Proceedings in Manufacturing Systems

Proceedings in Manufacturing Systems, Volume 8, Issue 3, 2013 ISSN 2067-9238

A VERTICAL SYSTEM FOR ORDERING, CUSTOMIZATION AND AUTOMATED DESIGN OF MECHANICAL PRODUCTS

George ANDREADIS[1,*], George KASSELAKIS[2]

[1)] Ass. Prof., Mechanical Eng. Dept, Aristotle University, Thessaloniki, Greece
[2)] World Economic Forum – Global Shapers, Athens, Greece

Abstract: *The explosive growth of the Web and the maturity of the CAD technologies lead most of the leading construction companies to incorporate online customer services into their business agenda. Such services include limited online product customization, preview in the form of CAD files, pricing and ordering. The availability of these services assumes the existence of a communication flow among customers, designers and construction lines where the human factor still plays the leading role while automation is limited. In this context we present a generalized service oriented architecture that models a workflow which binds together all the factors of such communication flow. More specifically, we examine the challenges and implement an online decentralized ordering system that enables high product customization, fully automatic product design and price estimation. In order to meet the above challenges we design and implement a system for ordering, customization and automatic design of hydraulic elevator. The primary contributions of this work are: a) the introduction of a Web service oriented decentralized architecture that can be extended to support any kind of mechanical product, b) the implementation of a system that bridges the ordering phase and the designing phase and c) the activation of an automated design of highly customizable products, reducing effectively the human factor in terms of product designing*.

Key words: *ordering, customization, automated design, CAD, expert rules.*

1. INTRODUCTION

CAD (Computer Aided Design) technologies became a key asset in every product design and construction process the last decades. Important contributions of the CAD modeling include a strict description of the illustrated prototype products, ability for further customization and easily managed and distributed product information.

The explosive growth of the Web opened new horizons in information exchange and distributed management. As reported in this [1] recent market report, construction companies understood the potential benefit from the Web and CAD technologies and invested on them. A common offered online service is the presentation of the supplied products in the form of CAD objects. The clients usually chose from a list of standardized prototypes or perform some kind of limited customization to the final product design. Therefore the customers have a complete view of the product they are interested in, in terms of specifications.

Besides this customer-supplier online service we may identify another communication flow enclosed in the strict limits of the respective company. This includes the exchange of CAD objects among the product designers (humans) and the construction queue.

In this context we describe a case study for a system that attempts to move the process of purchasing complicated and customizable mechanical devices to the online world. The product in review is a hydraulic elevator which should be adapted to the needs and physical constraints that affect each specific customer.

Traditionally the process of ordering such a product needed to be done in person. The intrinsic difficulty of customizing the user experience, resolving construction issues and agreeing on a price requires a breadth of knowledge that generally cannot justify the cost of purchasing an elaborate Expert System. However recent advances in the interoperability of systems that are based on Web Services allow for the creation of modular software that incorporates low cost and heterogeneous parts.

The rest of this paper is organized as follows. Section 2 describes the challenges the implementation of our system imposes. Section 3 presents the system’s architecture in general while Section 4 provides a detailed analysis of each architectural component. Finally, Section concludes this work.

2. CHALLENGES

The development of the mentioned system imposes several important challenges. In terms of product design there exist the human factor which currently play active role in the product customization process and it is prone

[*] Corresponding author: 24 Papapetrou str, 55131, Thessaloniki, Greece
Tel.: 2310996355
Fax: 2310996059.
E-mail addresses: *andreadi@eng.auth.gr* (G. Andreadis), *kcorax@gmail.com* (G. Kasselakis)

to errors. Therefore one of the goals is to reduce the humans' design responsibilities. This can be dealt by implementing a reasoning core in the form of software component that takes after the product customization and design given a set of customer specific parameters.

Another issue concerning the customer-supplier relationship is the definition of the final product cost. This can be done manually including the risk of errors or automatically as the summation of the individual product components or according to a standardized list of prices.

