network architecture is illustrated in Fig. 11. The numerical modeling OCVI depend on other two OCVIs. It obtains temperature data from OceanSense OCVI and temperature, salinity and current data from the aggregative OCVI. These data are used in the process of numerical modeling computation. Three OCVIs are combined in the integrated client to provide access to all the observations and forecast data. With the assistance of configuration tools, the aggregation and combination operation is completed in several days. User interface of the combinative OCVI software is illustrated in Fig. 12. The integrated client displays the stations and the contained variables in each station. When users select the desired ones from the variable list, the client obtains the observation data from the corresponding system through service interfaces and visualizes them to users.

The client also provides the function to compare the current observations with the model results, the model results before and after calibration respectively. The ocean current observations are obtained from ocean sensor observing system. The comparison results are shown in Fig. 13.



Logitude: 120.44

Latitude: 36.01

Depth: -1

Varialbe: Salinity

Date: 2011-10-22

Time: 10

Figure 12. User interface of the combinative OCVI.

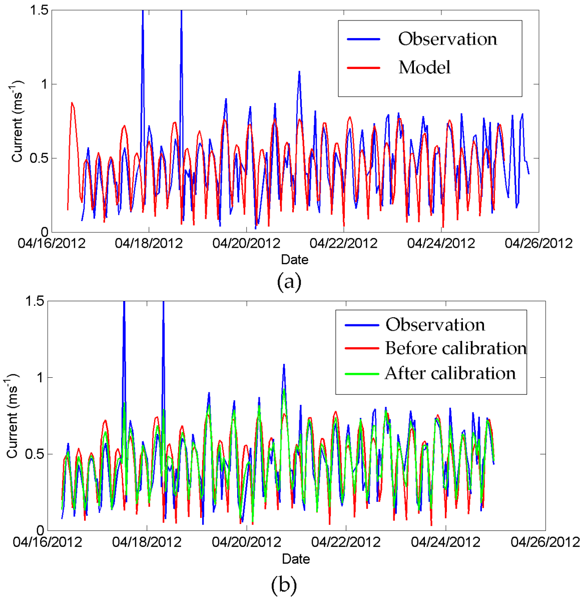


Figure 13. Numerical modeling validation for ocean current velocity: (a) comparision of observation and model results (b) compassion of observation results, model results before calibration and after calibration.

VI. Conclusions

In this paper, the design and implementation of complex virtual instruments for ocean observing are presented in detail. The unified data management, reusable VIs and supportive modules in OCVI design present resolutions to the problems in traditional VI systems. The implementation guides the designer to easy and visual developments. In order to reach the goal of GOOS, the OCVI composition model is proposed to formalize the composition operations between four roles. The problems in the composition process are mentioned to avoid redundant development efforts. Our further work will focus on simplifying the implementation process of the composition operations and broadening the application scope of OCVI.

Acknowledgment

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

References

[1] G. Xu, X. Shen, and K. Yu, “Modeling and hydrodynamic performance for a deep ocean manipulator based on numerical approach,” *Journal of Computers*, vol. 8, no. 5, pp. 1192–1199, 2013.

[2] H. J. Freeland, D. Roemmich, S. L. Garzoli, P.-Y. Le Traon, M. Ravichandran, S. Riser, V. Thierry, S. Wijffels, M. Belbeoch, and J. Gould, “Argo-a decade of progress,” in *OceanObs 09: Sustained Ocean Observations and Information for Society (Vol. 2), Venice, Italy, 21-25 September 2009*, 2010.

[3] R. Bassett, R. Beard, W. Burnett, R. Crout, B. Griffith, R. Jensen, and R. Signell, “Implementing the national integrated ocean observing system (ioos) from the federal agency perspective,” *Marine Technology Society Journal*, vol. 44, no. 6, pp. 32–41, 2010.

[4] R. Rayner, “The us integrated ocean observing system in a global context,” *Marine Technology Society Journal*, vol. 44, no. 6, pp. 26–31, 2010.

[5] S. Cox, “Observations and measurements-xml implementation,” *OGC document*, 2011.

