Marine Data Exchange Prototype System Based on XML

PICES 14th Annual Meeting

Vladivostok,Russia 4th,Oct. 2005

Dengwen XIA

NMDIS/SOA/P.R.China

NMDIS Tianjin, China

Abstract

In this paper, a prototype system for ARGO data exchange is discussed. The system is based on XML, taking internet as the data exchange platform and XML as the data medium, which brings about a key technology of marine data integration and exchange.

1.Introduction

- For data exchange, it is most important that both sides of data exchange must arrive at a uniform understanding of data format. Only if using uniform data format, it is possible to execute the processes of carrying and dealing with data automatically. XML is a good data medium:
- XML Characteristics
  - flexible and Simplicity--easy to study and use
  - Openness--non-proprietary, not encumbered by copyright, patent, trade secret, or any other sort of intellectual property restriction
  - Self description--provides DTD and XML Schema for describing data structure
  - Platform independent--supported on virtually every platform, any programming language

- In this paper, the marine data exchange prototype is proposed, which is designed based on the XML, ASP and database technology,
- About ASP (Active Server Page)
  - A Microsoft technology for developing application based on web
  - Combine with XML and database technology perfectly
  - Can be viewed in any browser--ASP files are returned as plain HTML

2. Application analyzing

- Marine Data Exchange Prototype has been tested with ARGO data. ARGO floats are special measuring devices for building a global ocean observing system.
- About ARGO floats
  - The broad-scale global array of temperature/salinity profiling floats
  - Observing the ocean in real time
  - Providing a global temperature and salinity data of the upper 2,000 m of the ocean

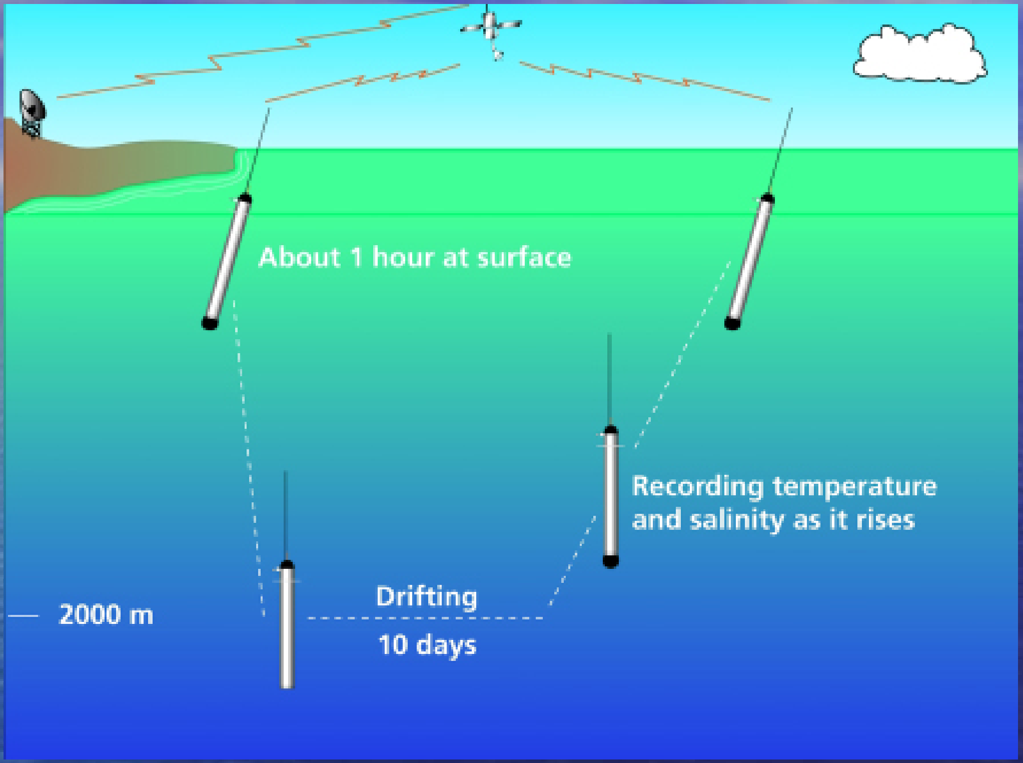
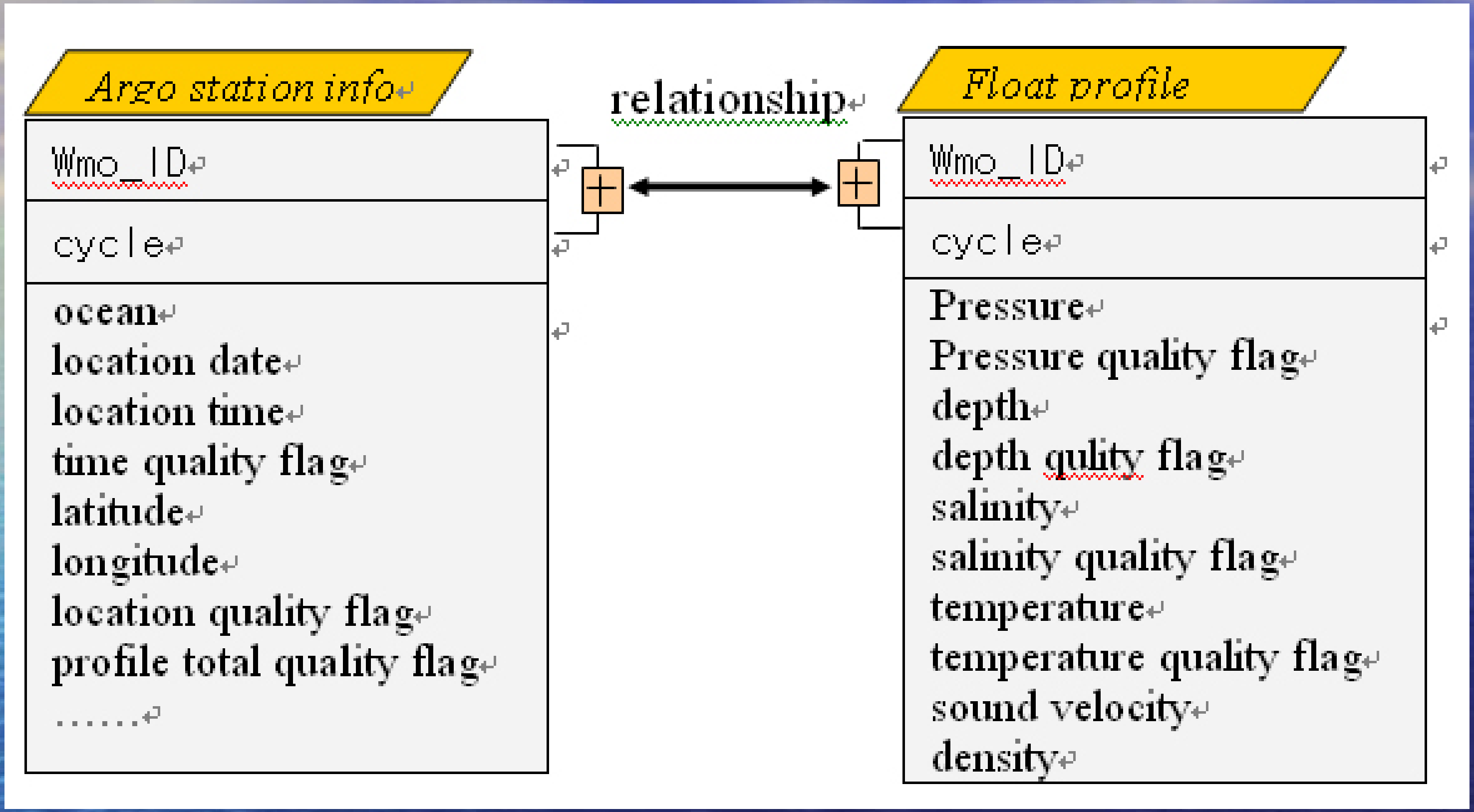


fig.1 Measure procedure of AOGO profiling floats

- The Argo data will help improve our ability to predict weather disaster (such as floods, droughts, unprecedented warm spells in winter) influenced by ENSO, and ability to understand the fluctuating climate or marine phenomena in the same way.
- NMDIS- National Marine Data and Information Service, as China ARGO Data Center, is responsible for collecting, processing and distributing the global ARGO data in real-time and delayed modes.

