Concurrent Engineering
http://cer.sagepub.com

Towards Semantic Interoperability of Collaborative Engineering in Oil Production Industry

Y.-S. Ma

Concurrent Engineering 2009; 17; 111
DOI: 10.1177/1063293X09105333

CONCURRENT ENGINEERING: Research and Applications

Towards Semantic Interoperability of Collaborative Engineering in Oil Production Industry

Y.-S. Ma*

Department of Mechanical Engineering, University of Alberta, AB T6G 2G8, Canada

Abstract: This article presents a new approach to investigate the interoperability in collaborative engineering across design, manufacturing, operation, maintenance, and end-of-life stages with an expanded generic feature paradigm. The fundamental strategy is to develop a unified and consistent interoperable semantic scheme in order to support an open and flexible knowledge realization system. Oil production industry is aimed as the application field.

Key Words: collaborative engineering, interoperability, product lifecycle management, generic feature.

1. Introduction and the State of the Art

Collaborative engineering is a technological approach of engineering informatics that emphasizes comprehensive virtual modeling of products and processes in order to support multidisciplinary collaborations transparently across the stages of the lifecycles of products or projects. With highly sophisticated product and process models, engineers at different stages of product lifecycles who use different semantics and engineering patterns [1,2] will be able to work together effectively and seamlessly regardless of their diversity in project roles [3–5]. The essence of this technology is a decision making process that can incorporate the input from different aspects of engineering. Given the complex nature of the modern engineering system integration in industry, the collaborative engineering capability has become a business competitive edge. In the oil production application, the success of large-scale refinery operations and oil sands projects demands a coherent collaborative engineering model that enables operation managers, production engineers and design engineers to freely share information for all the inter-related tasks, e.g., cost estimations and construction plans, the process design and engineering, equipment design and manufacturing [6], operational procedures, production capacity and maintenance management, etc. A recent breakdown of a catalytic cracking unit in Petro-Canada's Edmonton refinery caused significant unexpected damages to the company, and this unfortunate incident offers an example of the need for collaborative engineering [7]. Small and medium enterprises, such as an oil well drilling equipment provider, can also benefit from an integrated product and process management system; it enables higher productivity, more innovative designs, less project delay, and a quick response to market conditions.

However, there is a severe interoperability limitation in the current engineering systems. There are many isolated information 'islands' in a company where each island has a separate application domain and separate semantic definitions. The engineering knowledge and other engineering constraints cannot be commonly represented across different computer systems. This is because there is no connection mechanism for real time information updating and consistency checking for representation of knowledge in computer models. For this reason, different engineering information systems can not be interfaced effectively. Even within a single engineering domain, such as within a design department, incompatible semantic representations and lack of data associations mean that design information is impossible to be linked with the knowledge and constraints that are used to make the design choice. Therefore, the interoperability of engineering software tools must first be addressed [4].

Traditionally interoperability refers to the exchangeability of system data created by different computer software packages. In technical terms, it can be described as the interpretability of data types and information content. For example, computer-aided manufacturing software uses data from computer-aided design (CAD) models. Presently, interoperability refers

*E-mail: yongsheng.ma@ualberta.ca
Figures 1–3 appear in color online: http://cer.sagepub.com

1063-293X/09/02 0111–9 $10.00/0 DOI: 10.1177/1063293X09105333

to the capability of a networked computer system that has consistent information representation, self-embedded definitions, dynamic function realization, coherent and collaborative information sharing, and boundary-less information services. In simple terms, this type of systems can be best described as having the ability of 'plug and play' over a distributed network.

The field of engineering informatics still lacks the interoperability necessary to implement collaborative engineering effectively. For example, geometrical modeling is carried out by CAD systems whereas databases are used to store nongeometric data or meta-linked records, such as resources plans and schedules. To exchange geometrical data, standards like IGES and STEP have been developed, but the interoperability achieved with these standards is limited [5,8–11]. Nongeometric engineering information is striped off during translations. As a result of this truncation high-level information such as engineering principles and embedded relations defined in product or process models is either lost or absent due to the lack of a common modeling method. Such the data translation approach creates inconsistent copies of computer files leading to a 'snowball' effect, where mixed data sets become bigger and bigger without validation and associations.