Apart from the already described issues there exists the challenge of orchestrating the CAD exchange procedures in terms of customer-supplier basis and inside the company itself during the design and construction process. Each phase include its individual diversity of underlying infrastructure. Different operating systems and CAD software make the whole communication management a tough task. In some cases, even software from the same vendor can have inconsistencies to collaboration with other applications which were not specifically designed to collaborate. Given this heterogeneity, a homogenous communication methodology needs to be applied.

In this context, the authors in [2] implement a Web Services oriented system for exchanging procedural CAD models. Their implementation takes into account the communication between two different commercial CAD software products. Using this approach platform independency is achieved, dealing with the heterogeneity of the various systems. Another benefit of this approach is the decentralization of the various design procedures. Data centers, servers, design engines, applications and clients can be distributed in a wider network or intranet. Such a distributed scheme is also presented in [3]. Dealing with the various CAD file format incompatibilities the authors in [4] presented an XML-based language, namely MP-ML, which is suitable to describe machined parts. More specifically it is formatted as a SOAP envelope where the body covers a breadth of uses such as placing orders and describing common fabrication processes on primitive solids.

Beside the platform incompatibility issues, there exist two business logic concerns. The first is to bridge the communication of designers and manufacturers. Transparent communication procedures and independency of the design and manufacturing process must be ensured as reported in [5]. The second deals with the transparent switch of the workflow from the customer's custom order to the designer.

In order to meet the above challenges we design and implement a system for ordering, customization and automatic design of hydraulic elevator. More specifically the primary contributions of this work can be summarized as follows:

- Propose a Web service oriented decentralized architecture that can be extended to support any kind of mechanical product.
- Implement a system that bridges the ordering phase and the designing phase.
- Enables the automated design of highly customizable products, reducing effectively the human factor in terms of product designing and cost estimation.

3. ARCHITECTURAL OVERVIEW

This specific project aimed at the automation of placing orders for hydraulic elevators and eventually delivering them to the customer. It consists of several disparate subsystems:

- An Expert Rule component, which encapsulates every rule that the infrastructure needs to support. All components may rely on this to consume intelligence which is defined and validated centrally for the corporation.
- The order placement component, which consists of a user interface capable of allowing site visitors to create elevators with the custom equipment and passenger cabin of choice. It is also able to calculate the final price of the product according to the structural and mechanical preferences that the user has decided on.
- The automated CAD designer which is a system for creating custom and parametric designs that satisfy the structural limitations imposed by the building in which the elevator will be placed. The design is custom made and eventually delivered in the industry standard DWG format for consumption by the manufacturing plant.
- A Workflow engine which orchestrates the flow of information among the services. This component is based on the BPEL (Business Process Execution Language) and the BPEL4People to orchestrate the interaction of people in the company when their implication is needed.

4. DETAILED COMPONENTS DESCRIPTION

4.1. Expert Rules

The rule execution component is a central one in the operation of our platform. Its placement allows for very granular changes in the purpose and structure of the rest of the system. The component is based on the ILOG [6] rule execution engine, which provides a suite of visual tools that manipulate the rules and entities involved in their execution. Moreover it provides a friendly way to define business objects and the fundamental interaction that these expose. These enable the non technical users of various departments of the enterprise to create reusable intelligence without the need to explicitly transform their intentions into code. This detachment allows for very granular changes on the way with which the infrastructure operates with respect to validation of maintenance that usually occurs in code or databases. Rules govern all aspects of the application.

More specifically we may identify the following subsystems where such rule based reasoning is applied:

- *Design adaptation to the end client's desires and needs.* The product can be built either as the baseline of its series, or can be customized with materials and accessories, each of which has its own ramifications on the final design of the product.
- *Design parameters pertain to the characteristics of the construction site.* These include adaptation of the dimensions to the available physical space and selection of the fittest reusable mechanical parts.

- *Pricing estimation.* Pricing is calculated as the scalar sum of the cost of materials and components that have been selected. This is then added to the cost of assembling and manufacturing the parts that are needed, which is calculated using Case Based Reasoning.

Finally all the subsystems of this component are exposed as web services which can be consumed independently by every other architectural component in the project.

