Distributed Agricultural Model System

Takuji Kiura(kiura@affrc.go.jp)
Masayuki Hirafuji(hirafuji@narc.affrc.go.jp)
Kei Tanaka(ketanaka@narc.affrc.go.jp)
Matthew Laurenson(matthewl@narc.affrc.go.jp)
National Agriculture Research Center
3-1-1, Kannondai Tsukuba-Shi
Ibaraki 305-8666
JAPAN
Ph +81-298-38-8975
Fax +81-298-38-8551

Shigehiro Honda(honda@mi.mss.co.jp.)
Michihiko Shimomura(shimomura@mi.mss.co.jp)
Mitsubishi Space Software Co., Ltd
Tsukuba Division
1-17-15, Sengen Tsukuba-Shi
Ibaraki 305-0047
JAPAN
Ph +81-298-59-0311
Fax +81-298-59-0321

Abstruct:
In 1997 the project "Fundamental research of Reproductive Information-Base(RIB) to develop agricultural production assistance systems" was started. The RIB consists of distributed information servers on which we store information such as cases, facts and computer models. The project's main themes are (1)Fundamental research to develop RIB, and (2)Fundamental research to develop integrated huge agricultural computer models. "Distributed Agricultural Model System" under the latter theme is a Java framework for integrating models and database.

Key words: Java, Internet, distributed, model, database

A number of models and databases (datasets) have been developed in the field of agriculture. The original versions of applications listed in the section "Fundamental models as remote objects" successfully linked a fundamental model and a dataset, but their adoption did not grow. Possible reasons include:

1. Difficulty in finding an appropriate program to use.
2. Models being implemented on standalone PCs.
3. Model implementations depending on a particular data set format.
4. Stored data set formats differing between implementations of the same model.

To overcome the first problem, we collected information on software and database developed by Japanese agriculture institutes and made a web site named 'Software and database index of Japanese agricultural institutes'
http://ss.narc.affrc.go.jp/~kiura/meta_index/index.html (Japanese).

With regards to the second problem, conventional standalone agricultural software requires that users install software on to their own PC. Typically only one family of operating systems is supported (eg MS-Windows), and sometimes only a specific version (eg Windows NT). The agricultural software may require legacy hardware, shared code libraries (eg Borland Database Engine for database access), or a particular version of an operating system extension. Updating the operating system, or shared code libraries to be compatible with the agricultural program may stop older applications working.

The third problem means that, even if we can find a suitable program and install it correctly, we still need to get data from somewhere and convert it to a specific file format. Finding the data and converting it may be beyond the capabilities of most users.

The fourth problem means that we can't easily switch between model implementations (eg to compare model outputs). We may need to convert all the data sets we have. It also means that

we can’t use output files from one program as input files to another.

“Distributed Agricultural Model System" is proposed as an approach to solve these problems and develop integrated agricultural models. It uses the programming language Java, which has a number of useful features. Compiled Java classes can be run without modification on most operating systems in use today, provided the computer has a Java Virtual Machine (JVM) installed. Web browers such as Netscape and Internet Explorer include a JVM, and can download and execute Java code in the form of applets without any user intervention. Java’s built-in communications facilities such as Remote Method Invocation (RMI) mean that Java applications can easily access other Internet-based resources such as databases and server applications.

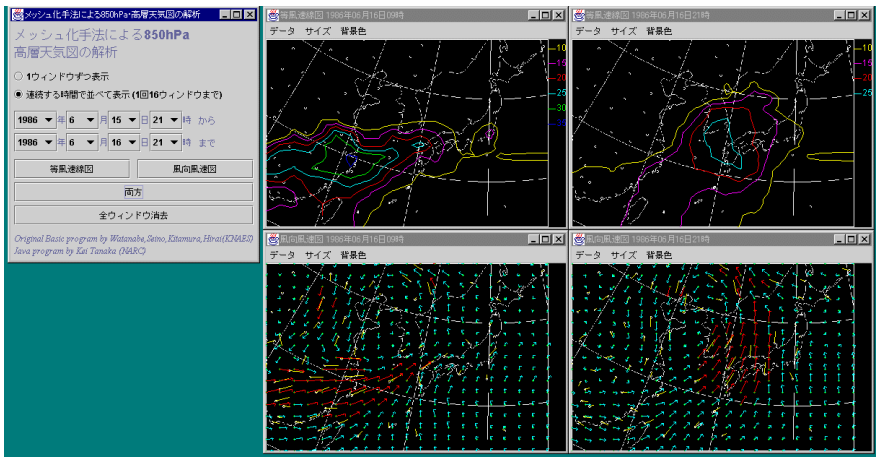


Fig. 1a LLJet calculates and displays high level winds conducive to pest migration.

