Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212
Ubiquitous International Volume **5**, Number **1**, January **2014**

Design of Data Archive in Virtual Test Architecture

Lian-Lei Lin, Ling-Yu Li, and Yue Ma
Department of Automatic Test and Control
Harbin Institute of Technology
Harbin, China
E-mail:linlianlei@hit.edu.cn

Received April, 2013; revised September, 2013

Abstract. *Combined with the new demands in test and training field, we proposed a new architecture HIT-TENA according to HLA and TENA architecture. In order to manage all kinds of test data in HIT-TENA, we designed a four-tiered hierarchical HIT-TENA Data Archive. This paper introduces the construction of Data Archive in detail, including the design of database, the design of Data Archive Manager and the design of Data Archive Web Service. Now, HIT-TENA Data Archive has been running on the platform, to good effect.*
Keywords: HIT-TENA; Data Archive; database; Data Archive Manager; Web Service

1. **Introduction.** At present, development based on architecture has become a general part of engineering practice [1]. A public architecture can effectively promote the reuse, interoperability and combination of the resources. Therefore, the architecture of the virtual test system is becoming distributed, open and interactive. In the field of modeling and simulation, the United States Department of Defense established the High Level Architecture (HLA). And in the field of combat and training, the United States developed the Test and Training Enable Architecture (TENA). We put forward HIT-TENA architecture after the study of HLA and TENA architecture, combined with the new demands in test and training field.

For different kinds of architectures or test systems, the management of simulation data has always been a very important task. The simulation data usually includes various observation and measurement data, test operation data, documentation, result reports and metadata. Simulation data is the product and core value of simulation with very high reference value and guidance value for the subsequent simulation activities and scientific research [2]. With the increase of the simulation systems, models, algorithms and data involved are becoming more and more. In order to ensure the sharing and synchronization of the test information, a unified data archive is necessary.

In 1995, the United States Department of Defense put forward the Modeling and Simulation Master Plan (MSMP), in which they proposed to construct Modeling and Simulation Resource Repository (MSRR): Create a distributed MSRR system, conveniently to provide timely, checked and effective data, metadata, algorithm, model, simulation applications, toolkit and some background information for modeling and simulation [3].In the same year, the United States used the data management in industrial fields. In 2000, a reservoir resource simulation data management system of Exxon Mobil Corporation was brought into use [2]. Later, new simulation data management solution appeared constantly, such as MSC company's SimManager[4] and AMSYS's enterprise cooperative R&D

platform PERA[5]. In the simulation field, the development of HLA largely promoted the management of simulation data [6]. And the core of TENA is the TENA Common Infrastructure, including the TENA Middleware, the TENA Repository, and the TENA Data Archive [4]. The TENA Repository is responsible for the storage and management of range resources, and the TENA Data Archive mainly realizes the storage and management of data associated to operation. In HIT-TENA, we also designed the Data Archive, to store and manage the data generated in different phases of simulation test.

This paper will briefly describe the HIT-TENA architecture, and mainly introduce the construction of HIT-TENA Data Archive.

2. **HIT-TENA Architecture.** An overview diagram of HIT-TENA is shown in Figure 1. HIT-TENA mainly consists of four basic categories of software:

2.1. **HIT-TENA Resource Application:** HIT-TENA test system is usually built in specific test tasks, using all sorts of HIT-TENA Auxiliary Tool on the basis of common infrastructure. HIT-TENA Resource Application is the basic unit of the test system. Resource application is built by the test system developer and configurated to each node to execute all important tasks in process of the test.

2.2. **HIT-TENA Common Infrastructure:** HIT-TENA Common Infrastructure means software sub-systems to achieve rapid construction of basic software in HIT-TENA applications. These include HIT-TENA Middleware, for real-time information exchange; HIT-TENA Resource Repository , as a means for storing resources, such as component model, object model and other information; and HIT-TENA Data Archive, for the storage of scenario data, data collected during the test, and so on[3].

