Journal of Industrial Information Integration 1 (2016) 14–19

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

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

A multidisciplinary design and analysis environment and its application to aircraft flight dynamics analysis

Chengen Wang*

State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China

ARTICLE INFO

Article history:
Available online 4 March 2016

Keywords:
Multidisciplinary design and analysis (MDA)
Aircraft flight dynamics
Information integration
Workflow management

ABSTRACT

Complex engineering problems have to be jointly solved by many disciplinary teams with multitudinous computational software packages and physical experiment systems. Thus, the efficiency and effectiveness of solving complex engineering problems largely depend on effective collaboration among specialist engineers, seamless integration of disparate cyber-physical systems, and transparent interoperations of heterogeneous data sources. This paper presents a multidisciplinary design and analysis (MDA) environment in conjunction with its application to aircraft flight dynamics analyses. The MDA infrastructure builds a cybernetic platform that integrates structure analysis and flow computation systems with wind tunnel experiment systems; reconciles and interoperates diverse data sources generated by the cyber-physical systems. It is demonstrated that application of the MDA environment cost-effectively enhances work efficiency, team collaboration, and solution optimality in aircraft flight dynamics engineering processes.

1. Introduction

Conventional engineering methodology decomposes a large engineering problem into disciplinary sub-problems, and attains the problem solution by combining partial solutions to the sub-problems. Apparently, conventional engineering methodology fails to take full account of interactions among the disciplinary sub-systems. A complex engineering system, such as various vehicles, power generation systems and ships, invariantly exhibits heavy interdisciplinary interactions. Steadily growing engineering complexity demands an engineering problem be solved in terms of global optimums from multidisciplinary perspectives. To this end, a holistic information infrastructure is needed to accommodate myriad cyber-physical systems and divers data sources, to manage engineering processes/workflows, and to booster human collaboration. Therefore, various information integration infrastructures are growingly implemented to enhance interoperability of heterogeneous data sources in fields of multidisciplinary design optimization [15].

For more than a decade, we have been studying and developing a multidisciplinary design and analysis (MDA) environment that enables engineers to proficiently tackle complex engineering problems [16]. The MDA environment provides a wide spectrum of enabling techniques including application and information integration, parameter mapping, data management, product lifecycle modeling, project scheduling, searching algorithms, etc. The MDA environment has been applied to solve a number of engineering problems, such as turbine disc design [5] and aerodynamic blade design [17]. This paper presents the MDA environment in the context of its application to aircraft flight dynamics analysis.

The structural design of aircraft requires the knowledge of external loads acting on individual components, which depend on an aircraft's mission, and thus its operational environments [1]. Aircraft flight dynamics analysis calculates aerodynamic loads exerting on cockpits, fuselages and wings, and predicates corresponding structural responses, such as deformations, stress distributions, vibrations, and failures in the worst cases. In addition to individual disciplinary computations, aircraft flight dynamics analysis has to deal with extensive interactions among dissimilar disciplines of aerodynamics, structural dynamics, flight control, strength analysis, wind-tunnel experiments, etc.

Conventionally, the structural design, aerodynamics computation, flight control design, and other engineering activities were independently performed by different groups of engineers. And, solutions to disciplinary problems were gradually improved by iterative design loops, which frequently engendered higher engineering costs and performance degradations. Conceivably, effective and efficient coordination and collaboration among engineers from different backgrounds are crucial to rapidly achieving desirable engineering solutions within reasonable time windows. And, the multidisciplinary nature of aircraft flight dynamics analysis demands for MDA techniques to bolster up collaboration among disciplinary engineer teams. Using collaborative engineering, the

---

* Tel.: +86 24 83683619.
E-mail address: wangc@mail.neu.edu.cn

http://dx.doi.org/10.1016/j.jii.2016.03.002

collocated team is replaced with an interactive team structure where the team members are geographically distributed and the best engineering talents can be applied to the design effort regardless of physical location [9]. Collaborative design supports sharing product data and know-hows, and providing collaborative tools to bring the multidisciplinary team together [11].

It was recognized that the increasing amount of collaborative work in engineering made it necessary to create environments that would foster the coordination and cooperation among engineering groups [8]. For instance, a collaborative aircraft design environment, called AMANDA, was implemented for building and controlling aircraft design workflows, which integrated applications by conjunctively using CORBA and JavaBean components [7]. And, it was designed to deal with parallel and sequential programs and massive data exchange between the integrated programs, with potential use in designing elastic airplane and an air-cooled turbine. Another multidisciplinary aircraft design and evaluation (MADE) framework was developed to integrate legacy software packages, to manage design projects, and to authenticate data access [4]. A recent publication also described a collaborative platform for managing tool integration, translating and exchanging engineering product data, maintaining data consistency and propagating change, and co-ordinating the distributed design process [18].