Semantic content sharing and consistency checking is a new and exciting area of research that could offer solutions to the interoperability problem. To incorporate engineering semantics, feature-based technologies have been used in engineering IT systems because 'features' have flexible definitions and are easily understandable by engineers [6]. A limitation of these technologies is that they are confined to separate software packages and there is no interoperability among them [12,13].

2. Objectives

This article presents a research framework exploring the most fundamental and challenging mechanisms of interoperability at the semantic level in order to develop an effective and efficient system integration and collaboration method among engineering information systems. The author's long-term vision is to achieve an integrated, consistent, flexible and scalable product, and process engineering management system [3,14]. The prototype system to be developed is expected to demonstrate the capability of supporting complex collaborative tasks that are dynamically configured among collaborators in a consistent, intelligent, and automated manner.

3. Research Methodology

The research methodology will be based on both the object-oriented software engineering approach [5,15] and the web-service oriented architecture design [1,2,16]. The object-oriented approach has been effective in the expanding feature-based technology to complex engineering applications via more flexible and versatile component-level, assembly-level, and other advanced associative features. Web-service embeds autonomous object procedures in the form of agents and distributes functions across the internet. The proposed methodology is scalable because these two technologies are consistent in behavior and there is no limit to their application complexity and data volume.

Informatics models are effective to IT development via two essential aspects, namely generic data structures and the adaptability of functional methods. Generic data structures allow the resulted informatics models to be applied to a wide range of applications; adaptability, enabled by properly designed foundational codes and modules, allows for these models to be easily implemented. Hence, in the context of this research framework, the measures of a scientific and successful method are the generic reusability of templates developed in a prototype system, the effectiveness and efficiency of application procedures, and the potential values that can be created for end users in the sense of new functionality, and productivity. In order to test and prove the usefulness and completeness of such a scheme, the scheme must be applied to several nontrivial applications. Once applied it is necessary to verify that the predefined definitions and methods of the scheme can be used without significant or substantial repetitive work other than the necessary one-time development, instantiations, and limited cycles of functional tests and enhancements. The article uses a few cases in the oil production industry to show some inherent system integration issues. These issues affect all aspects of engineering projects, e.g., design, construction, installation, and maintenance.

4. Framework of Proposed Research Approach

This research framework consists of four integrated key components of research as discussed below.

Part 1: Standardization of Generic Feature Types and Methods

To enable information repository, communication, and processing, the generic definitions of data structures corresponding to the software conceptual elements, algorithms, processes, modules, and knowledge rules are necessary to ensure the implementation functionality, usability, quality, and maintainability. To achieve an interoperable environment, the neutral data structure definition hierarchy, covering all types of data entities, is required [11]. As a unique set of data structures, a standard generic feature definition scheme must be established for different application domains, i.e., the *generic feature* class.

The first part of this proposed research is to continue the investigation of *generic feature* class in order to support information diffusion between different features, such as CAD and computer-aided engineering features and other tailored information needs. To test the class reusability and to support the intended industrial applications, this proposed research will incorporate those features encountered in pressure vessel and pipeline equipment lifecycles and apply them throughout conceptual design, detailed design, analysis, manufacturing process planning, resources management, and to maintenance. A similar method as used in the author's previous modeling work for cooling channels in plastic mold design can be used [15]. The expected results include a 'blue print' of feature *class* definitions and their corresponding functional *methods* that are supposed to be extendable for the engineering of most chemical processing devices and equipment.

Part 2: Feature Mapping Scheme and Dynamic Repository

An interoperability technology in an engineering integration platform with a generic feature scheme must support dynamic and flexible interactions among collaborative users [16]. Representing each engineering collaborator as a grid in a vertically and horizontally intersecting grid matrix, the technical issue of information sharing is the establishment of effective interfaces. The vertical relations represent the integration between different abstraction levels of information, data entities, features, and macrosemantic facts. Horizontal relations represent information sharing between various applications. The vertical interfaces will be discussed in the next part; this part addresses multiple systems working together with different inter- or intra-feature interactions horizontally [8].

The essential challenge of horizontal interactions is that all the semantic features of a product or process model become automatically interpretable, accessible, and available by using a scalable engineering repository, selected data streaming, and scope-sensitive algorithms via web-services. This technological target requires dynamic mapping of feature types or semantic patterns used among different systems based on a neutral and consistent data structure. This structure will contain semantic features that are defined in the form of *sets* with well-defined entity properties and constraints. Web-services are to be used for updating feature contents via agents in an active manner across the internet. Project information dependency among different parties in a refinery construction can be best shown with Figure 1.