2.3. **HIT-TENA Utilities:** HIT-TENA Utilities is designed for the management of HIT-TENA Common Infrastructure and basic resources, including Resource Encapsulation Tool, Resource Repository Manager, Data Archive Manager, HIT-TENA Gateway and Data Collectors. Resource Repository Manager provides the configuration management and security of the HIT-TENA Resource Repository. Data Archive Manager assists the user in managing and ensuring the consistency of the diverse distributed data in HIT-TENA Data Archive. Data Collectors record public LROM information, and collect data during a test. Resource Encapsulation Tool achieves the encapsulation of all kinds of resources, later the resources will be interoperability, reuse, and composability. HIT-TENA Gateways are bridges between other isomerous systems and HIT-TENA system [3].

2.4. **HIT-TENA Auxiliary Tool:** HIT-TENA Auxiliary Tool is designed for HIT-TENA Application, to achieve rapid construction of application with HIT-TENA tools, including Resource Application Integrated Development Environment, Integrated Display Software, Data Analysis and Processing Software and Integrated Environment Supporting Software. Integrated Environment Supporting Software provides different information of environments. Data Analysis and Processing Software provides expended application about data analysis and processing model of some software in HIT-TENA, such as MATLAB and SIMULINK.

2.5. **Non-TENA Applications:** Systems those are not built in accordance with HIT-TENA, but needed in a logical range [3].

As we can see from Figure 1, this paper will describe the construction of resource repository in HIT-TENA Common Infrastructure, and the design of resource repository manager in HIT-TENA Auxiliary Tool.

