Advances in Engineering Software 40 (2009) 623–629

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

A Problem Solving Environment Portal for Multidisciplinary Design Optimization

Ju-Hwan Kim, Ho-Jun Lee, Sang-Ho Kim, Jeong-Oog Lee *

Dept. of Aerospace Information Engineering, Konkuk University, 1 Hwayang-dong, Gwangjin-Gu, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 30 July 2008
Received in revised form 14 October 2008
Accepted 24 November 2008
Available online 6 January 2009

Keywords:
Problem Solving Environment Portal
Multidisciplinary Design Optimization
Globus toolkit
Web services

ABSTRACT

Multidisciplinary Design Optimization (MDO) is a design methodology that derives optimal design solutions by concurrently considering various mutually dependent design elements from an assortment of disciplines. As such, it is applicable to the designing of ships and automobiles, as well as to aero vehicles. However, applying MDO methodologies in the real world would require a designer to spend an enormous amount of time arranging and integrating resources used in the process. This paper proposes a Problem Solving Environment (PSE) Portal for MDO methodologies, providing an environment that enables designers to utilize design resources conveniently even without working knowledge of the systems. Furthermore, the PSE portal yields an optimal MDO environment by allowing for global collaborative sites, which securely share design resources, and by offering users an efficient interface.

1. Introduction

Since each system in an engineering design consists of multiple disciplines linked together, a design process should consider all of these systems concurrently. In addition, to optimize the design given the resources and time at hand, design optimization should take into account all relevant disciplines simultaneously.

Multidisciplinary Design Optimization (MDO), which simultaneously takes into account mutually dependent design elements from various fields, has the advantages of reduced time and cost compared to serial design approaches. MDO was originally devised and developed for designing aircrafts, though it has recently been applied to diverse design applications, including ship building and automobile engineering [1]. Applying MDO methodologies to actual design problems requires the organization of various analysis tools and other resources, optimization aids, and Computer Aided Design (CAD) software. Considerable time is required for the appropriate integration of these elements. Therefore, an integrated MDO environment is necessary to conduct such processes more conveniently and efficiently.

The software for an integrated MDO environment should be capable of integrating the resources used in the design process. It should efficiently link resources while providing an easy user interface that can help users utilize the resources even without a working knowledge of the system. In addition, as the research environment becomes universal, global collaboration and sharing of design resources should be made possible.

In response to these requirements, this paper proposes the formulation of a Problem Solving Environment Portal (PSE) for MDO which provides users with a convenient and efficient way to utilize design resources and services. The PSE portal introduced in this paper comprises transparent and efficient web interfaces which allow users to solve specific problems without much difficulty [2]. Using web browsers, the PSE portal offers a collaborative environment for designers scattered over remote locations. It also allows integration of design resources via the Internet. Finally, it ensures fast and reliable transmission of large amounts of data using web service-based GT4, a standard technology for distributed middleware.

This paper is organized as follows. Section 2 introduces the fundamental concepts of MDO, GT4 and PSE, while Section 3 describes the architecture of the PSE portal for MDO. The implementation and application of the PSE portal is discussed in Section 4. Comparison with other works is presented in Section 5, followed by the conclusion in Section 6.

2. Fundamental concepts

2.1. Multidisciplinary Design Optimization (MDO)

Currently, product design and development in individual fields requires collaboration and specialization. Serial design systems waste time and money, especially when applied to aircraft or ship building industries that deal with massive amounts of data.

As an optimization methodology working on an integrated environment, MDO accounts for design elements from various engineering fields through integration, automation and optimization. It has been studied primarily for aerospace systems development, though it has also been employed for efficient product

* Corresponding author. Tel.: +82 2 450 3435; fax: +82 2 444 6670.
E-mail address: ljo7@konkuk.ac.kr (J.-O. Lee).

0965-9978/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2008.11.006

design in other diverse applications, including mechanical and electrical & electronic engineering fields. In addition, techniques such as system identification and multistage fragmentation have been studied in pursuit of efficient and fast optimization. The goal is to achieve a completely integrated optimal design, along with design cycle and cost reduction, by extending communication between experts of the design.