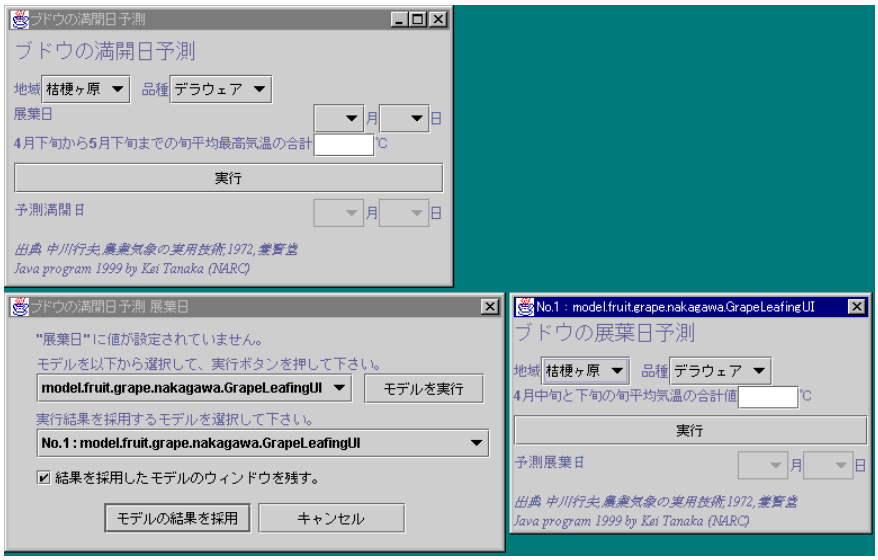
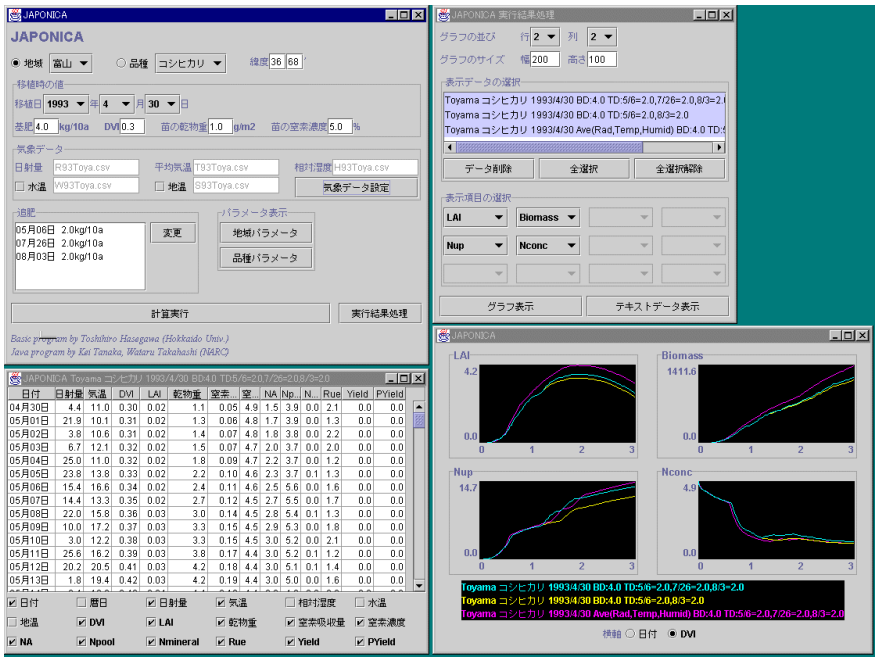


Fig. 1b Grape leaf development model



Materials and Methods

1. Fundamental models as remote objects

We collected fundamental models in the agricultural field. These legacy models had been written in various programming languages, such as Basic, C, or C++. We rewrote the models and also wrote new models, as Java object first, and then as Java applets(Fig.1). Because most standard web browsers support Java applets, users can run the models without any installation or associated risk to their existing applications and operating system. Finally, we have wrapped the fundamental models as Java RMI remote objects, so that we can integrate them with other Java programs.