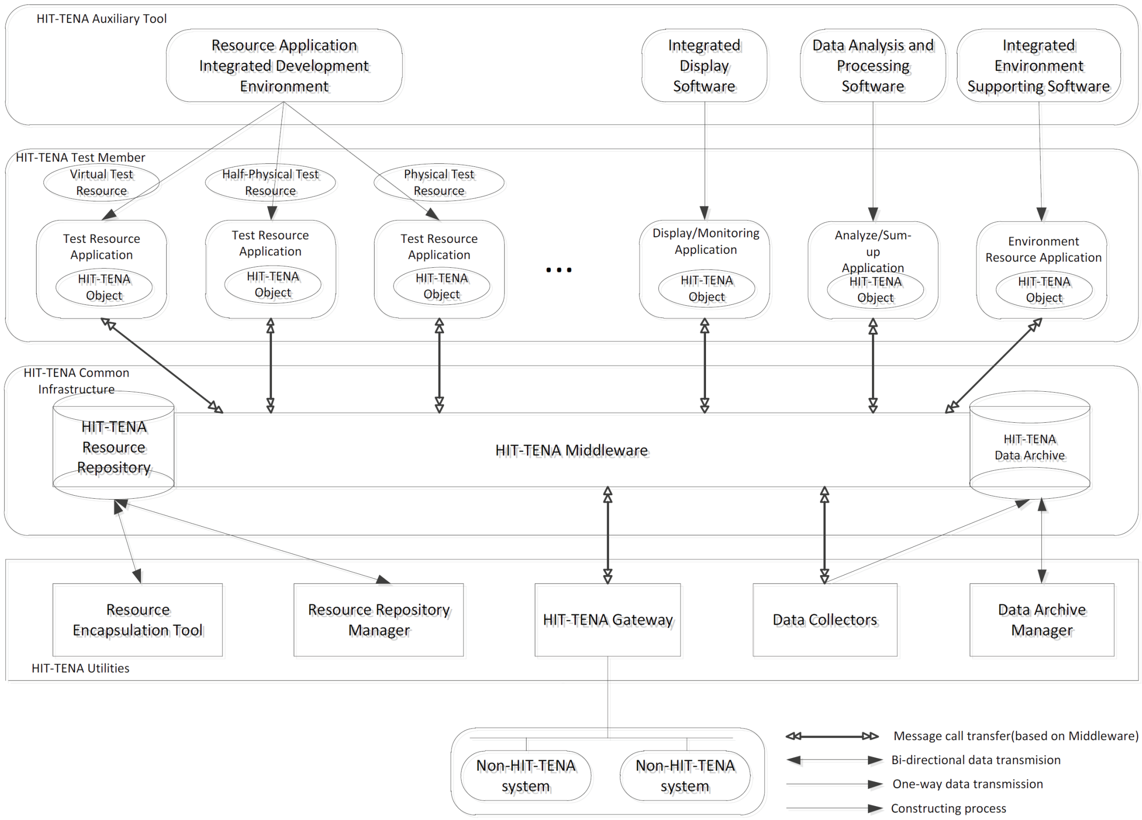

FIGURE 1. An overview of HIT-TENA

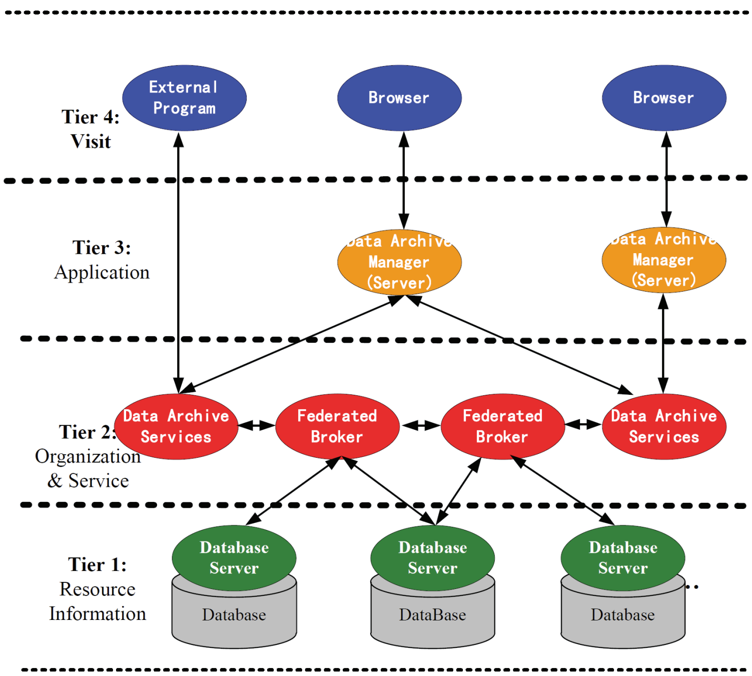

FIGURE 2. A four-tiered HIT-TENA Data Archive

3. **Organization of HIT-TENA Data Archive.** HIT-TENA Data Archive is a high-performance, distributed and timing sequence organized database distributed on multiple computers. It supports real-time query and provides the necessary data to keep the normal operation of test systems, mainly includes the Data Archive Manager and the Data Archive Web Service.

A four-tiered structure adopted in HIT-TENA Data Archive as showed in Figure 2.

The first tier contains the raw data (test schemes and test data) in databases. Each database has a database server that serves the raw information in whatever form is best for that information. A database server could serve a relational database, a multi-media database, an object-oriented database, or any other persistent storage mechanism.

The second tier provides for unification of the schemas of this information, and is mainly responsible for establishing "joint multi-database". The Federated Broker components share the original data served in tier 1 through mutual communication. The Data Archive Services components are distributed, and provide needed global services for other software (Data Collectors, Middleware, etc.) to access the information in HIT-TENA Data Archive.

The third tier provides users information. It's responsible for the construction of Data Archive Manager through based on the Data Archive Services served in tier 2. And the web browsers can obtain information through the Data Archive Manager.

The fourth layer provides server-side program based on B/S (Browser/Server) for users to visit the base information in data archive through the web browsers.