Patel et al. [10] argued that successful collaboration requires effective support of technical tools for managing teams, tasks, resources, and processes. To increase tractability of intensive collaborative design interactions, engineering activities have to be managed in the context of engineering workflows or processes. Workflow management techniques build interlinks among engineering tasks, and create associations among engineers, tasks, applications, and data sources. Workflow management techniques make collaborative design more palpable. Effectively defining and managing engineering processes are prerequisites of design collaboration in developing complex engineering systems. Hürlimann et al. [6] developed a multidisciplinary process for linking model generation, load generation, load application, structural sizing and postprocessing. Their multidisciplinary process features modules for the generation and processing of all static loads including aerodynamic loads, fuel loads, engine loads, landing gear loads and inertial loads of both structural and non-structural masses.

For nearly two decades, researchers from management science and engineering disciplines have extensively studied workflow management techniques. Consequently, an increasing number of workflow management systems have been implemented in diversified sectors such as public administration, health care, enterprise planning, as well as engineering design and analysis. Differing from its counterparts in administrative and managerial sectors, a workflow management system in engineering fields remarkably accentuates integration of engineering resources and global optimization of contradicting disciplinary goals. Naturally, workflow management in conjunction with integration and optimization techniques constitutes a core competency of the MDA environment.

The rest of this paper is organized as follows. Section 2 describes the MDA environment's integration services in the context of flight dynamics analysis. Section 3 addresses the MDA environment's project and workflow management functionalities. Section 4 finally concludes this paper.

2. Engineering integration

It is impractical for engineers to achieve design collaboration without support of a unified cybernetic framework or environment that tightly integrates and effectively manipulates diversified software packages and heterogeneous data sources. The MDA environment offers two types of integration techniques, which respectively conjoin legacy problem-solving codes and updated software tools, and perform operations across disparate data sources.

Categorically, software packages for aircraft flight dynamics analysis includes geometric modellers, which may be commercial CADs or ad hoc B-rep modeling codes, computational structural dynamics (CSD) solvers, computational fluid dynamics (CFD) solvers, and optimization tools. CSD solvers may be commercial CAEs or in-house developed finite element method (FEM) packages. Meanwhile, commercial or in-house developed CFD systems often solve Navier–Stokes equations by using either finite difference methods (FDM) or finite volume methods (FVM). Particularly, aircraft flight dynamics analysis adopts a number of ad-hoc software tools to construct scaled component prototypes for being used in windtunnel simulations, collecting and post-processing experimental data.

Despite evident differences in their intrinsic algorithms, software packages used in aircraft flight analysis have some characteristics in common. For example, most of the software applications used for aircraft flight analysis are monolithic software packages coded in procedural programing languages, such as C and Fortran. It is common to find that these applications are deficient and devoid of graphical user interfaces, which force users to memorize numerous command lines. Another distinctive drawback of these applications is that they completely rely on unstructured data files for inputs and outputs.

The MDA environment provides component-based services for plugging the diverse software applications into a shared platform without changing existing codes. In general, application integration environments are implemented by using software components technologies, namely the CORBA, Enterprise Java Bean (EJB), and COM/DCOM techniques. Component techniques offer advantages of packaging their data and implementation details through a commonly shared programming model and only exposing their services via public interfaces.

In this work, component servers developed to integrate the legacy applications are basically the out-of-process remote COMs (executable files). A proxy/sub model is used to allow calls to an out-of-process COM's methods to be invoked via a specific pointer, *vtable pointer,* which has access to methods exposed in a COM interface. Interested readers may find details of component based integration services in our previous publications [12,13,14,16]. Software applications integrated into the MDA environment are encapsulated by software components and their executions are controlled via surrogate objects.

In the MDA environment, applications are integrated on basis of registration information including their names, disciplines, command lines, versions, running environments, executable files with a suffix of *exe* or *bat*, numbers and names of the input/output files, host machines and their absolute paths. Applications integrated render their full control to surrogate components in the MDA environment. Fig. 1 presents an exemplary screen snapshot that illustrates how structural displacements of a wing are calculated by activating an integrated commercial CAE package. During computation processes, the MDA environment automatically handles input/out data, and activates sub-computations in accordance with predefined rules. It can also highlight characteristic computational results of interest, such as critical displacements and stress distributions approaching thresholds. Integration of applications into the MDA environment makes it viably attainable for software executions to be effectively scheduled and controlled in serial and parallel sequences.

On basis of integrating the cyber-physical systems, the MDA environment provides information integration services for interoperations across heterogeneous data sources. Information integration services are categorized into three groups: (1) building mapping relationships among data sources, (2) transforming data schemas, and (3) maintaining global information consistency and