Figure 1(a) gives a partial outlook of a refinery. It is a common practice that grating panels are used to construct working platforms around such facilities by sub contractors. Figure 1(b) shows a close look of those obstacles, i.e., pipes, devices, etc., when applying grating panels. In fact the platform outer profile and the inner obstacle openings have to be considered simultaneously in order to make a feasible design for easy installation. The author has developed a software package for a company to design grating panels by generating drawings, cutting profiles automatically. Figure 1(c) shows the user interfaces developed reflecting a *grating platform feature* data structure and methods. Figure 1(d) shows a generated platform layout. Optimization for the standard sizes of material supplies, panel gaps, and cutting profile allowances is carried out by the software.

Clearly, the outer boundary and inner opening profiles are dependent to final design model of the facility. If the 3D virtual facility model is available, such that profiles can be extracted from those related entities, then the effort to develop grating design can be significantly reduced. The grating area boundaries can be easily generated from the intersection between the equipment model and a platform plane at a working platform level. The grated platform supporting structure can also be automatically generated. The automation of certain design steps can be the functions of a separate system as this example software tool developed by the author even though the final proof check by a qualified engineer is necessary. Therefore, with effective information sharing, the design preparation for grating can be done with a fraction of the time that is currently required as well as higher accuracy. On-site modification of grating design and cutting of grating panels can be reduced or even totally avoided.

Based on the recent work, this part of the research is to investigate an effective mapping method to efficiently map features between the master engineering model and the individual application view models. This method will also ensure the consistent evaluation of all 'co-existing' facets of the engineering lifecycle space. Since there are unlimited types of application features, identifying advanced engineering features is a separate domain of research beyond the scope of this proposal. However, existing predefined features reported so far in the literature, and those defined in STEP and other earlier works [17], are to be studied and analyzed. A number of advanced features encountered during the testing case analysis are also to be modeled and implemented.