4. **Design of databases in Data Archive.**

4.1. **Database Needs Analysis.** The main purpose of Data Archive is to store the data information generated in different phases of simulation test, including the test scheme information planed before the test, the data gathered during the test and analysis and the summary after the test. What's more, the Data Archive should enable users to manage the information and users. Specific data fields and structures are listed as follows:

Test scheme information: scheme information of the test system designed by the test personnel through Resource Application Integrated Development Environment, including scheme identification, scheme name, created time, create units, unit IP, and LROM file and scheme description file.

Test information: to record the relevant information about each test, including test identification, test time, test units, test personnel, remarks, etc.

Test data information: data gathered during the test and test analysis and summary after test, including test identification, scheme name, file name, file type, metadata file, remarks, etc.

User information: to be responsible for the storage of all users, including user name, user password, user unit, user roles, etc.

User roles information: such as role identification, role name.

Log information: to store the run-time errors of Data Archive, including log identification, error message, etc.

4.2. **Design of Database Conception Structure.** We use the E-R (Entity-Relationship Approach) diagram to express the data fields and structures. Figure 3 reveals the E-R diagram of data archive.

As the E-R diagram shows, each user owns a role, and different roles have different permissions, every role can own multiple users. The users could manage the test scheme. Each test scheme corresponds to many tests, and each test corresponds to a lot of test

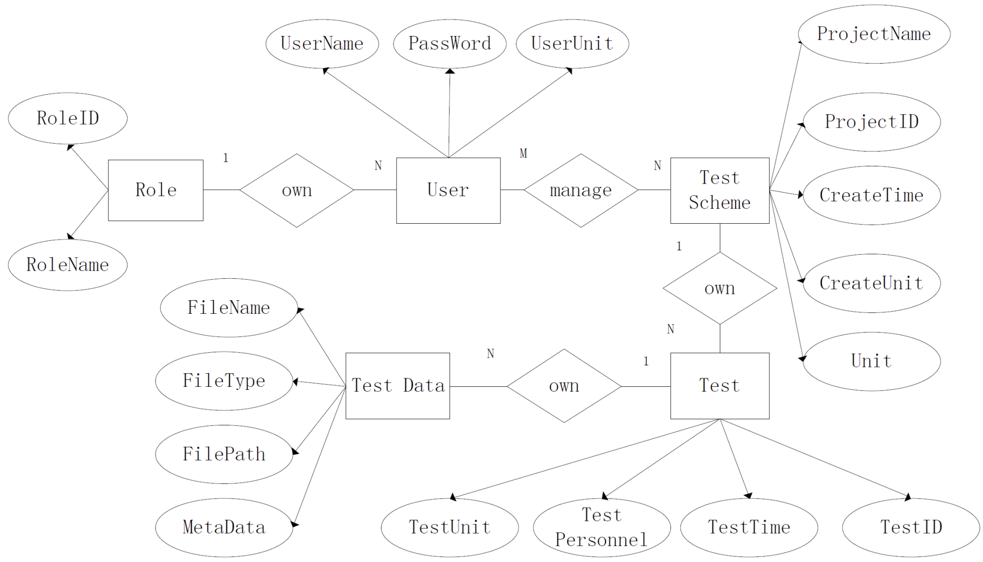

FIGURE 3. The E-R Diagram of Data Archive

data. In addition, every entity such as user, test scheme, and so on, also has its own properties.

4.3. **Design of Database Physical Structure.** By transforming every entity in the E-R model into relation model to complete the design of logical structure, properties of original entity become the properties of the new relation model, and so do the primary keys [7]. The next step after the transform is to design the physical structure (design of database tables). Figure 4 gives a final united view of database tables.