- NMDIS has established metadata database and ARGO profile database for Argo data management and online service.
- The Argo data in ARGO profile database are:
  - Stored in Microsoft SQL Server
  - Mainly including ARGO station information and ARGO profile data
  - ARGO station information is related to profile data by float ID (Wmo_ID) and Profile ID(cycle) (see fig.2)



Argo station info

Wmo_ID

cycle

ocean
location date
location time
time quality flag
latitude
longitude
location quality flag
profile total quality flag
……

relationship

Float profile

Wmo_ID

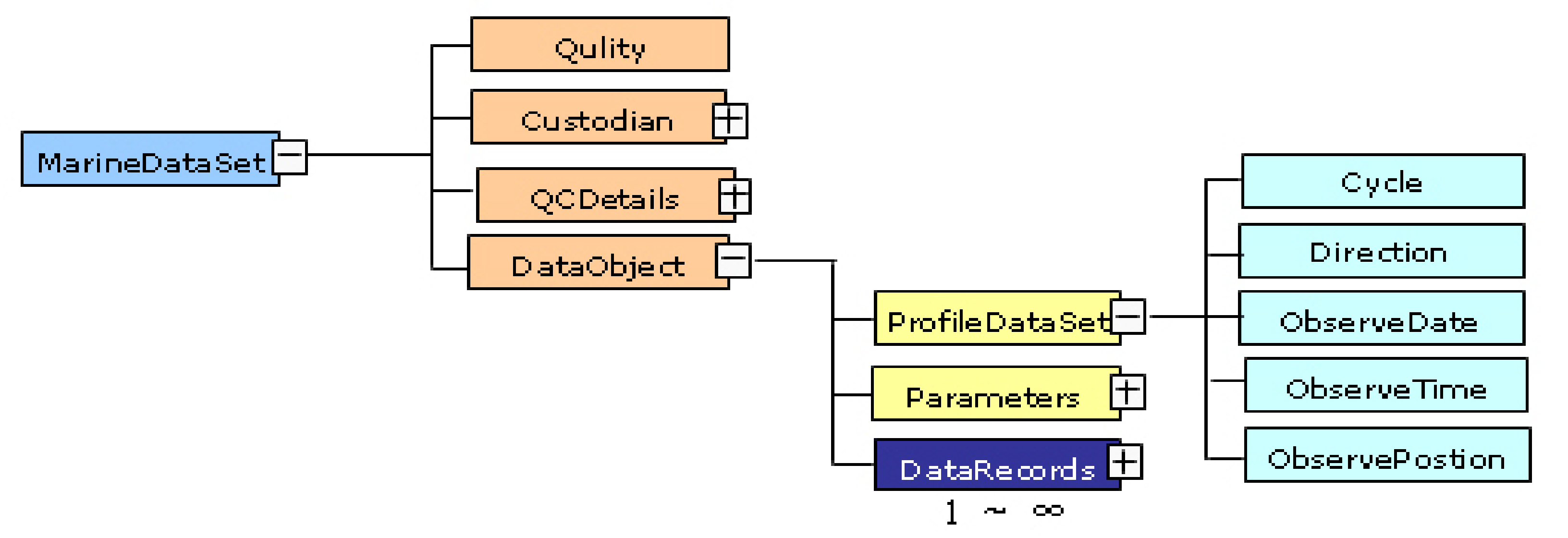
cycle

Pressure
Pressure quality flag
depth
depth qulity flag
salinity
salinity quality flag
temperature
temperature quality flag
sound velocity
density

fig.2 Structure of ARGO Profile Database

- XML is a structured description language, using a tree-structured storage method, and supports nested objects. Therefore it is possible to integrate Argo station information and float profile data into a XML document.
- Fig.3 is a sketch map of tree structure of Argo data XML document.

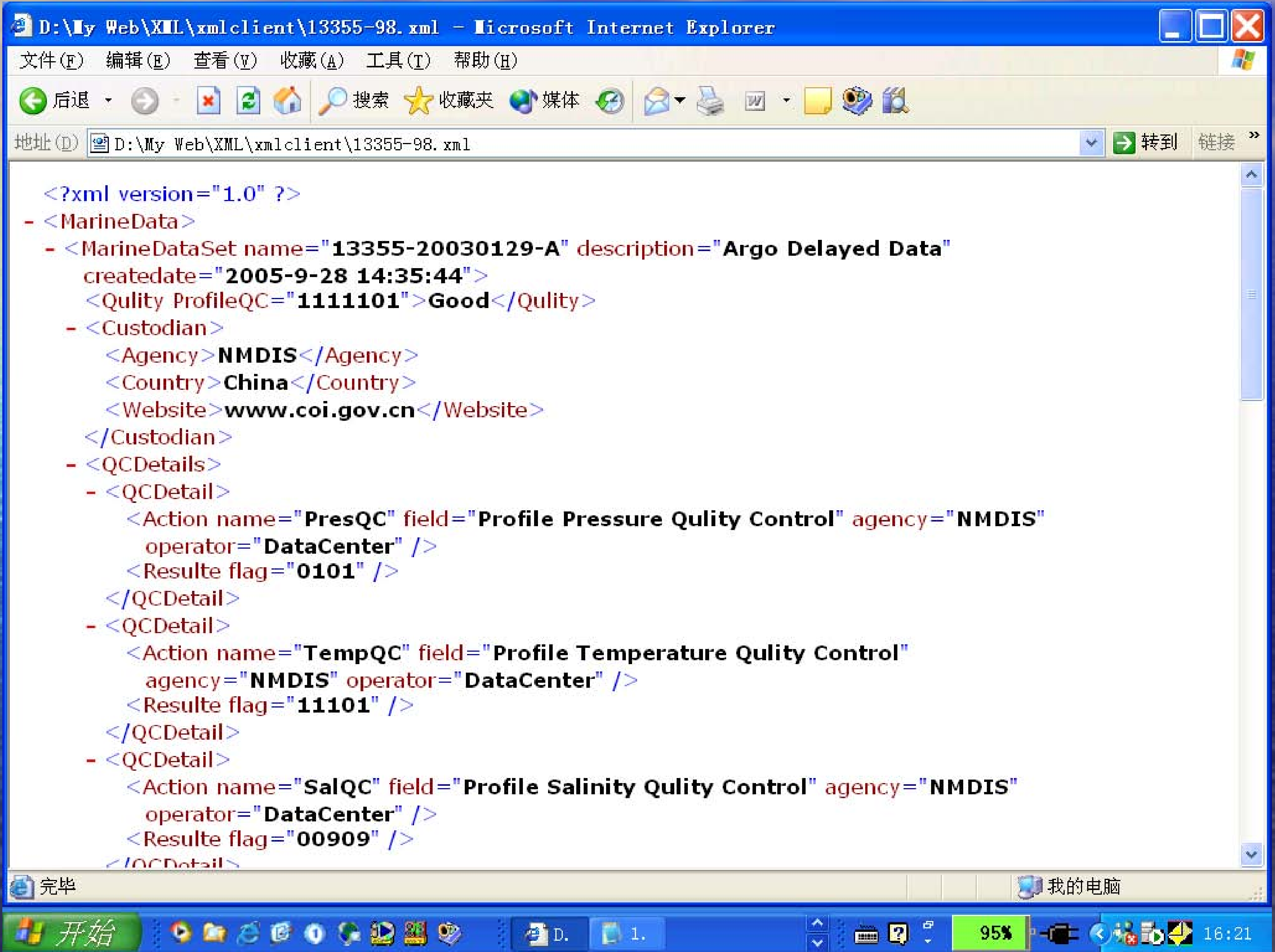
fig.3 Tree structure of ARGO Profile data XML document