2. "Model base" to store the fundamental models (not completed)

Model base is an object oriented database which stores fundamental models. Users can restore any model together with its links to other fundamental models. Model base is not yet completed, but a user interface to retrieve an appropriate model has been implemented using the conceptual retrieval technique (Hirafuji et.al. 2000).

3. Data sets as objects

By making every source of data an object, differences in data file format and database implementation can be hidden from the models that use the data. We developed wrapper classes which present existing remote data source as Java RMI servers. Data from these servers can be transmitted to fundamental models using RMI by making serializable objects that encapsulate data of the same kind within a consistent interface. The fundamental RMI wrapper objects for servers and datasets are re-used by multiple data sources and datasets respectively.

Similar techniques have been used in a mechanism for retrieving meteorological data from heterogeneous sources called MetBroker (Laurenson et.al. 2000).

4. Directory services to find remote objects

"Directory base" is a distributed database, which stores the properties of remote objects (data sources and fundamental models) currently available. A Java application can look up the URLs of required fundamental models using directory base. If any required object is not found in the local directory, it connects to the upper directory and tries to find it. The prototype directory base was also developed as a RMI remote object. Fig. 2 shows the structure of the distributed agriculture model system.

5. "Workspace" to link the fundamental models and data sets(not yet implemented)

Workspace is a Java application to develop integrated models (Fig. 3). It connects to Directory base and finds required fundamental models and data objects. It can link the properties of fundamental models to other remote objects. It also generates user interfaces for fundamental models. The models developed in Workspace will be tested and stored in the integrated model base (not yet implemented).

Acknowledgement

This study was funded by the Japan' ꞩational research project of MAFF (Ministry of Agriculture, Forestry and Fisheries), "Fundamental Research of Reproductive Information-Base to Develop Agricultural Production Assistance Systems" (http://model.narc.affrc.go.jp/RIP.htm).

References

Hirafuji, M., K. Tanaka, et al. (2000). Modelbase System : A Distributed Model Database on The Internet. Pre-Proceeding: Application Area - IWS 2000 International Workshop on Asia Pacific Advanced Network and its Applications., Tsukuba , Japan. 57-61.

Laurenson, M., T. Kiura, et al. (2000). Accessing online weather database from Java. Pre-Proceeding: Application Area – International Workshop on Asia Pacific Advanced Network and its Applications., Tsukuba , Japan. 193-198.