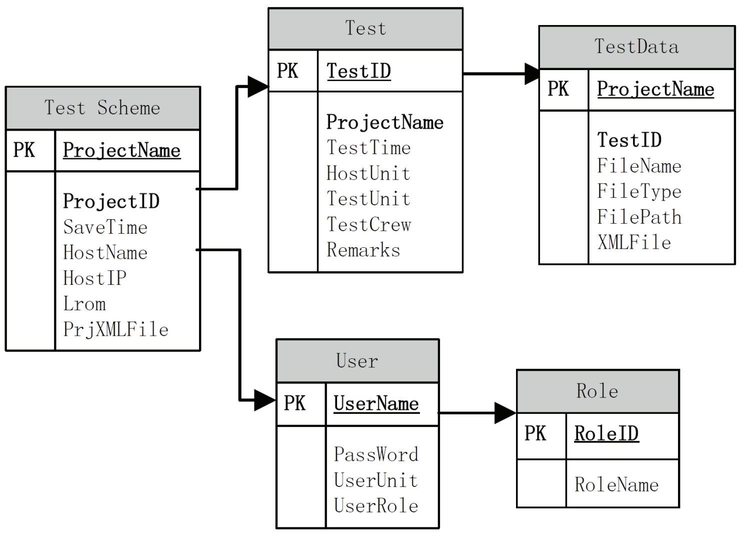

FIGURE 4. Relationship between the database tables

4.4. **Design of the stored procedure and trigger.** A stored procedure is a set of pre-compiled T-SQL codes, stored in database server to improve the performance and consistency of the repetitive tasks [8]. When it comes to execute, they don't have to be compiled again, so operating efficiency of the program is improved. The stored procedure can accept parameters and return multiple parameters in the form of the output parameters to the process or batch processing those call stored procedures. Using stored procedure has many advantages, such as fast speed of execution, the modular programming and the reducing of network communication. The stored procedure can be called by the application program as a unit. Data Archive contains a lot of repetitive operations such as add, delete, update, and so on, so we must design the stored procedure to improve the operation efficiency.

A trigger is a special type of stored procedure defined on a particular table. The trigger will automatically occur when operations affect the data protected by the trigger [8]. When a DELETE trigger is triggered, the deleted records are stored in a special deleted table. Modify a record is equal to insert a new record and delete an old record. UPDATA statement can be considered as a DELETE statement that deletes a record and a INSERT statement that inserts a record. And trigger may be nested. In Data Archive, test scheme, test and test data are all related, if a test scheme was deleted, all tests belong to it, as well as the test data belong to the tests should be removed, otherwise it will lead to the inconsistency of information. Therefore, we must set triggers among the associated tables to cascade update related fields and cascade delete related fields.

5. **Design of Data Archive Manager.** HIT-TENA Data Archive Manager is responsible for the management and control of HIT-TENA Data Archive. It provides graphical interface for users to interact with Data Archive. By Data Archive Manager, users can view, backup, restore, delete and manage all information in data archive, including test scheme information, test data and metadata information.

HIT-TENA Data Archive Manager adopts three-tiered architecture based on B/S and presents users information in the form of Web pages. Figure 5 shows the three-tiered architecture that based on B/S, including data tier, logic tier and presentation tier. Data tier is responsible for the storage of all data; Logic tier is responsible for the management of database and service; Presentation tier provides visual interface for management.