[6] D. Snowden, M. Belbeoch, B. Burnett, T. Carval, J. Graybeal, T. Habermann, H. Snaith, H. Viola, and S. Woodruff, “Metadata management in global distributed ocean observation networks,” *Community White Paper for OceanObs*, vol. 9, 2009.

[7] L. Cristaldi, A. Ferrero, and V. Piuri, “Programmable instruments, virtual instruments, and distributed measurement systems: what is really useful, innovative and technically sound?” *Instrumentation & Measurement Magazine*, vol. 2, no. 3, pp. 20–27, 1999.

[8] H. Goldberg, “What is virtual instrumentation?” *Instrumentation & Measurement Magazine*, vol. 3, no. 4, pp. 10–13, 2000.

[9] L. Angrisani, P. Daponte, and M. D’Apuzzo, “A virtual digital signal-processing instrument for measuring superimposed power line disturbances,” *Measurement*, vol. 24, no. 1, pp. 9–19, 1998.

[10] J. J. G. de la Rosa, A. Agera-Prez, J. C. Palomares-Salas, J. M. Sierra-Fernndez, and A. Moreno-Mu?oz, “A novel virtual instrument for power quality surveillance based in higher-order statistics and case-based reasoning,” *Measurement*, vol. 45, no. 7, pp. 1824–1835, 2012.

[11] L. Fernandes, O. Coiado, R. Bassani, E. Costa, and S. Mhlen, “Virtual instrument for experimental studies in biological systems,” in *World Congress on Medical Physics and Biomedical Engineering May 26-31, 2012, Beijing, China*. Springer, 2013, pp. 1412–1415.

[12] J. Wang, X. Gao, and Q. Gao, “The design of wireless testing system for concrete rigidity based on virtual instrument,” in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*. IEEE, 2012, pp. 2968–2971.

[13] F. Toran, D. Ram?rez, A. Navarro, S. Casans, J. Pelegr?, and J. Esp??, “Design of a virtual instrument for water quality monitoring across the internet,” *Sensors and Actuators B: Chemical*, vol. 76, no. 1, pp. 281–285, 2001.

[14] Z. Zhao and R. Bai, “A fmcw radar distance measure system based on labview,” *Journal of Computers*, vol. 6, no. 4, pp. 747–754, 2011.

[15] V. S. Kovc, “Virtual instrumentation and distributed measurement systems,” *Journal of Electrical Engineering*, vol. 55, no. 1-2, pp. 50–56, 2004.

[16] B. Eaton, J. Gregory, H. Centre, B. Drach, K. Taylor, and S. Hankin, "Netcdf climate and forecast (cf) metadata conventions: http://cf-pcmdi.llnl.gov/," 2011.

[17] Z. Guo, C. Liu, X. Wang, H. Ma, Y. Jiang, B. Zheng, and B. He, "Cvis: Complex virtual instruments system architecture," 2013, in press.

[18] C. Liu and D. Liu, "Qos-oriented web service framework by mixed programming techniques," *Journal of Computers*, vol. 8, no. 7, pp. 1763–1770, 2013.

Keyong Hu received the B.S. degree in computer science and technology from Ocean University of China in 2008. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Ocean University of China, Qingdao, China, where he is researching ocean sensor web and underwater wireless sensor network.

Zhongwen Guo received the B.S. degree in computer science and technology from Tongji University, Shanghai, China, in 1987 and the M.S. and Ph.D. degrees from Ocean University of China, Qingdao, China. He is currently a Professor and Doctoral Advisor with the Department of Computer Science and Technology, Ocean University of China. His main research interests focus on sensor networks, distributed measurement systems, ocean monitoring, and so on.

Zhaodong Che is currently a researcher at network and information center, Ocean University of China, Qingdao, China. His current research interest includes computer network and wireless network security.

Feifei Shen received the B.S. degree in computer science and technology from Ocean University of China in 2007. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Ocean University of China, Qingdao, China, where he is researching underwater wireless sensor network and cloud computing.