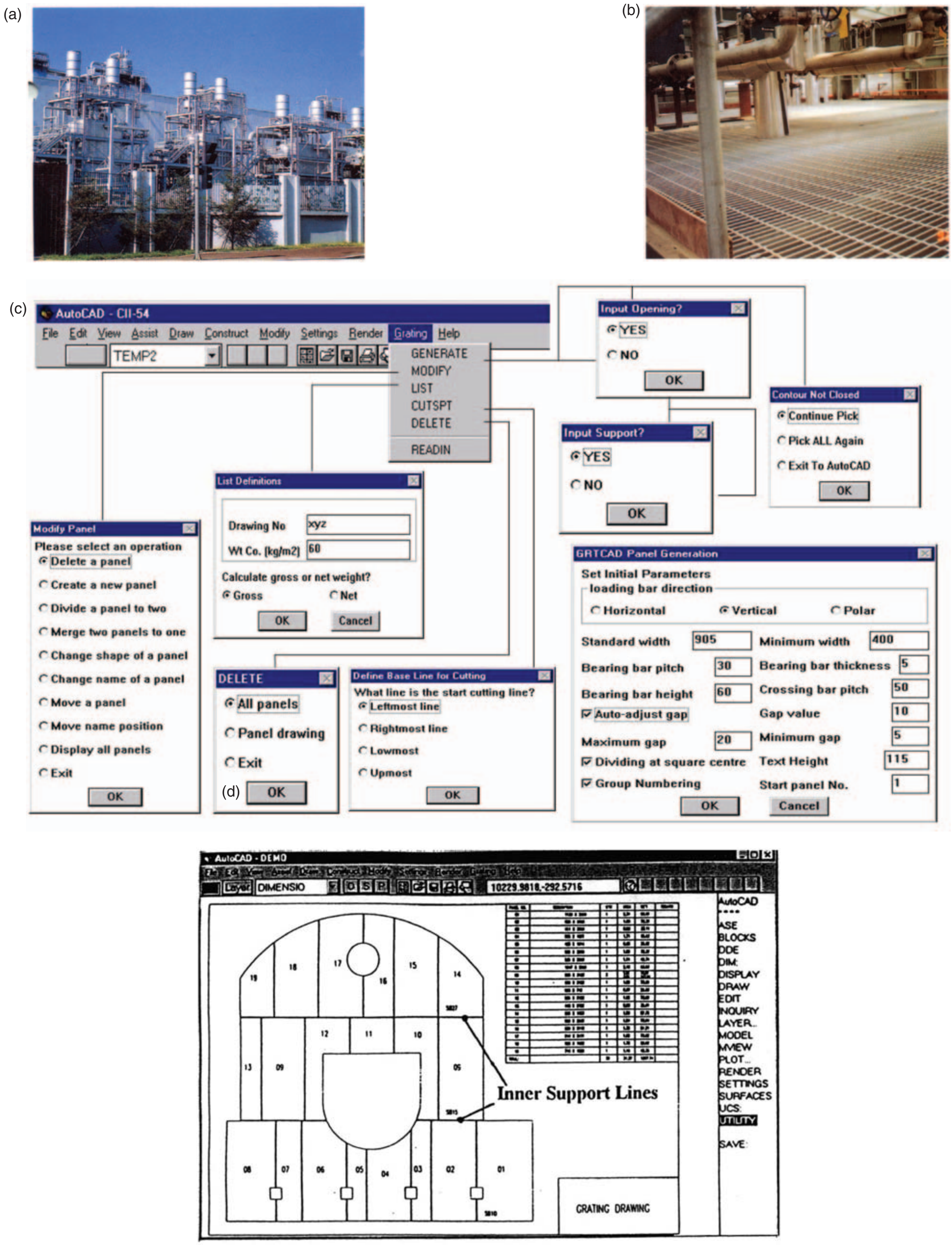

Figure 1. An example of grating design software: (a) a picture of oil refinery facility, (b) a close look to on-site grating space constraints, (c) a grating design software tool user interfaces showing the defining steps and settings, (d) a grating design layout.

The author's previous work has a limitation that the repository database table structure was manually created and, consequently, the automatic and dynamic definition of feature tables has not been realized. New effort is required to explore the dynamic manipulation of the database schema and the automatic configuration of feature relations by interpreting instructional messages generated from end users [10,13,18] to support the creation, modification, and editing of flexibly defined features with complex entities.

Part 3: Constraint Management and Interfacing Mechanisms

This part covers a unified modeling method for different application natures with an emphasis on the abstracted structure of constraints. Constraints provide invariant characteristics in a process model and must be explicitly defined to specify relationships among features as well as geometric or topological entities. With the generic feature definition, constraints are regarded as associated attributes attached to a set of entities. Although constraints are applied for different purposes, they fall into some common types that can be generalized, such as geometric and nongeometric ones [19,20]. This part proposes a research effort to develop a comprehensive method that can effectively model, apply, evaluate, and edit complex, associative, and smart constraints in a collaborative environment. In the proposed collaborative environment, researchers have to develop a unified approach that can support constraints that are complex, associated or cross-referred, and self-defined and maintained. They have to prove the capability for versatile definitions because end users could flexibly define their types of constraints.