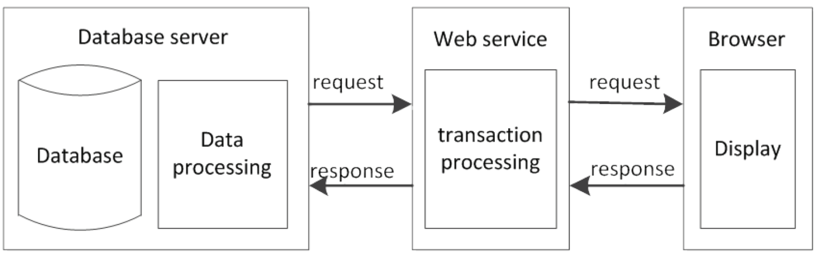

Figure 5. Three-tiered architecture base on B/S

HIT-TENA Data Archive Manager consists of several functional modules, as shown in Figure 6. Users can store the test scheme and data by upload module and browse them through the retrieval module. Besides, management module provides users functions to update them. Before that, users can pro-process data to improve data quality according to individual demand. Database management module is to ensure the safety of the databases. Users can backup or restore databases when necessary. When a runtime error occurs, log management module will record it for developers to study and modify. Not everyone can visit the Data Archive Manager the user management module is designed over whole distributed system to manage the users' grants. This paper mainly introduces the upload and management of the test scheme and data in the next moment

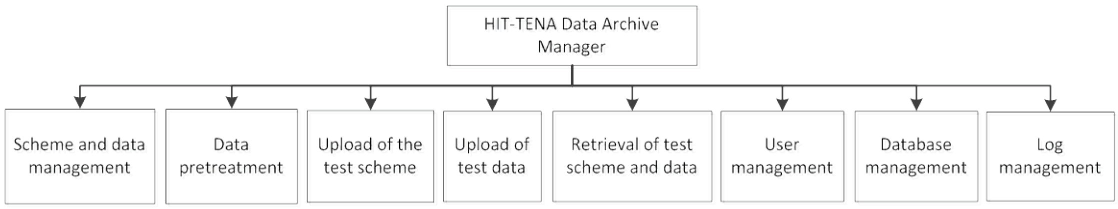

FIGURE 6. Main functional modules of HIT-TENA Data Archive Manager

5.1. **Upload of the Test Scheme and Data.** Test data are composed by the test metadata and test data. The test metadata are used to define the data format of the test data, and test data refer to all test information gathered during the test. For Data Collected, there are two methods to store, in the form of document or database tables. The Data Collectors would generate test metadata before storing the test data, similarly, the test metadata are stored in two forms, documents and database tables.

Therefore, the upload module should take the above two cases into consideration, allow users to upload in the form of document or tables. The former refers to save the document to the databases, and the latter is to copy the test data tables to data archive databases from original database, save the related test data information at the same time.

5.2. **the Management of the users.** There are three kinds of roles in this system correspond to users' authority, senior manager, general manager and visitor. This paper adopts the RBAC (Role Based Access Control) model to realize the authority management, which is based on "user-role-permission". Permissions are assigned to roles, not users, endue user definite roles according to their duty, finally users can get their permissions through their roles [9] Figure 7 shows the method of RBAC model. A user can own more than one role and a role can have more than one permission.In order to achieve the separation of the users and permissions, the users' permissions are associated with the roles, and the roles are associated with the users, then it is easy to manage the users' permissions.

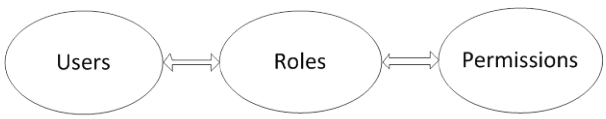

FIGURE 7. RBAC model

When administrators add a new user, they must set the user a role. As a result, the user will own corresponding permissions. Figure 8 shows the different permissions of different roles.

Compared to visitors, general managers have the right to manage the information of users. Senior managers can manage the database and log besides this. Visitors only can modify their personal information, upload and manage test scheme and data.

5.3. **Pretreatment of data.** In order to ensure the integrity, accuracy and consistency of the data, it is necessary to process data before storing the data. The data preprocessing can define as the operations as followings: data cleaning, data integration, data conversion, data reduction. The Data Archive Manager realizes data processing by organizing SQL statement to manipulate database content according to the users' settings.