- Element “Qulity” of “MarineDataSet” describe total quality about Argo profile, and Element “QCDetail” describe the quality of each item in the profile
- Element “ProfileDataSet” of “DataObject” encodes the station information (such as profile ID, observational date and observational location).
- Element “DataRecords” encodes all observational records of the profile

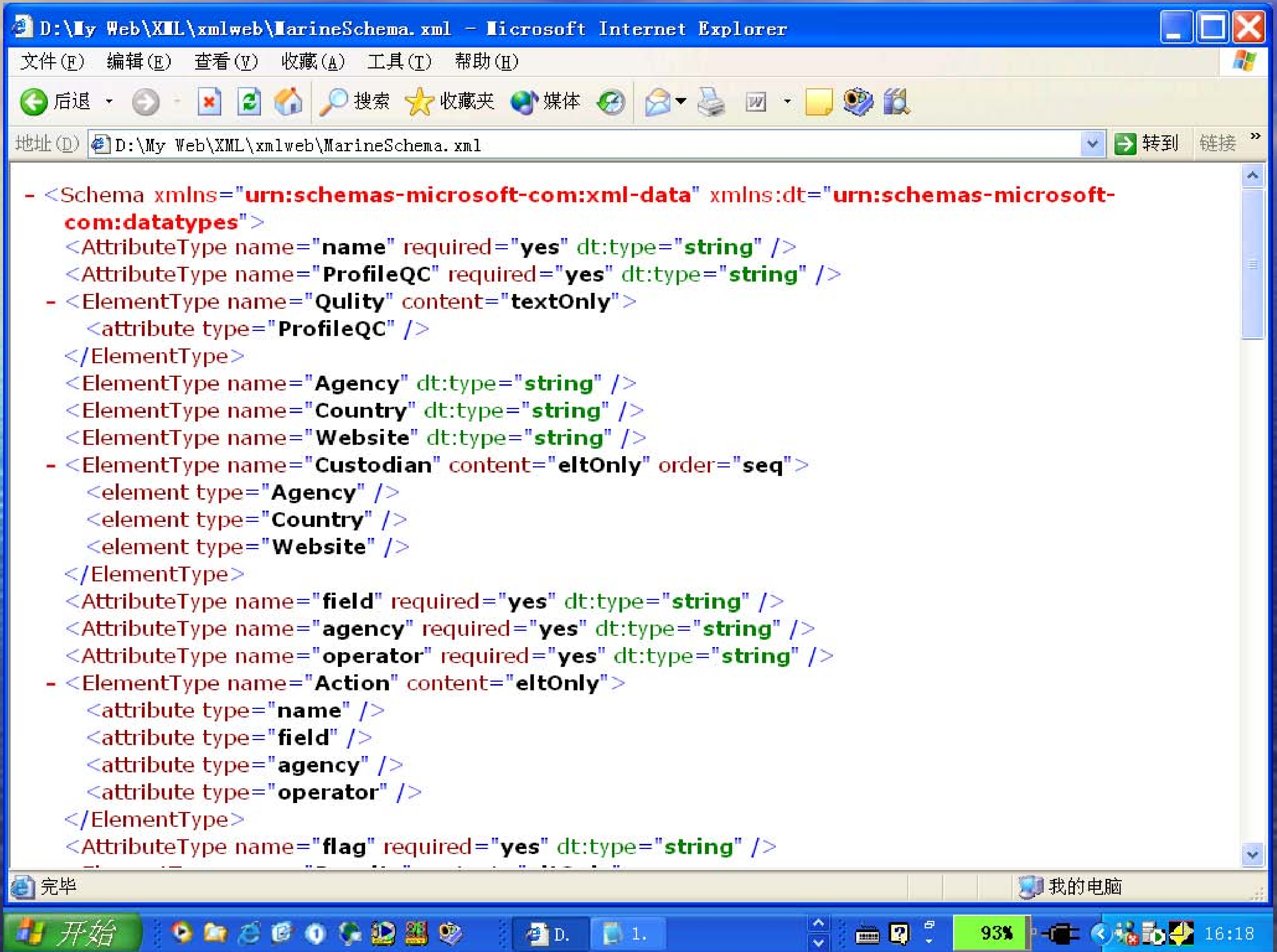
NMDIS Tianjin, China

Argo Profile data XML Document：


D:\My Web\XML\xmlclient\13355-98.xml - Microsoft Internet Explorer
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退
搜索
收藏夹
媒体
地址(D)
D:\My Web\XML\xmlclient\13355-98.xml
转到
链接
完毕
我的电脑
开始
D.
1.
95%
16:21

```
  <?xml version="1.0" ?>
- <MarineData>
  - <MarineDataSet name="13355-20030129-A" description="Argo Delayed Data"
      createdate="2005-9-28 14:35:44">
      <Qulity ProfileQC="1111101">Good</Qulity>
    - <Custodian>
        <Agency>NMDIS</Agency>
        <Country>China</Country>
        <Website>www.coi.gov.cn</Website>
      </Custodian>
    - <QCDetails>
      - <QCDetail>
          <Action name="PresQC" field="Profile Pressure Qulity Control" agency="NMDIS"
            operator="DataCenter" />
          <Resulte flag="0101" />
        </QCDetail>
      - <QCDetail>
          <Action name="TempQC" field="Profile Temperature Qulity Control"
            agency="NMDIS" operator="DataCenter" />
          <Resulte flag="11101" />
        </QCDetail>
      - <QCDetail>
          <Action name="SalQC" field="Profile Salinity Qulity Control" agency="NMDIS"
            operator="DataCenter" />
          <Resulte flag="00909" />
        </QCDetail>
```

Argo Profile data XML Schema:



D:\My Web\XML\xmlweb\MarineSchema.xml - Microsoft Internet Explorer
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退
搜索
收藏夹
媒体
地址(D) D:\My Web\XML\xmlweb\MarineSchema.xml
转到
链接
完毕
我的电脑
开始
D.
1.
93%
16:18

```
- <Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-
    com:datatypes">
    <AttributeType name="name" required="yes" dt:type="string" />
    <AttributeType name="ProfileQC" required="yes" dt:type="string" />
  - <ElementType name="Qulity" content="textOnly">
      <attribute type="ProfileQC" />
    </ElementType>
    <ElementType name="Agency" dt:type="string" />
    <ElementType name="Country" dt:type="string" />
    <ElementType name="Website" dt:type="string" />
  - <ElementType name="Custodian" content="eltOnly" order="seq">
      <element type="Agency" />
      <element type="Country" />
      <element type="Website" />
    </ElementType>
    <AttributeType name="field" required="yes" dt:type="string" />
    <AttributeType name="agency" required="yes" dt:type="string" />
    <AttributeType name="operator" required="yes" dt:type="string" />
  - <ElementType name="Action" content="eltOnly">
      <attribute type="name" />
      <attribute type="field" />
      <attribute type="agency" />
      <attribute type="operator" />
    </ElementType>
    <AttributeType name="flag" required="yes" dt:type="string" />
```

3. Prototype system designing

- Features of the Prototype system:
  - Prototype system using client/server mode, the server side receives, validates and deals with ARGO XML data uploaded from clients and put them into database.
  - Users not only can upload ARGO XML data, but also can query, browse server-side ARGO profile data using browser, and then translate them into ARGO XML format automatically on demand, and so much as download to client side.