These MDO techniques require an integrated optimal design environment capable of dealing with extensive calculations. This environment must combine the analysis and optimization tools developed in various languages on various platforms in a convenient and efficient way. The PSE portal that this paper proposes enables easy construction of such a design environment.

2.2. Globus toolkit ver.4 (GT4)

The Globus toolkit is a series of libraries and programs that handle the general problems occurring in developing grids and distributed systems [3]. The three containers that constitute GT4, i.e. the Java container, the C container, and the Python container, use services developed by the client in Java, C, and Python, respectively. They run the security system, retrieval, and state management when a service is established. The individual resources that compose the system are developed on the containers provided by GT4. They take the form of web services using JAVA, with XML as a standard protocol technique. This allows the portal to be independent of the platform and the programming language. Another advantage is that by using Hyper Text Transfer Protocol (HTTP), existing technologies for proxies and firewall can be used without modification.

GT4 is based on five components: security, data management, execution management, information services and common runtime, and web service. It combines resources, constructs the Virtual Organization (VO), and provides services required for grid and distributed computing. These services include grid security, information service, execution management, and data management [4]. GT4 also provides a web-based MDS (Monitoring and Discovery System) function that allows monitoring of the status of resources, including Grid Resource Allocation and Management (GRAM) service for execution management. Grid Security Infrastructure (GSI) is implemented by the Public Key Infrastructure (PKI) technique to meet the security requirements.

2.3. Problem Solving Environment (PSE)

A PSE should be able to solve extremely complex mathematical models, manage massive quantities of data, use high-performance resources, construct a collaborative environment for experts, and utilize remotely scattered resources. The characteristics required for a PSE are detailed as follows [5]:

- Problem-oriented: a PSE should be a convenient portal for clients to use the system, even if they have no knowledge of the details of the system's functioning.
- Distributed integrated: since a single computing system may be insufficient for complex operations and massive data processing, resources should comprise a distributed environment which should, furthermore, be concatenated for effective management.
- Collaborative: a PSE should allow the construction of a VO and provide design experts with an intuitive and convenient user interface for a collaborative environment.
- Persistent: operations that are complex and time-consuming should maintain operative status at all times. Data for problem solving, instead of being disposed of, should be constantly managed and preserved and should be retrievable from the database without any time constraint.
- Open flexible adaptive: the PSE portal should be a flexible and open environment that can add or remove resources according to demands, rather than just integrating specified resources.
- Security: security is a crucial issue for a distributed environment based on web services. The system should be open only to classified users who have permission to use it. In addition, data transmission should be reliable and stable.

PSE portals are being widely studied for their problem-solving applications in various science and engineering fields. Prominent examples include the BioGrid Application Toolkit [6] in the biomedical field and Medical Image Analysis [7] in the medical field. Other applications include the e-Science Aerospace Integrated Research System (e-AIRS) [8] for flow field computational analysis and remote wind tunnel test services, Grid-Enabled Computational Electromagnetics (GECEM) [9] for integrated computational simulation and visualization, GEODISE (one of Britain's pilot projects) [10], and the Distributed Aircraft Maintenance Environment (DAME) [11] for the remote monitoring of aircraft engine status.

Incorporating research in bioinformatics, biomechanics, and biosignal processing introduces certain requirements to biomedical data analysis and engineering. Specifically, they necessitate parallelized, distributed computing algorithms and high-throughput computing resources in the area of data analysis and management. The BioGrid Application Toolkit by PSE, developed to meet these requirements, provides a high-performance web-based user interface for biomedical data analysis. The logical architecture of the system consists of the User Services Layer (USL), the Business Logic Layer (BLL), the Data Access Layer (DAL), the database, the grid services, the security, and communication layers. Since it was written in JAVA, the USL is platform-independent and the user is able to access each application via the USL. The BLL performs business tasks consisting of one or more steps, while the DAL manages access to stored data during business processes. The database stores information regarding users and projects. The grid service ensures transparent access to distributed resources. The security layer conducts authentication, secure communication, and authorization, and the communication layer defines the means of communications between the components of the application.