Figure 2 shows a scaffolding design case for a spherical oil storage tank. Figure 2(a) gives a picture of the working site where the spherical tanks were being constructed. This case is based on a consultancy project carried out by the author. Figure 2(b) shows a part of an automated external scaffolding design by making use of the geometry of the tank design model. Figure 2(c) shows a similar design for the tanks' internal scaffolding. The software tool developed can generate the special layout of those standardized tube structures, clamps, planks, ladders, guard rails, on-site bill of materials. As shown in the figure, clearly, the rules for designing scaffolding structures have been embedded into the *scaffolding_generation()* function method. The data structure of the software allows the selection, configuration, evaluation, and modification of standardized units which can be appreciated as instances of design or engineering features. The embedded design rules are applied to ensure that the scaffolding is well supported and attached to the spherical external or internal surfaces. Working platforms are designed according to the typical industrial practice guidelines. The high level rules to be applied include *applying the right supporting spans to ensure strength*, and *align all the support tubes as much as possible to maintain good loading distribution*, etc. If the given 'input parameters' from the tank geometry are changed, the program can regenerate the scaffolding structure automatically. Then, the system can support 'if-then' design evaluation during the design stage. The detailed bill of materials can significantly reduce the on-site cutting or 'fitting' work and hence the erection processes are smoothen with great on-site time saving. With the unified coding of each 'bundle' of the pre-cutting materials, logistic work is streamlined. Eventually, the integration of storage tank design and its scaffolding design can be concurrently done although these tasks are carried out by different contractors in the engineering project hierarchy. If there is any issue during the re-evaluation process, notifications to the right task-owners will be served. Hence, effective change propagation, evaluation and validation methods within the master product and the process models will be investigated and demonstrated. Similar cases from the industry, such as oil well equipment design and drilling process simulation can be further studied to evaluate the effectiveness and efficiency of such mechanisms.