- Data flow chart of prototype system Showing data exchange flow between Argo database and ARGO XML documents.
  - Step 1 and step 2 describe the process of converting ARGO XML documents into ARGO database.
  - Step 3 and step 4 describe the process of translating ARGO data in database into ARGO XML documents.

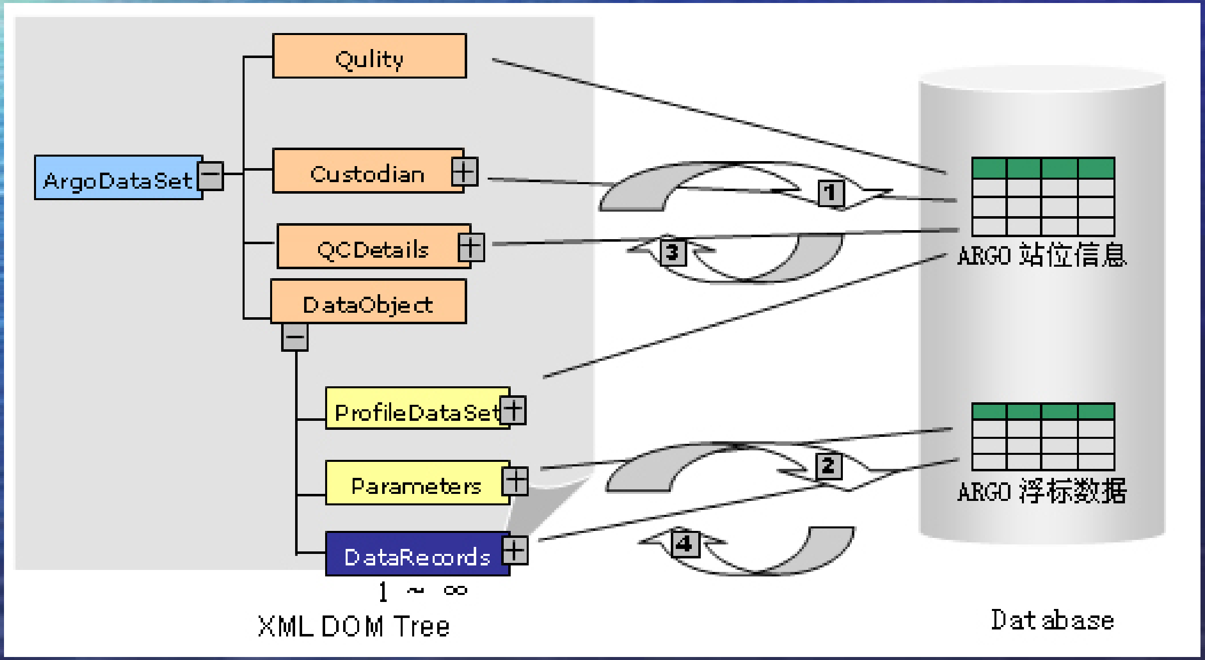


fig.4 Data flow chart of prototype system

- Prototype System architecture
  - The process of data exchange includes uploading and delivering ARGO XML profile data

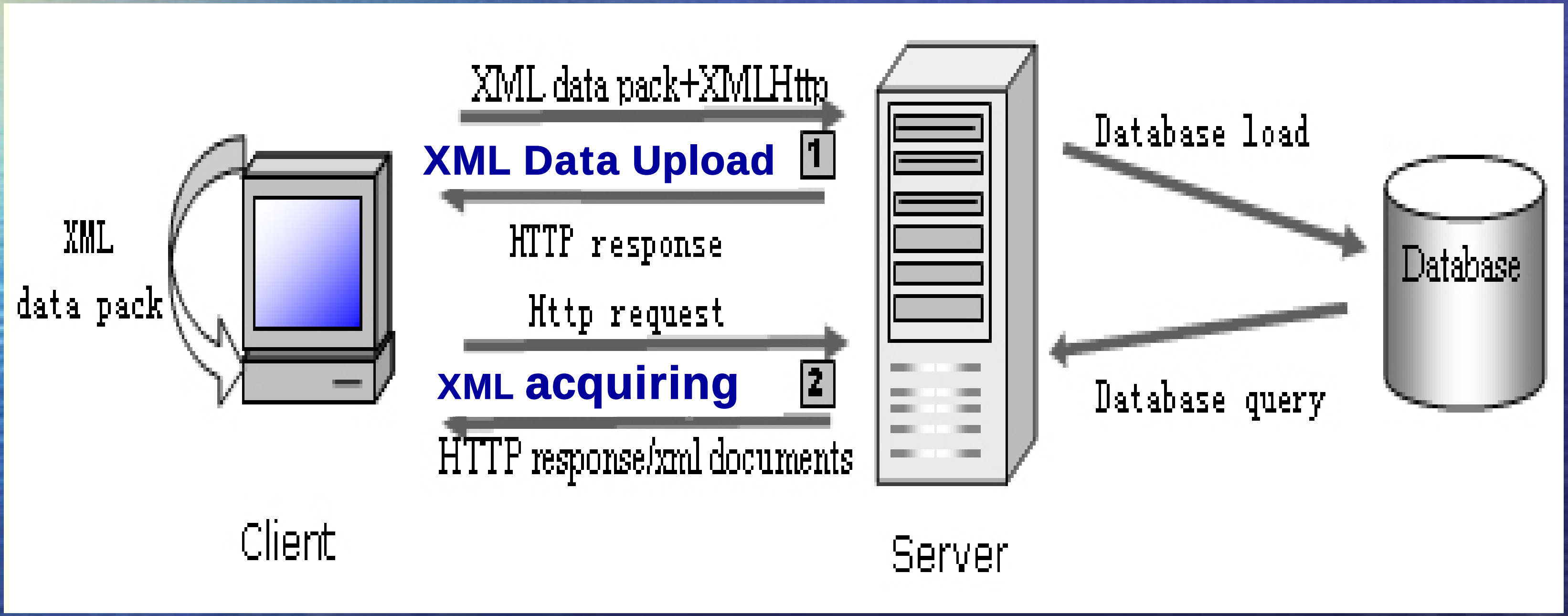


fig.5 Prototype Systems Architecture

ARGO XML data upload

- Client side application load ARGO XML data storing in local to an object of XMLHTTP, and then submit a request to server ASP page in web server
- server execute ASP page, validating ARGO XML data uploaded and making relevant process.
- the server application (ASP page) loads XML data to database and return a true message to client side
- Or return a error message to client side.