Medical Image Analysis (MIA) requires implementation on a heterogeneous environment. MIA applications can be used for clinical trials that involve a lifecycle, i.e. development, optimization, evaluation, and deployment. The processes that pass through the MIA lifecycle include tasks that complement each other and run on the heterogeneous environment. The PSE for MIA applications exchanges information among MIA lifecycle components and provides a flexible mechanism. Its elements include the resources that are run and distributed on the heterogeneous environment. Its architecture shows the flow of information at each step of the MIA lifecycle. The architecture consists of three layers: the user interface, the web-service and database, and a legacy system. The web-service layer comprises Application Description Management (ADM), Data Provenance Management (DPM), Project Management (PM), and Application Execution Management (AEM). ADM describes, stores, accesses, and retrieves for MIA applications, while the information needed between steps is stored through AEM. DPM allows previously used data to be searched. PM creates the set of information which makes up the project and provides access to it. Services are called from the legacy system.

3. Architecture of the PSE portal for MDO

The PSE portal for MDO should be capable of both massive data transmission and management of the resources that constitute the system. In addition, design experts require an environment they can use more conveniently without knowing the details of the system functions or where their information is protected. Fig. 1 represents the architecture of the PSE portal proposed by this paper in response to these requirements. The architecture includes the user interface, problem description management, security management, workflow management, resource management, and data management, allowing for user convenience and intuition. Since experts in each discipline are not concerned with the working knowledge of systems outside their own fields, there is a demand for an environment in which the system can be used without detailed knowledge of its functions. Workflow management and data management fulfill these requirements. The workflow is defined in the Workflow Management Coalition (WfMC) as a whole or a part of the automated business process in which a document of information or work is transferred from one participant to another, following a series of procedures [12]. Despite the complexity of the process in a PSE portal for MDO, deleting, adding to, and editing of the process should be possible during its execution. Workflow management service allows the designer to easily use the environment and efficiently utilize the scattered resources. This is necessary because an engineering designer would want to get results by defining a simple process rather than needing to understand the flow of the internal functions. The user can define a process with the help of the process definition service and directly edit, store and correct the process. The defined process is transformed into XML format and becomes ready to execute. Once the defined process is transmitted, the execution service executes the scattered resources according to the situation, via the Ws-GRAM service that GT4 provides. Both serial and parallel processes are possible. Messaging with the resources utilizes SOAP, a standard web service technique.

Data management takes charge of the list of input and output data and the link configuration of the data. The resource management service provides the design object, for which the requirement analysis is done with suitable resources, and uses the resource monitoring service to confirm the availability of the resources to the user. It also manages the resources. In order to execute the resources, files should be transformed into a form used by each resource. This is done by the agent management service that performs wrapping and parsing of the files. In other words, the input and output files for a resource are transformed by wrapping and parsing into a form that the resource requires. Then, the transformed input/output data are linked to each other according to the data relation configuration.

MDO aims for an integrated optimal design by relating various design disciplines and stimulating participation of experts in the disciplines. Meanwhile, the user needs a collaborative environment which is free of spatial restrictions. The user interface consists of functions such as user authentication, requirement analysis, problem formulation, input/output for analysis, optimization, and configuration. Using these functions, authenticated users can get results ranging from the problem formulation to verification of the final configuration.

The problem description management service consists of project and task management. A project refers to a management unit involving multiple designs, whereas a task entails a single design. The user selects an analysis module, which may or may not be optimized, and decides on the parameter to optimize cost or size.

To integrate spatially scattered design resources and disciplines for a design, it is necessary to share resources. This raises associated security issues, which should be addressed. Security management is divided into user-level security and system-level security and is handled by the security management service. The user-level management administers authentication for the users of the system. Only authenticated users are allowed to access to the system. The system-level management uses Certificate Authority (CA) to take charge of the security of the entire system.