To achieve the interoperability along the vertical direction, interfacing at different levels of abstractions has to be supported as well. At the semantics level, a knowledge-based system can be seamlessly interfaced with a feature-based system in order to create integrated product and process models. A more comprehensive and efficient method is needed for the evaluation of coherent consistency between high-level semantic product models, representative features, and associated product and process data structures. In turn, feature-based systems can be seamlessly interfaced with detailed data-oriented procedures for tedious geometrical and nongeometrical entities via streamed and reusable entity-type-based methods supported by object polymorphism.

Figure 3 shows a case of piping system design during an oil rig design process. The conceptual design is usually based on the past experience and many equipment items on board have to be supplied by specialized companies after bidding for the project. The rules to be embed in the design could be "*the minimum crossing space between pipelines has to be more than x meters*", "*the use of lower space is prioritized for the ease of maintenance*", "*reserving maximum space for operational use*", etc. In implementation, to support these rules, constraints between the pipelines and the surrounding environment have to be created and evaluated.

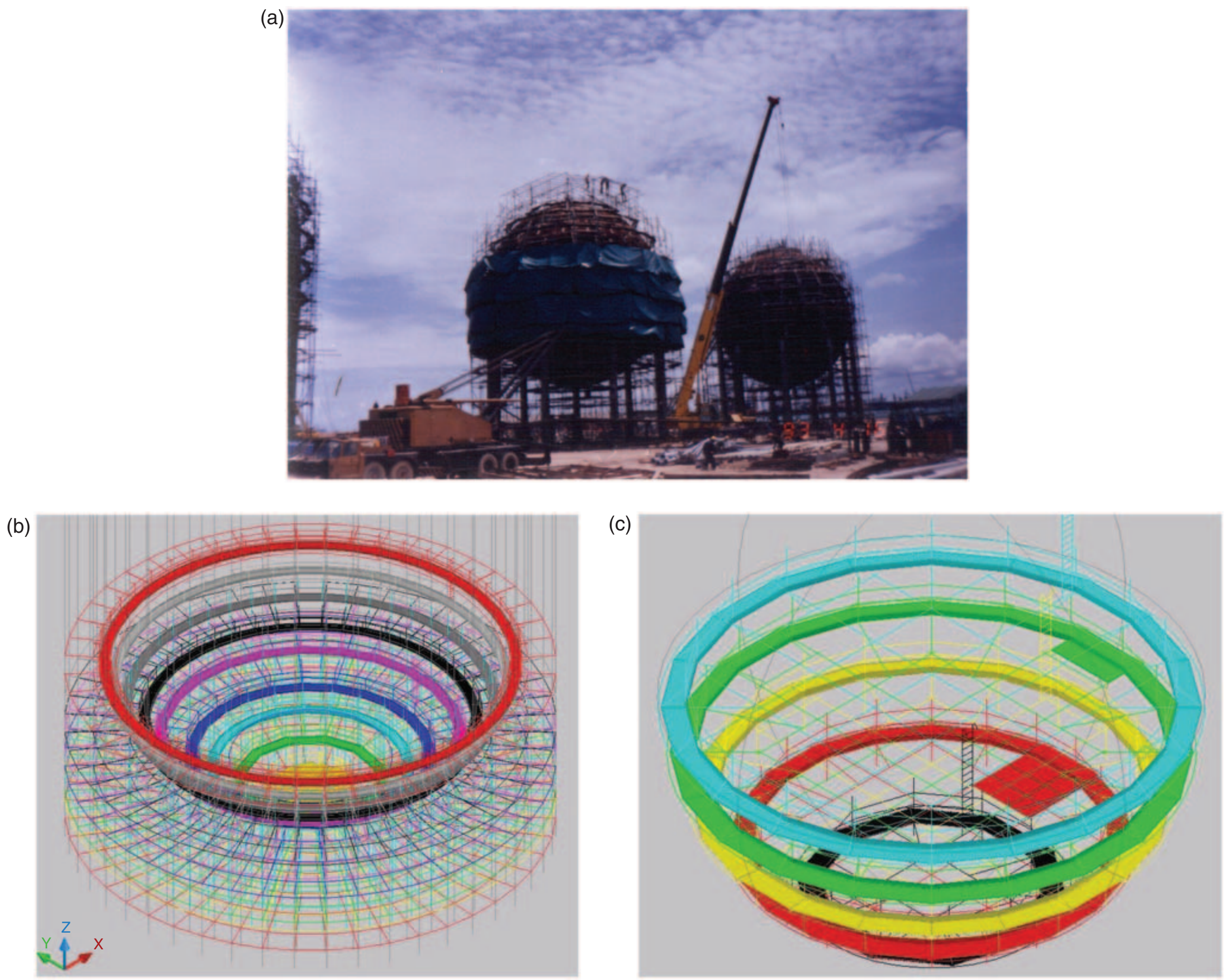

Figure 2. Partial scaffolding design for spherical oil storage tank's external and internal operations.

The issue is that the surrounding environment has to be evolved from a very vague estimation, to abstracted and detailed design versions, the design states with previously-used equipment in phases, and eventually to the design results with confirmed equipment installed. Figure 3(a) shows an intermediate stage of the design. Figure 3(b) shows one of the optimized pipeline paths. Many optional alternatives exist when a sub-system is to be added or replaced in the design and the optimization procedure has to be run iteratively. The whole design process has to be verified and reconstructed frequently.

Part 4: Communication Language and Supporting Algorithms

A custom subset of the commonly-used XML language across the internet, i.e. feature-based mark-up language (FML), is required to support communications or transactions of features and the associated processing methods or algorithms. The feature operation concept is generated and tested from the author's past research as presented in [19,21]. This proposed feature mark-up language should support the interpretation, processing, editing, and generation of feature operation packages over the web [6,12]. This new language is the packaged 'commands' to enable effective functioning of application operations. To prove the concept, a prototype version of the suggested FML is to be developed. Information integrity check, flow management, language compiling and interpretation, and the accessibility to evaluation and validation procedures are to be investigated. Research activities include term definitions, the development of message composing structures, establishing evaluating rules and processes, and the investigation of execution and status reporting methods. This language is to be tested via simulated engineering collaboration interactions and designed scenarios within a controlled