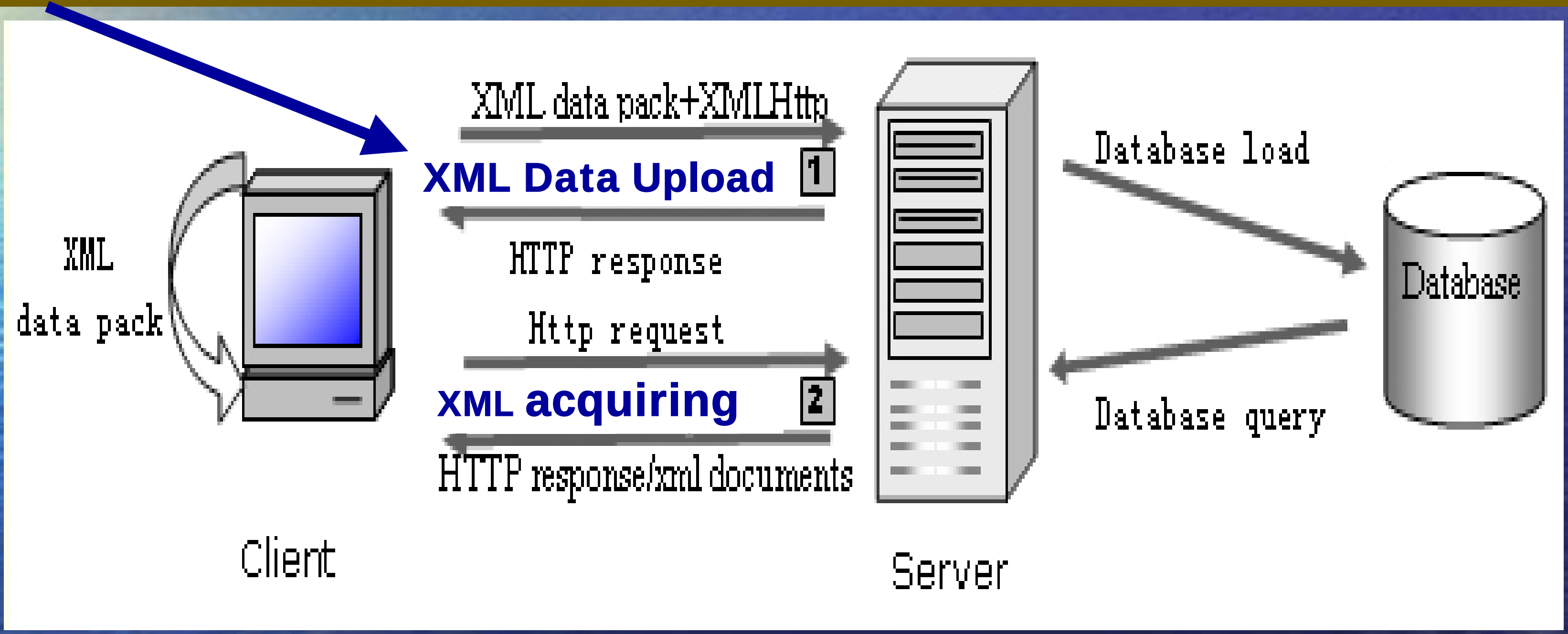


fig.5 Prototype Systems Architecture

ARGO XML acquiring

- Client side browser request a page in web server, input search conditions, submit to ASP page in server side,
- Server execute ASP page and return ARGO profiles from database to browser on client side.
- Users select the profile data needed, the server generates XML format from database dynamically for users to download.

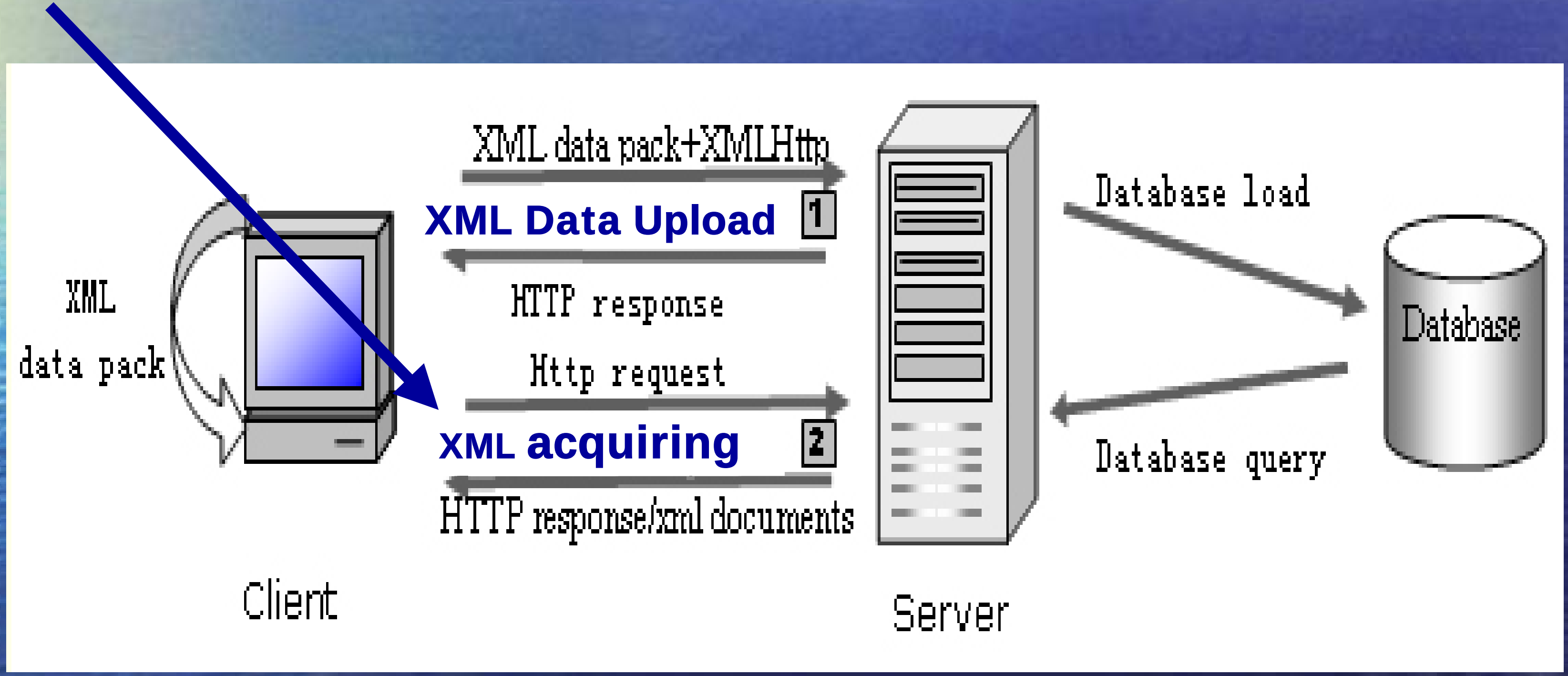


fig.5 Prototype Systems Architecture

Server side designing

- Server side has three tasks:
  - the first is receiving ARGO XML data uploaded from client side,
  - the second is searching ARGO database and returning profile data list to client side browser,
  - the last is dynamically generating XML data from database and returning the data to client side.

—ARGO XML data receiving

Step1: Creating a XMLDOM object by XML parser (MSXML) to receive data packs uploaded from client side

Step2:By setting name space attribute of XMLDOM tree as the URL of XML Schema storing on server side, server executes the validation to ARGO XML data automatically

Step3:Loading node data on XMLDOM object to ARGO database following the data transform rules.