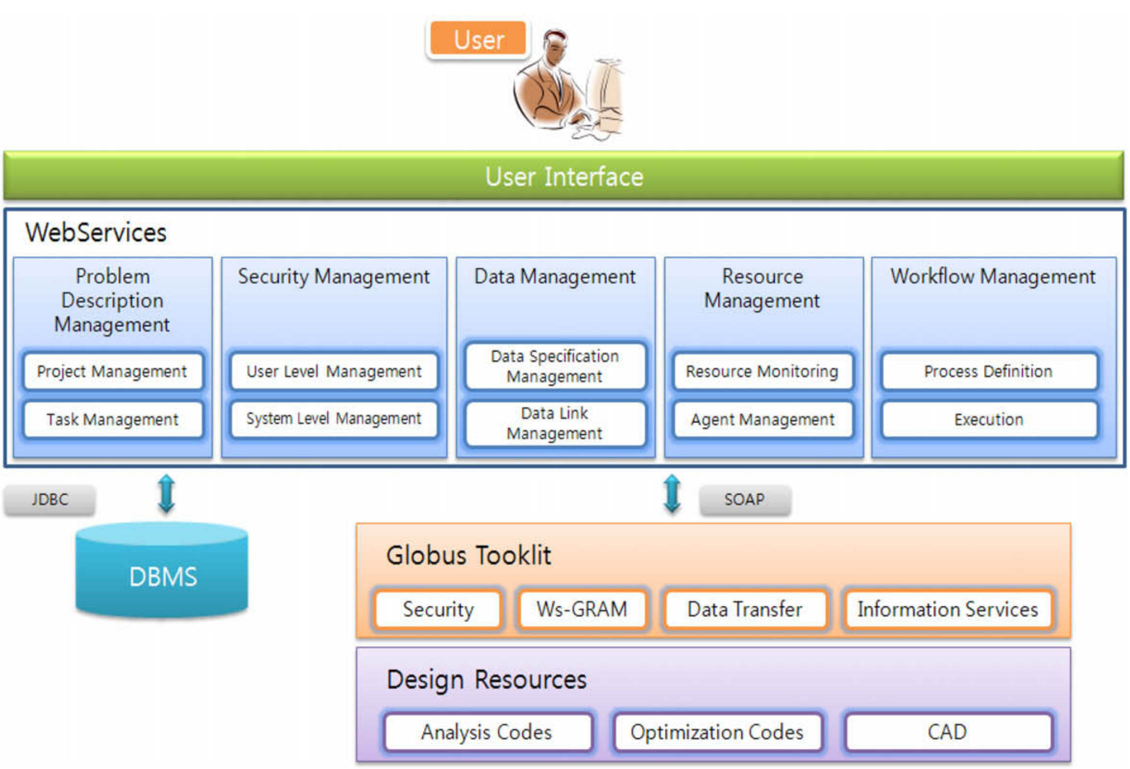

Fig. 1. The architecture of the PSE portal for MDO.

Design resources are spatially scattered; therefore, reusable and continuous control is necessary for the analysis, optimization, and final configuration of the data. Data is managed by the DBMS. For engineering designs, a DBMS should be centralized and allow users to retrieve required data on time. A DBMS also stores data for the analysis and optimization of the engineering design and manages data input and output for the system. In addition to easy information retrieval for projects and tasks, the system should be able to reuse existing data. It should also meet the requirements of data integrity, referential integrity, data security, extendibility, and freeness from redundancy. Data is protected by restricting designer access to only pertinent data, not the entire design.

The proposed PSE portal for MDO meets all of the requirements addressed in Section 2.3. It employs a problem-oriented workflow, data management, collaborative UI, project description management for integration, resource management, security management, and DBMS for maintenance. By using standard techniques like GT4, the architecture is able to add and delete resources and is easily adaptable to the system.

4. Implementation and application of the PSE portal

Aero vehicles are classified into a variety of configurations, such as passenger airplanes, fighters, Very Light Jets (VLJ), missiles and Unmanned Aerial Vehicles (UAV), etc. A designer who wants to design an aero vehicle should compose an MDO framework on the PSE portal. The proper MDO framework depends on the design target. Once the MDO framework is constructed, the designer is ready to begin a substantial MDO-based design. The designer may create a new project using the Project Management service or construct a proper task after opening an existing project. A task is constructed in the project, and projects and tasks can be stored for the future use.

Fig. 2 illustrates a process flow for constructing an MDO framework on the PSE portal. First, the designer defines a target aero vehicle and analyzes its requirements. Next, the designer uses the Resource Management service to search for and select resources such as analysis codes, optimization codes, and CAD tools from the resource repository. The resource repository uses specific rules to store resources registered by individuals or agents. Once the resources are selected, the designer defines the design process with the Workflow Management service. The Data Link Management service is then used to define relationships between the resources. The system automatically creates agents for each resource and DB tables for storing the input and output data. This feature provides the designer with a more convenient MDO optimization environment.