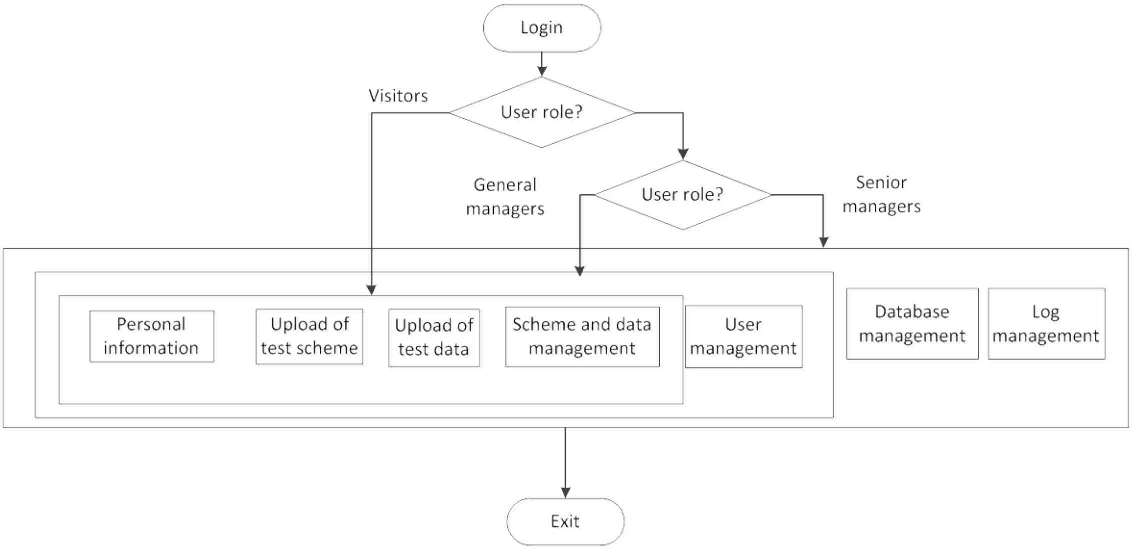

FIGURE 8. The different permissions of different roles

6. **Design of Data Archive Web Services.** As we can see from the HIT-TENA architecture, Data Archive needs to exchange information with various tools, such as the Resource Application Integrated Development Environment needs the test scheme access service to complete the reading and storage of test schemes. The Data Analysis and Processing Software calls the test data query service to get test data information, and saves processed results through test data storage service. Data Collectors need the test data storage service to save the test information and playback the test data after obtaining data through test data query service. Therefore, it is necessary for HIT-TENA Data Archive to serve appropriate interfaces for other tools.

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards [10]. The data format of Web Service is XML, which has the advantages of platform independence, so the Web Service's work processes do not need to consider the system and platform of the client and server [11] The design of operation functions in the interface is the most important part of Web service. Classify the operation function in accordance with their function. Table 1 shows the classification.

7. **Conclusions.** This paper proposes a method to manage data generated in various stages of the tests through Data Archive in virtual test architecture. Therefore, this paper focuses on the construction of HIT-TENA Data Archive, including the design of databases, the design of Data Archive Manager and the design of Data Archive Web Service. Eventually a real-time and unified storage and management of test data came true to ensure the share and consistent of the test information. HIT-TENA Data Archive laid the foundation for the normal operation of the HIT-TENA platform.

Acknowledgement: This work is supported by National Science Foundation of China under Grant No. 61201305, Heilongjiang Provincial Postdoctoral Foundation of China