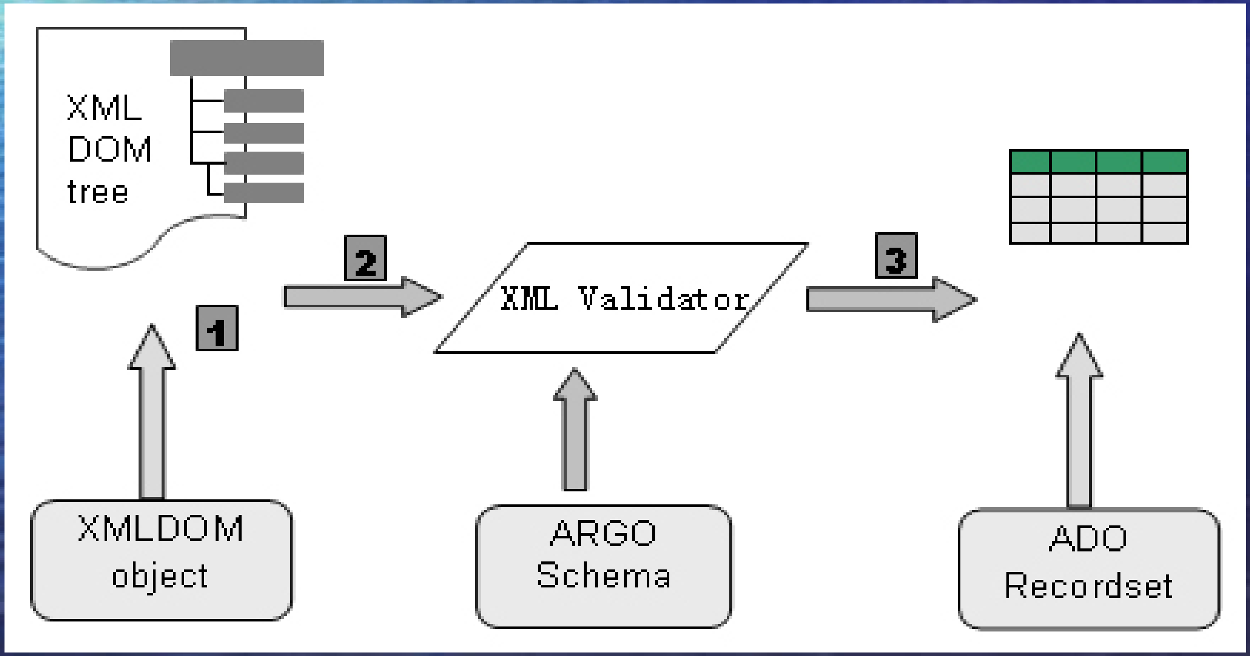


fig.6 Receiving and processing XML data packages

—ARGO database searching

Step1: The client side submits HTTP request (for search) to the application (ASP page) on server side

Step2:The server executes application (asp page) and generates SQL statements automatically, store the query result to a instance of "RecordSet" object

Step3:Dynamically generates HTML page (including list of ARGO station information), and return it to the client side

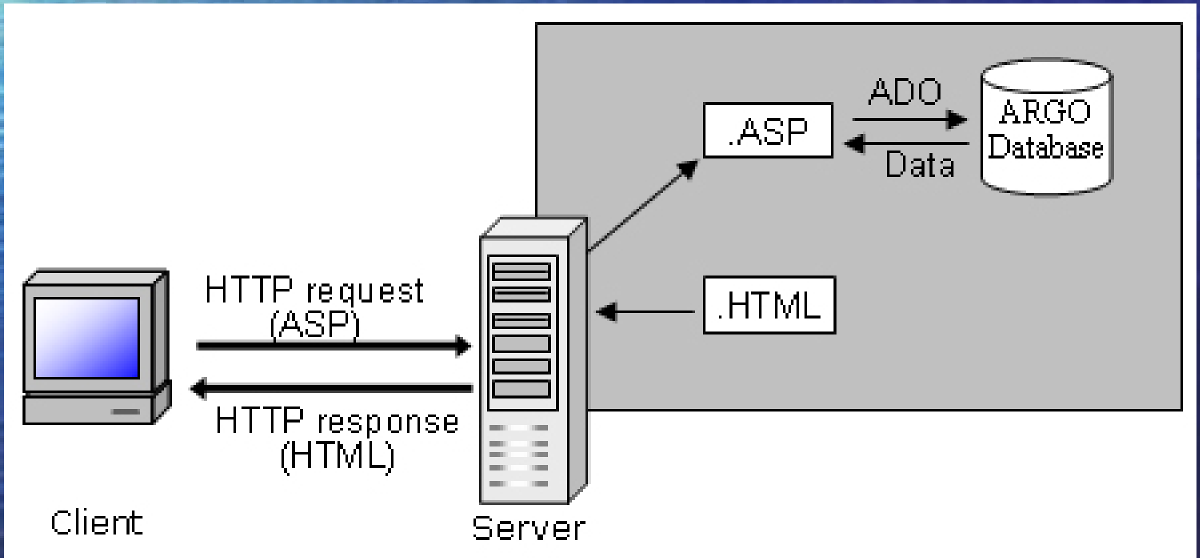


fig.7 Flow chart of Database Processing

— Generating ARGO XML documents dynamically

Step1:Creating a ADO object for access to ARGO database, store profile records selected by users via “RecordSet” object

Step2,3: create a XMLDOM object, encode result to XML format and return to client following the data transform formula.

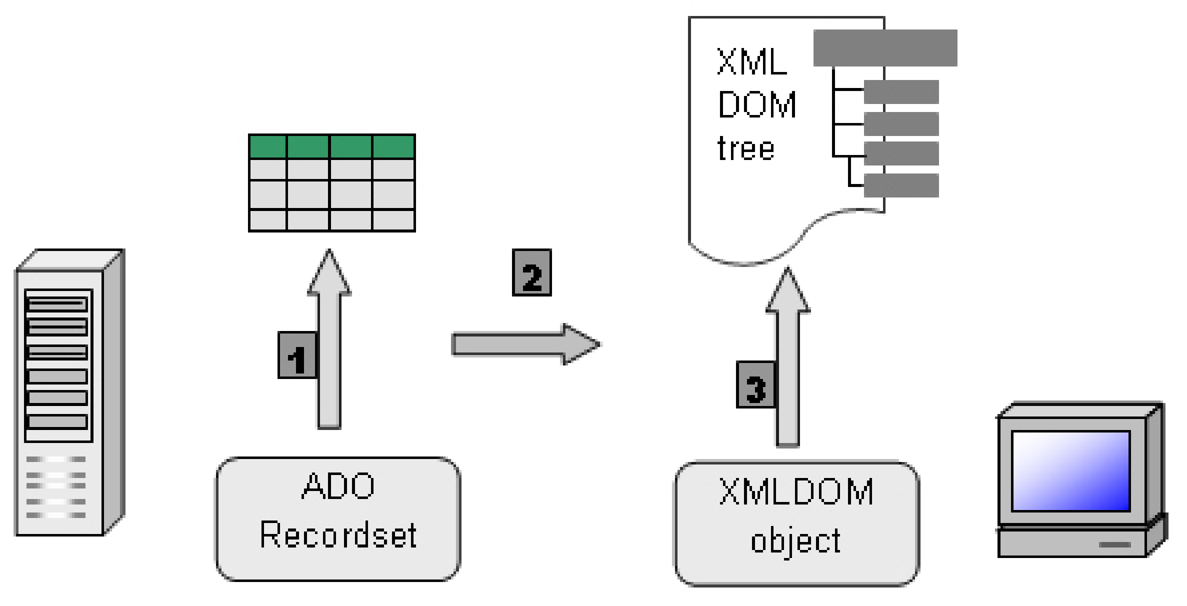


fig.8 Generating XML documents from database dynamically

Client side designing

- Step1:Building a XML data pack on client side, the source of the data pack may be any one XML document or a piece of XML document , or even a XML document dynamically generated information acquired from server by users;
- Step2: Creating a XMLDOM Object, using the MSXML parser as the carrier of ARGO XML data on client side;
- Step3,4,5: Creating a XMLHTTP object (as shown in fig.9(2)) for sending ARGO XML data pack to application (ASP page) on server side(fig.9(3),(4) and(5)) and prepare for receiving response message at the same time.