The Agent Management service wraps and parses the input and output data in the way that the resource requires. Meanwhile, the Workflow Management service helps the designer easily define the design process and executes the distributed design resources according to the defined process.

The procedure for constructing the MDO environment to design a cruise missile is introduced here as an example of application of the PSE portal. First, in the Project step, the designer should compose a design environment by selecting the required design resources. Fig. 3 illustrates the concept of the resources and the Multidisciplinary Feasible (MDF) method composed for the MDO environment of the cruise missile.

Five tools are selected as analysis resources: Digital DATCOM [13] for aerodynamic analysis, an in-house engine sizing code from Konkuk university, a weight analysis code, a performance analysis code, and a mission analysis code [14]. GEnetic algorithm for Numerical Optimization for COnstrained Problems(GENOCOPβ) [15] is used for the optimization, and the geometric results are created by an in-house automated parametric geometry generator [16]. In addition, the MDF method is employed to integrate and optimize the design variables of these resources. The MDF is known to be an optimization method that considers the integrated design resources as a single analysis tool [17].

After selecting the analysis resources, the designer should configure the order of execution and the relationship between input and output data. Fig. 4 shows a screen shot of the workflow process definition for the cruise missile design with selected resources. The weight analysis code and the aerodynamic analysis code are exhibited in Fig. 4, together with their input and output lists and the connections in between. The system automatically creates the analysis agents. These agents conduct the pre-processing and post-processing that stores input and output data in the database and reads data back from the database to generate input files.

The PSE portal for MDO is made of HTML, JSP, and Javascript, and it synchronizes with the database through the JDBC driver. Fig. 5 is a screen shot that shows the results of the integrated analysis. The interface was designed so that the user can find useful information such as input and output data for each analysis tool, project information, and task information all at once.

The optimization problem requires the designer to go through a series of steps. These include using the Process Definition service to define the design process according to the flow shown in Fig. 3, setting the objective functions and defining the proper design variables, and others. Once the designer constructs an optimization problem, the Workflow Engine of the system iterates the specified process and finds the optimal results. Next, the Agent Management service generates input files for geometric design. Finally, the auto-

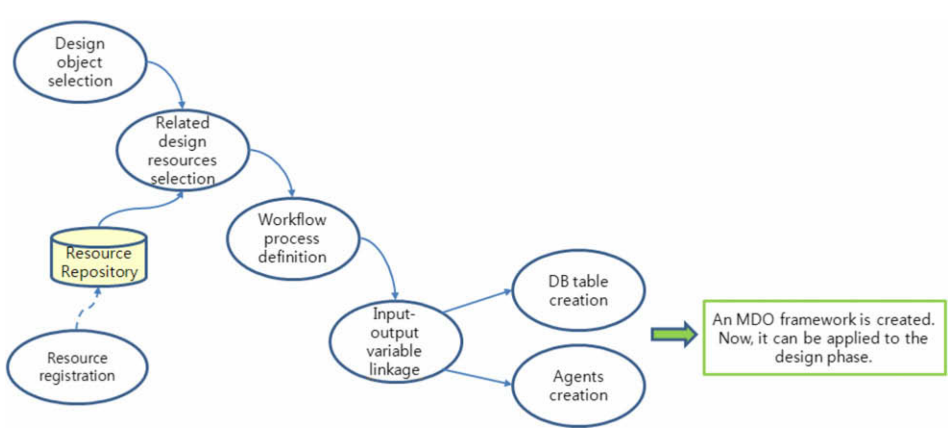

Fig. 2. Process flow for developing an MDO framework.