TABLE 1. Classification of the functions in web service

Classification	*function*	*input*	*return*	*Function Name*
Judge (To judge whether the specified item is in database)	Query database to determine whether the project has been in existence	Project Name	BOOL (exist or not)	IsProjectExist
	Query database to determine whether the test data has been in existence	Test ID		IsTestExist
	Query database to determine whether the file has been in existence	File Name, Project Name		IsTestFileExist
	Query database to judge whether the user has been registered	User Name		IsUserExist
Obtain (To obtain specific item from database)	Download specific project file from database	Project Name	Document binary flow (XML file of selected project)	GetXMLofProject
	Download specific LROM file from database	Project Name	Document binary flow (LROM file of selected project)	GetLROMFile
	Get all projects' basic information from database	None	Project ID and name of all projects	GetIDofProject
	Get the number of all projects in database	None	Number of all projects in database	GetNumofProject
	Get the number of all tests in database	Project Name	Number of tests in a project	GetNumofTest
	Get the number of all test files in database	Project Name, Test ID	Number of test file in a test	GetNumofTestFile
	Query database to get the basic information of selected test	Project Name, Test ID	Basic information of selected test	GetTestData
	Query database to get the basic information of all tests in a project	Project Name	Basic information of all tests in a project	GetTest
	Download selected test file from database	Project Name, Test ID, File Name	Document binary flow (selected test data)	GetOneTestFile
	Download the metadata of selected test data	File Name, Test ID, Project Name	Document binary flow (Meta data of selected test data)	GetMetaData
	Download all test data in a test in the form of zip file	Project Name, Test ID	Structure of document binary flow (all test data in a test)	GetAllTestFile
	Query database to get the basic information of selected project	Project Name	Basic information of selected project	GetProjectInfo
Store (To save specific item to database)	Add new test scheme to database	Information of test scheme	BOOL (success or not)	NewProject
	Store new test to database	Information of test		NewTest
	Store new test data to database	Information of test data		NewTestData

under Grant No. LBH-Z11170, and the Fundamental Research Funds for the Central Universities of China under Grant No.HIT.NSRIF.2012015.

REFERENCES

[1] R. M. Feng, G. Y. Wang, and K. d. Huang, Research on test and training enabling architecture (TENA), *Journal of System Simulation*, vol. 16, no. 10, pp. 2280-2281, 2004.

[2] R. S. Nattermann, and R. Anderl, Simulation data management approach for developing Adaptronic Systems - The W-model methodology, *Journal of World Academy of Science, Engineering and Technology*, vol. 51, pp. 429-435, 2011.

[3] R. Daehler-Wilking, S. Hunt, and B. Greenberg, The modeling and simulation catalog for discovery, knowledge and reuse, *Proc. of Fall Simulation Interoperability Workshop*, pp. 176-187, 2010.

[4] S. C. Gencer, B. P. Ketcherside, G. O. Morrell, and K. D. Wiegand, Data management in reservoir simulation, *Proc. of SPE Reservoir simulation Symposium*, pp. 159-167, 2007.

[5] V. P. Holmes, W. R. Johnson, and D. J. Miller, Integrating metadata tools with the data services archive to provide web-based management of large-scale scientific simulation data, *Proc. of The 37th Annual Symposium on Simulation*, pp. 72-79, 2004.

[6] L. M. Teng, R. Y. Liu, and N. Liu, Study on integrated management of marine remote sensing data, *Journal of Shanghai Jiaotong University*, vol.42, no. 10, pp. 1674-1677, 2008.

[7] Foundation Initiative 2010 Project Office, *The TENA architecture reference document*, available at https://www.tena-sda.org/public_docmanager/userdocuments/TENA%20ARCHITECTURE%20REFERENCE/TENA%20Architecture%20Reference%20Document%202002.pdf.

[8] P. Lan, Application of SQL server database trigger in data integrity and security, *Journal of Database and Information Management*, vol. 2, pp. 23-24, 2013.

[9] J. F. Wu, Z. M. Zhang, X. T. Tian, and L. J. Huang, An access control method based on improved RBAC model, *Journal of Computer Measurement & Control*, vol. 20, no. 8, pp. 2147-2148, 2012.

[10] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web services: concepts, architectures and applications*, Springer, Berlin-Heidelberg, Germany, 2010.

[11] M. Junghans, and S. Agarwal, Web service discovery based on unified view on functional and nonfunctional properties, *Proc. of The 2010 IEEE Fourth International Conference on Semantic Computing*, pp. 224-227, 2010.