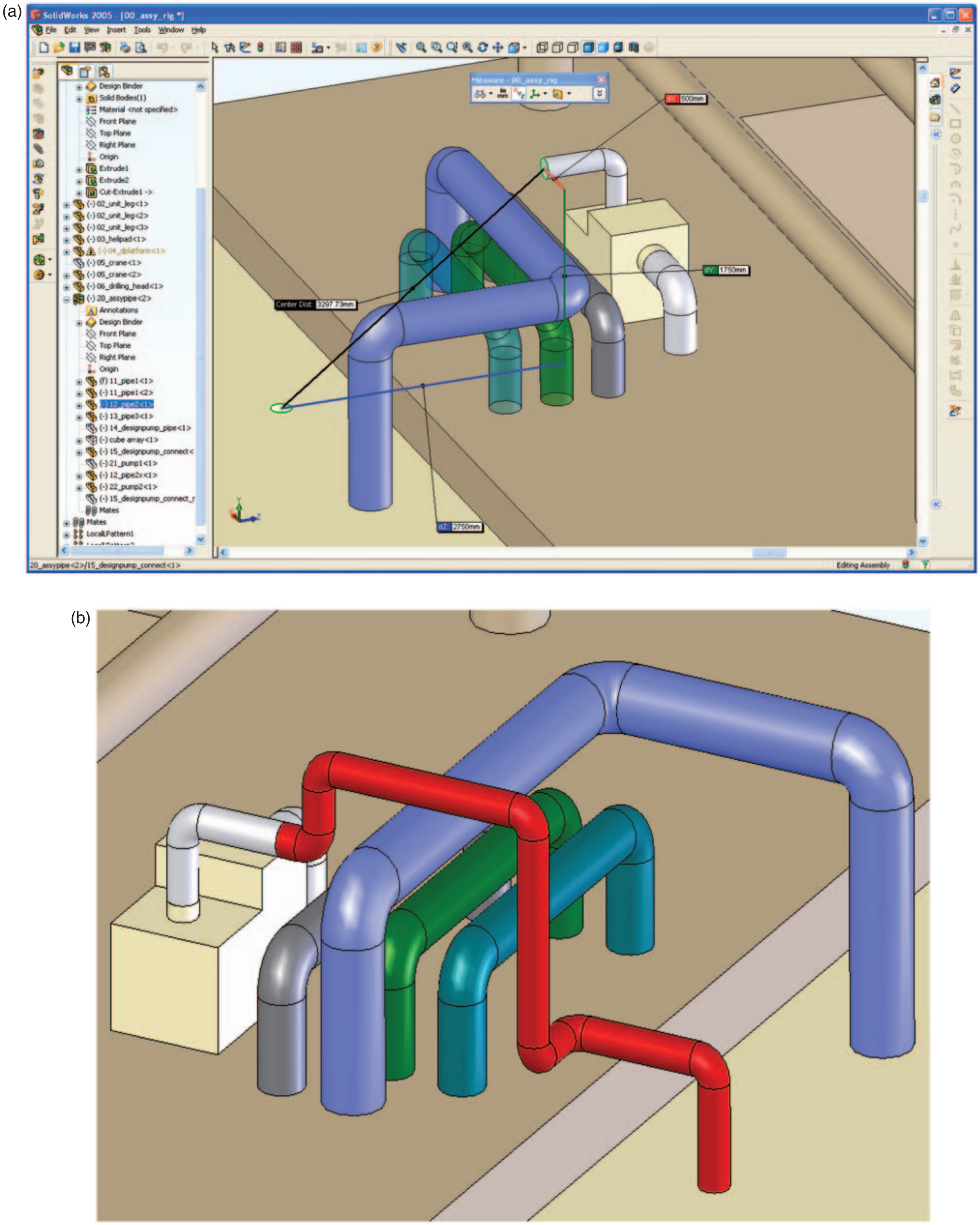

Figure 3. A case of pipeline design optimization in oil rig design: (a) a smart design environment with specified constraints, (b) a resulted piping path after optimization.

computer network in stages. Two approaches, i.e., using a centralized database and using a cluster of federal associative databases, are to be investigated as benchmarks.

5. Novelty and Anticipated Significance

The focus of this article is to seek a solution under a new interoperability paradigm based on *associative engineering features* [15]. Feature-level interoperability could support associated semantics, patterns, relations and change propagation in a dynamic and closed-loop manner. This would result in the enhancement of the richness and flexibility of knowledge representation and constraint management within a well-defined product or project modeling space [22,23]. In application, new collaborative engineering management systems can be developed supporting contractors and equipment manufacturers in engineering projects systematically. Once a unified solution is implemented, via a portal service, collaborative engineering operations can be supported from small to large scale.