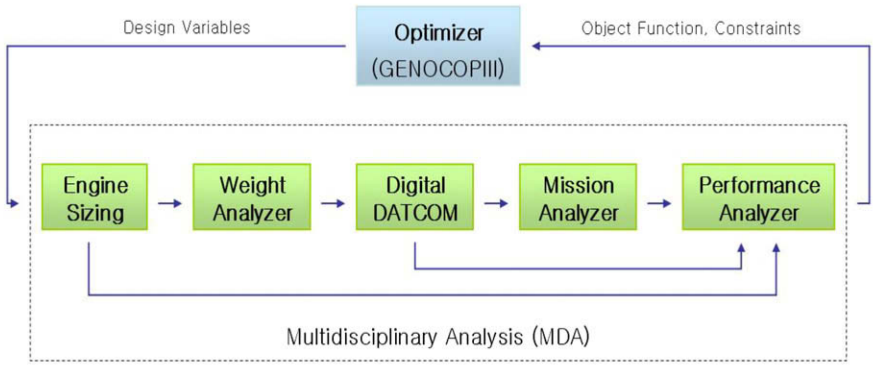

Fig. 3. The design resources and the MDF method for missile design.

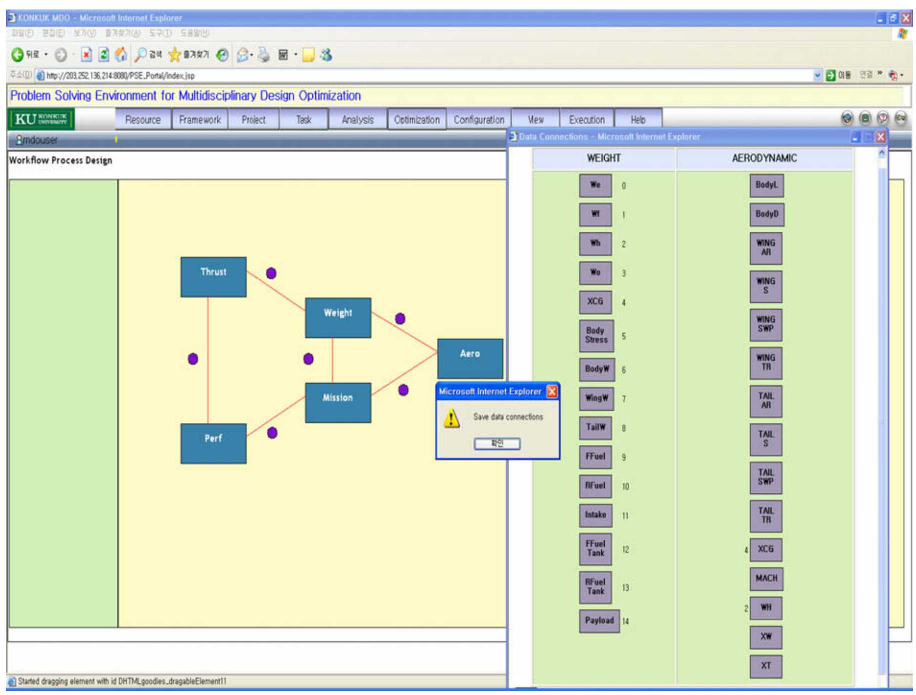

Fig. 4. Data connections.

matic geometry generator creates CAD files from the input data. This allows the designer to verify the optimized geometry and use it for preliminary and detail designs.

5. Comparison with other work

The current market for commercial Process Integration and Design Optimization (PIDO) software is driven by ModelCenter [18] from Phoenix Integration Inc. and iSIGHT products [19] from Engineous Software Inc. Multiple Analysis Servers on the network can be connected to a ModelCenter. Although ModelCenter works on Windows O/S, it can construct heterogeneous distributed environments by using Analysis Server, which controls various programs such as commercial CAD tools and engineering analysis tools. In addition, it supports Jscript and VBScript to connect new entities. This feature makes ModelCenter very flexible in cooperating with Internet/Intranet environments and databases. ModelCenter is equipped with DOT, a gradient-based method, as an optimization tool. It provides robust design, reliability design, and 6-sigma design for the optimization design methodology. It also comprises various design tools and is capable of integrating a number of CAD/CAE/PDM tools. If it is used in conjunction with CAD tools, the user can conduct a parametric geometry design on the specified parameters.

iSIGHT originated from the computing system for Computer Aided Optimization (CAO) of General Electric (GE) and is currently the most popular tool for product design. It employs MDO Language (MDOL), a unique operation script language, to provide a customized environment depending on the architecture of the problem and the user's circumstances. iSIGHT can be used in conjunction with CAD/CAM/CAE/PDM tools, whether the platform is Windows, UNIX, or LINUX. The Task Manager allows various optimization methodologies, including gradient methods, genetic algorithms, approximate methodologies, and quality engineering methodologies. It is capable of parallel processing of the multi-disciplinary analysis. It also provides features that define design process, data flow, and various design strategies.