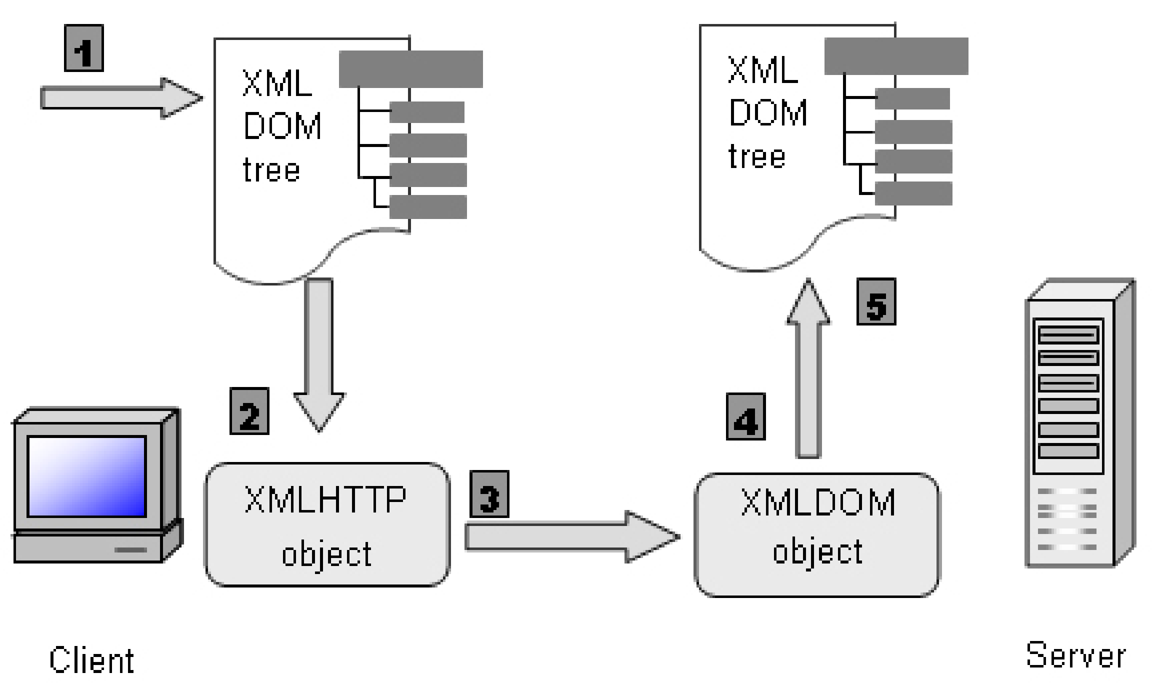


fig.9 Flow chart of transmitting XML data packages

4.Prototype system example

- Upload XML data
  - In this case XML data uploading is implemented by means of uploading XML document storing on client side to the server
  - Users input the full path of local XML document in “File Name” textbox,
  - Clicking “Submit” button, then the XMLHTTP object will upload XML document to destination i.e. the server application

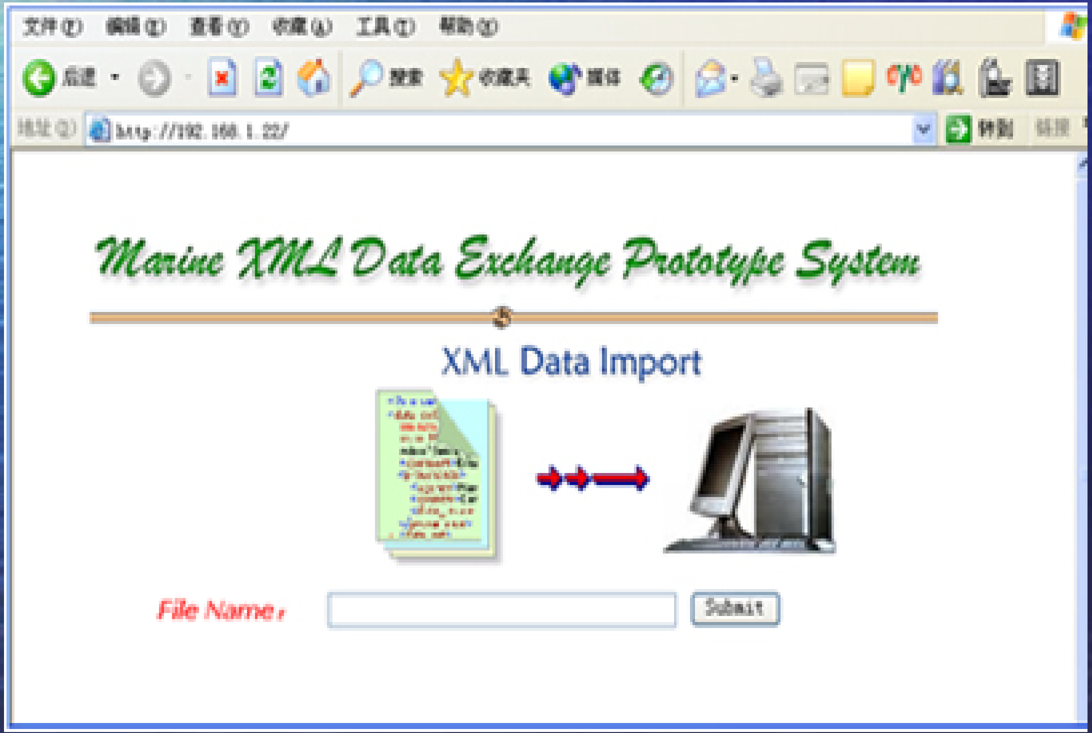
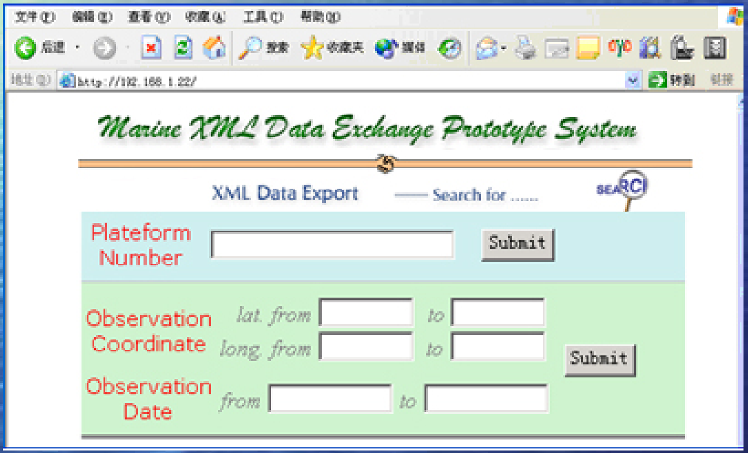

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退
搜索
收藏夹
媒体
地址(D)
http://192.168.1.22/
转到
链接
Marine XML Data Exchange Prototype System
XML Data Import
File Name:
Submit

fig.10 Upload XML data

- Search and Browse ARGO data
  - We can make a query by ARGO float number (WMO number), We can also make a query by the date and location (longitude and latitude) (fig.11)
  - All ARGO profiles are listed in fig.12 according to search conditions above.
  - The last column is the XML icons, if you click one of them, the application on server side will generates a ARGO XML document of this float profile and return it to client side automatically (see fig.13)