References

1. Dolenc, M., Klinc, R., Turk, Ž., Katranuschkov, P. and Kurowski, K. (2008). Semantic Grid Platform in Support of Engineering Virtual Organisations, *Informatica*, **32**(1): 39–49.
2. Shehab, M., Bhattacharya, K. and Ghafoor, A. (2007). Web Services Discovery in Secure Collaboration Environments, *ACM Transactions on Internet Technology*, **8**(1): 5:1–22.
3. Anumba, C.J., Pan, J., Issa, R.R.A. and Mutis, I. (2008). Collaborative Project Information Management in a Semantic Web Environment, *Engineering, Construction and Architectural Management*, **15**(1): 78–94.
4. Jiang, P., Shao, X., Qiu, H. and Li, P. (2008). Interoperability of Cross-organizational Workflows Based on Process-view for Collaborative Product Development, *Concurrent Engineering: Research and Applications*, **16**(1): 73–87.
5. Kim, K.Y., Manley, D.G. and Yang, H.J. (2006). Ontology-based Assembly Design and Information Sharing for Collaborative Product Development, *Computer Aided Design*, **38**(12): 1233–1250.
6. Wang, Y. and Nnaji, B.O. (2006). Document-driven Design for Distributed CAD Services in Service-oriented Architecture, *Journal of Computing and Information Science in Engineering*, **6**(2): 127–138.
7. Mcintyre, D. (2008). Petro-Can's Cat Cracker Woes Continue, The Edmonton Sun, Sun Media. Available at: http://cnews.canoe.ca/CNEWS/Canada/2008/08/22/6537001-sun.html.
8. Bronsvoort, W.F. and Noort, A. (2004). Multiple-view Feature Modeling for Integral Product Development, *Computer Aided Design*, **36**(10): 929–946.
9. Ouertani, M.Z. and Gzara, L. (2008). Tracking Product Specification Dependencies in Collaborative Design for Conflict Management, *Computer Aided Design*, **40**(7): 828–837.
10. Ou-Yang, C. and Chang, M.J. (2006). Developing an Agent-based PDM/ERP Collaboration System, *International Journal of Advanced Manufacturing Technology*, **30**(3–4): 369–384.
11. Pratt, M.J. and Srinivasan, V. (2005). Towards a Neutral Specification of Geometric Features, *International Journal of Computer Applications in Technology*, **23**(2/3/4): 203–218.
12. Li, W.D., Fuh, J.Y.H. and Wong, Y.S. (2004). An Internet-enabled Integrated System for Co-design and Concurrent Engineering, *Computers in Industry*, **55**(1): 87–103.
13. Xue, D. and Yang, H. (2004). A Concurrent Engineering-oriented Design Database Representation Model, *Computer Aided Design*, **36**(10): 947–965.
14. August, D., Chang, J., Girbal, S., Gracia-Perez, D., Mouchard, G., Penry, D., Temam, O. and Vachharajani, N. (2007). UNISIM: An Open Simulation Environment and Library for Complex Architecture Design and Collaborative Development, *IEEE Computer Architecture Letters*, **6**(2): 45–48.
15. Ma, Y.-S. and Tong, T. (2003). Associative Feature Modelling for Concurrent Engineering Integration, *Computers in Industry*, **51**(1): 51–71.
16. Shen, W., Hao, Q., Wang, S., Li, Y. and Ghenniwa, H. (2007). An Agent-based Service-oriented Integration Architecture for Collaborative Intelligent Manufacturing, *Robotics and Computer-Integrated Manufacturing*, **23**(3): 315–325.
17. Ma, Y.-S., Britton, G.A., Tor, S.B. and Jin, L.Y. (2007). Associative Assembly Design Features: Concept, Implementation and Application, *International Journal of Advanced Manufacturing Technology*, **32**(5–6): 434–444.
18. ORACLE. (2008). Online Documentation, Available at http://www.oracle.com
19. Li M., Gao, S. and Wang, C.C.L. (2007). Real-time Collaborative Design with Heterogeneous CAD Systems Based on Neutral Modeling Commands, *ASME Transactions Journal of Computing & Information Science in Engineering*, **7**(2): 113–125.
20. Panchal, J.H., Fernandez, M.G., Paredis, C.J.J., Allen, J.K. and Mistree, F. (2007). An Interval-based Constraint Satisfaction (IBCS) Method for Decentralized, Collaborative Multifunctional Design, *Concurrent Engineering Research and Applications*, **15**(3): 309–323.
21. Khaled, A., Ma, Y.-S. and Miller, J. (2008). A Service Oriented Architecture for CAX Concurrent Collaboration, In: *Proceedings of IEEE Conference on Automation Science and Engineering*, 23–26 August Washington, DC, USA.
22. Sudarsan, R., Fenves, S.J., Sriram, R.D. and Wang, F. (2005). A Product Information Modeling Framework for Product Lifecycle Management, *Computer Aided Design*, **37**(13): 1399–1411.
23. Yang, Q.Z. and Zhang, Y. (2006). Semantic Interoperability in Building Design: Methods and Tools, *Computer-Aided Design*, **38**(10): 1099–1112.

Y.-S. Ma

Dr Y.-S. Ma joined the Department of Mechanical Engineering, University of Alberta, Canada, as an Associate Professor in September 2007. His main research areas include engineering informatics, CADCAM, product lifecycle management, feature-based product and process modeling. Dr Ma received his BEng degree from Tsing Hua University, Beijing (1986) and obtained both MSc and PhD degrees from UMIST, Manchester University, UK in 1990 and 1994, respectively. Before joining the University of Alberta, he had been an Associate Professor in the school of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore since 2000. He started his career as a Lecturer at Ngee Ann Polytechnic, Singapore from 1993 to 1996. Then, between 1996 and 2000, he worked as a Senior Research Fellow and a Group Manager at Singapore Institute of Manufacturing Technology.