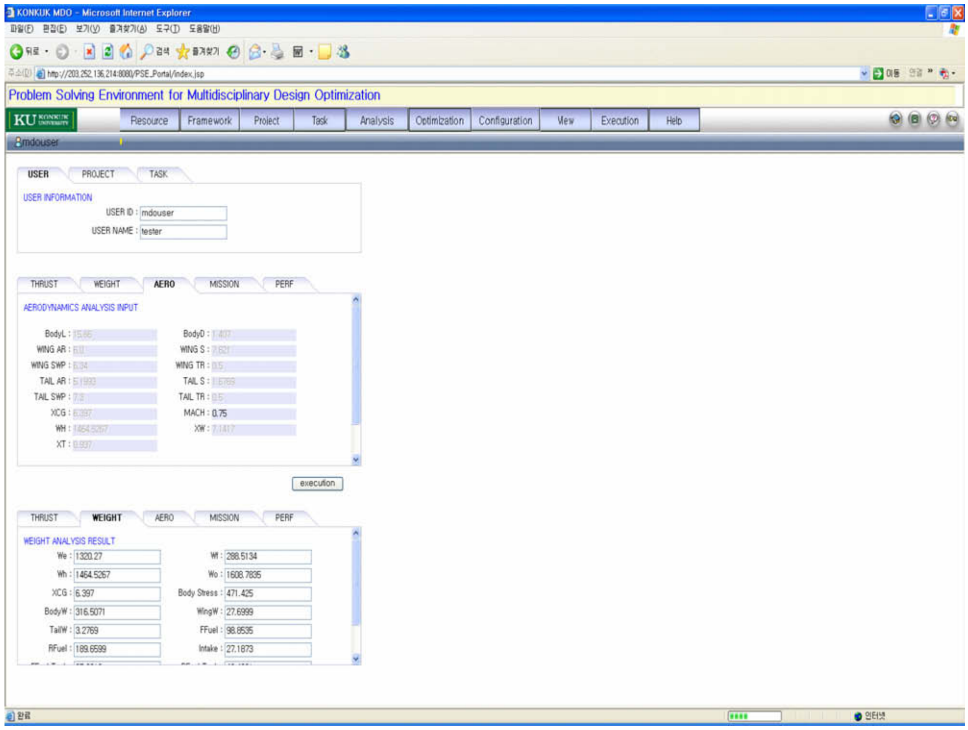

Fig. 5. The result of integrated analysis.

A couple of representative works for the MDO framework are Framework for Interdisciplinary Design and Optimization (FIDO) [20] from NASA Langley and Intelligent Multidisciplinary Design and Optimization (IMAGE) [21] from Georgia Tech Aircraft System Design Lab (ASDL).

Despite all the features and advantages of PIDO tools such as ModelCenter and iSIGHT products, they are not the perfect solution for the MDO environment. The PIDO software focuses on optimization methods at the expense of data management and collaboration between users. In addition, although it can handle engineering software such as CAD/CAE, it does not provide a satisfactorily convenient environment for users.

This paper proposes a PSE portal for MDO that employs GT4 components to provide Virtual Organization (VO) management, which is essential for global collaboration. In order to efficiently utilize scattered design resources, the PSE portal adopts a workflow management system. In addition, it is independent of platform and very flexible, and the resources can be shared conveniently because it is based on standard web technologies. Finally, the PSE portal combines analysis codes, optimization codes, and a Data Base Management System (DBMS) which stores and manages data. This makes the interface convenient and intuitively understandable, allowing users to efficiently utilize its features.

6. Conclusions

Because all disciplines in the engineering design process are tightly interrelated and affected by each other, the process requires MDO methodologies that account for multiple fields simultaneously. The environment must be convenient and intuitive for the user, yet it must still integrate distributed resources. In response to these requirements, this paper proposed a collaborative design environment for experts by establishing a PSE portal for an MDO. Authenticated design experts can use the system, design the target, and use all the resources available via the PSE portal, from anywhere and without any spatial restriction. The PSE portal proposed in the paper meets all the requirements of the MDO and has a flexible architecture that can be expanded to products with complex design processes, including aero vehicles, automobiles, and ships.