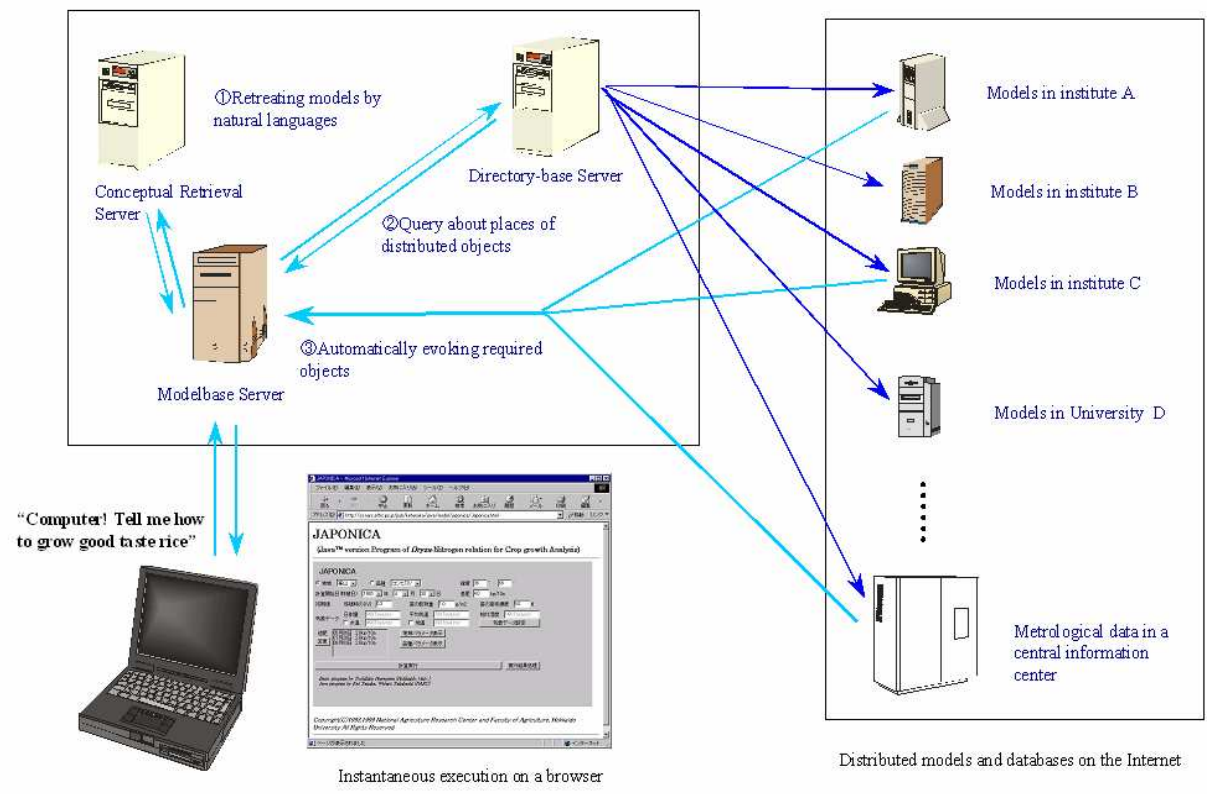


Fig. 2 Structure of the distributed agricultural model system

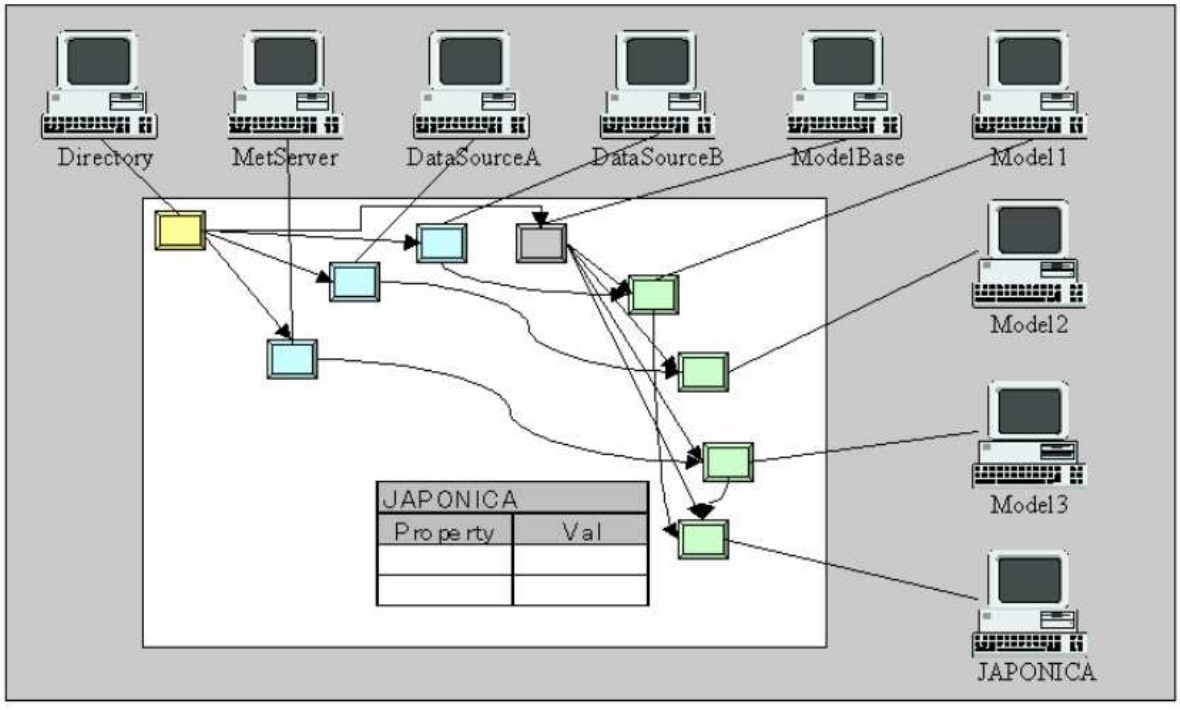


Figure 3 Image of WorkSpace