Marine XML Data Exchange Prototype System

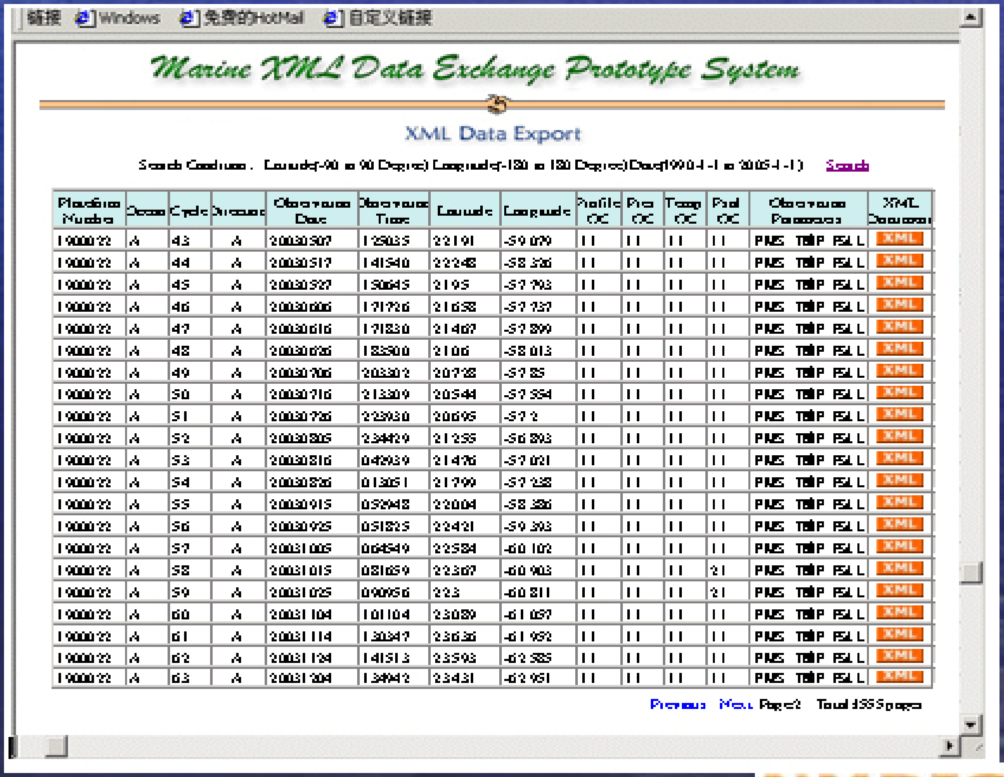
XML Data Export — Search for ......

Plateform Number [   ] Submit

Observation Coordinate lat. from [   ] to [   ]
long. from [   ] to [   ] Submit

Observation Date from [   ] to [   ]

fig.11 Search for ARGO data



Marine XML Data Exchange Prototype System

XML Data Export

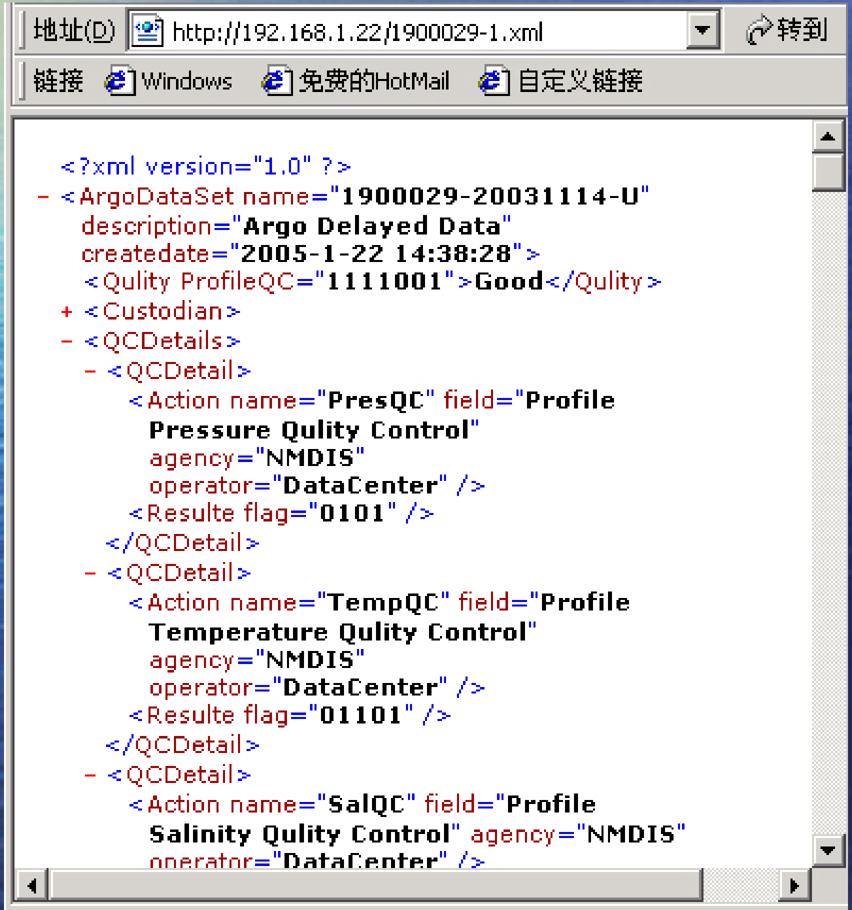
Search Condition : Latitude(-90 to 90 Degree) Longitude(-180 to 180 Degree) Date(1990-4-1 to 2005-4-1) Search

Platform Number	Ocean	Cycle	Direction	Observation Date	Observation Time	Latitude	Longitude	Profile QC	Pres QC	Temp QC	Psal QC	Observation Parameters	XML Document
1900022	A	43	A	20030507	125035	22.191	-59.079	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	44	A	20030517	141540	22.248	-58.326	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	45	A	20030527	150645	21.95	-57.793	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	46	A	20030606	171726	21.658	-57.737	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	47	A	20030616	171830	21.407	-57.899	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	48	A	20030626	183500	21.06	-58.013	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	49	A	20030706	203302	20.728	-57.85	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	50	A	20030716	213309	20.544	-57.554	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	51	A	20030726	223930	20.695	-57.2	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	52	A	20030805	234429	21.255	-56.893	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	53	A	20030816	042939	21.476	-57.021	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	54	A	20030826	013051	21.799	-57.238	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	55	A	20030915	052048	22.004	-58.326	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	56	A	20030925	051825	22.421	-59.393	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	57	A	20031005	064549	22.584	-60.102	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	58	A	20031015	081059	22.307	-60.903	11	11	11	21	PRES TEMP PSAL	XML
1900022	A	59	A	20031025	090056	22.3	-60.811	11	11	11	21	PRES TEMP PSAL	XML
1900022	A	60	A	20031104	101104	23.089	-61.057	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	61	A	20031114	130347	23.636	-61.952	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	62	A	20031124	141513	23.593	-62.585	11	11	11	11	PRES TEMP PSAL	XML
1900022	A	63	A	20031204	134942	23.431	-62.951	11	11	11	11	PRES TEMP PSAL	XML

Previous Next Page:2 Total 1355 pages

fig.12 Browse ARGO data

- Generating XML documents dynamically
  - The XML document generated dynamically is returned to client browser
  - Users click menu “file” and select item “save” in browser, The XML data of this profile are downloaded to client side



```
  <?xml version="1.0" ?>
- <ArgoDataSet name="1900029-20031114-U"
    description="Argo Delayed Data"
    createdate="2005-1-22 14:38:28">
    <Qulity ProfileQC="1111001">Good</Qulity>
  + <Custodian>
  - <QCDetails>
    - <QCDetail>
        <Action name="PresQC" field="Profile
          Pressure Qulity Control"
          agency="NMDIS"
          operator="DataCenter" />
        <Resulte flag="0101" />
      </QCDetail>
    - <QCDetail>
        <Action name="TempQC" field="Profile
          Temperature Qulity Control"
          agency="NMDIS"
          operator="DataCenter" />
        <Resulte flag="01101" />
      </QCDetail>
    - <QCDetail>
        <Action name="SalQC" field="Profile
          Salinity Qulity Control" agency="NMDIS"
          operator="DataCenter" />
```

fig.13 Dynamically generating XML documents

5. Conclusions

- The prototype system proposed in this paper shows that marine data exchange based on XML is a feasible method for establishing a marine data exchange system.
- Marine data exchange technology based on XML can be widely applied to data delivery, data exchange and information sharing of all kinds of marine observing and monitoring data.

THANKS FOR YOUR ATTENTION!