Acknowledgement

This paper was supported by Konkuk University in 2006.

References

[1] Lee Jae-Woo. Multidisciplinary aircraft wing design using the MDO framework, KSAS; August 2004. p. 23–33.
[2] Hey AJG, Papay J, Keane AJ, Cox SJ. Component based problem solving environment. LNCS, vol. 2400, Springer-Verlag; 2002. p. 105–12.
[3] Foster I. Globus Toolkit Version 4: software for service-oriented systems. In: IFIP international conference on network and parallel computing. LNCS, vol. 3779, Springer-Verlag; 2006. p. 2–13.
[4] Czajkowski K, Foster I, Kesselman C. Agreement-based resource management. In: Proceedings of the IEEE, vol. 93(3); March 2005. p. 631–43.
[5] von Laszewski G, Foster I, Gawor J, Lane P, Rehn N, Russell M. Designing grid-based problem solving environments. In: 34th Hawaiian international conference on system science; 2001.
[6] Nobrega R, Barbosa J, Monterio AP. BioGrid application toolkit: a grid-based problem solving environment tool for biomedical data analysis. In: VECPAR 7th international meeting; June 2006.
[7] Maheshwari KC, Olabarriaga SD, Botha CP, Snel JG, Alkemade J, Belloum A. Problem solving environment for medical image analysis. In: Twentieth IEEE International Symposium on Computer-Based Medical Systems; June 2007. p. 165–70.
[8] Cho Jung-Hyun, Kim Yun-Hee, Kim Jin-Ho, Kin Jong-Am, Cho Keum-Won. e-AIRS : e-Science portal for collaborative research of aerospace field, KISS, vol. 15(2); 2008. p. 96–104.
[9] Walker DW, Giddy JP, Weatherill NP, Jones JW, Gould A, Rowse D, et al. GECEM: grid-enabled computational electromagnetics. In: Proceedings of the UK eScience all hands meeting; 2003.
[10] Chen L, Shadbolt NR, Tao F, Puleston C, Goble C, Cox CJ. Exploiting semantics for e-Science on the semantic grid. In: Web intelligence (WI2003) workshop on knowledge grid and grid intelligence; 2003. p. 122–32.
[11] Jackson T, Austin J, Fletcher M, Jessop M. Delivering a grid enabled distributed aircraft maintenance environment. In: Proceedings of the UK eScience all hands meeting; 2003.
[12] Workflow Management Coalition, Interface 1 – Process definition interchange V1.1, WfMC Specification, WfMC-TC-1016-P; 1999.

[13] Public Domain Aeronautical Software. The USAF Stability and Control DATCOM, vol. 1, Users manual; 1999.
[14] Lee Seung-Jin. A study on the cruise missile design using multidisciplinary design and optimization approach, Thesis for the Master of Science in Aerospace Engineering, Konkuk Univ.; 2007.
[15] Michalewicsz Z, Nazhiyath G. Genocop III: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In: Proceedings of the 2nd IEEE International Conference on Evolutionary Computation, vol. 2, Perth; 1995. p. 647–51.
[16] Lee Seung-Jin, Kim Woo-Hyun, Lee Jae-Woo, Byun Yung-Hwan. Development of integrated missile system design framework using distributed computing environment. ICCSA 2007, August 2007, p. 301–5.
[17] Kodiyalam S. Evaluation of methods for multidisciplinary design optimization, Phase I, NASA/CR-1998-208716; 1998.
[18] <http://www.phoenix-int.com/documents/ModelCenter7.pdf>.
[19] <http://www.engineous.com/products.htm>.
[20] Weston RP, Townsend JC, Eidson TM, Gates RL. A distributed computing environment for multidisciplinary design. In: 5th AIAA/NASA/ISS MO Symposium on Multidisciplinary Analysis and Optimization, AIAA-94-4372-CP; September 1994, p. 1091–7.
[21] IMAGE: Tutorial, Version 1.17, Aerospace systems design lab., Georgia Institute of Technology; 1999.