4.2. Order Placement

Placing an order for an elaborate product without parts that are always common in its class cannot rely on a hardcoded interface. The breadth of the components that constitute a mechanical entity necessitates the creation of a dynamic user interface that can present certain data types. Taking into account the need for a low cost deployment environment a web interface was chosen as the basis of the order placement architectural component.

The component receives information about the objects that must be presented by the rule engine. These can contain information about the physical items that are involved in the manufacturing, details about the design in CAD or the representation that is needed by the rule engine. The objects are formatted as SOAP serialized entities, nested where appropriate in separate objects. They are then examined for their data type and presentation as well as validation needs.

Eventually the component renders the appropriate HTML that can display and edit the contents as needed, and client side JavaScript that is used for the validation of the user input.

The component also satisfies traditional enterprise needs including integration with the existing infrastructure including a CRM.

The Order Management System is a Web based application that provides a unified interface to the users (clients and employees) to manage all the aspects and phases of the production process.

Specifically the major functionalities of the system are summarized as follows:

- Order management. The user may search for already created offers according to various criteria (i.e. customer name), create new orders and associate them to a customer. Moreover, price evaluation of the orders is performed taking into account the corresponding rules. In this context a set of states is defined that an order may be. An indicative example of this set is {started, canceled, suspended, production, …}. One may monitor the status of the order and if it has the necessary credentials, to advance to the next states. Additionally, a wizard is provided that guides the sales people throughout the creation of a new order. The wizard significantly reduces the required time for a new order to be created and it verifies the lift's configuration according to the rules.

- Product Management. Every major lift company introduces new products and continuously evolves and improves the various construction parts of the lifts. Such a development causes a large catalog of products to be created. This catalog needs to be maintained (inserting, deleting and updating). This is a cumbersome task for the employees and it is prone to errors. The maintenance procedure is dramatically simplified by the application. Whenever a modification of the catalog is required the appropriate personnel logs in to the system and with the help of the interface performs the changes.

- Pricelist management. Two distinct problems are addressed. The first is to maintain the pricelist information of the products and the second to perform pricing of an order. Both issues are covered by the application.

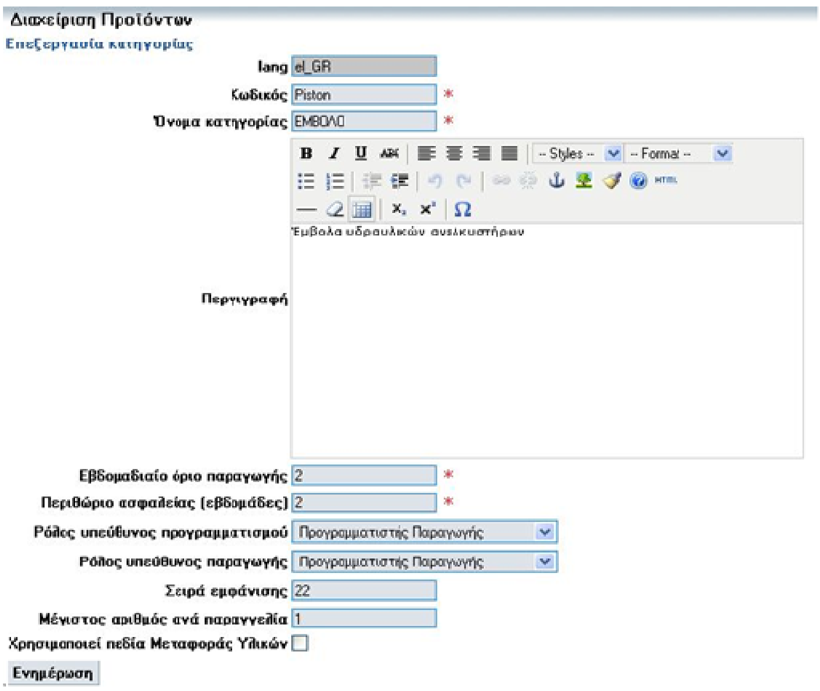

Fig. 1. Product Management.

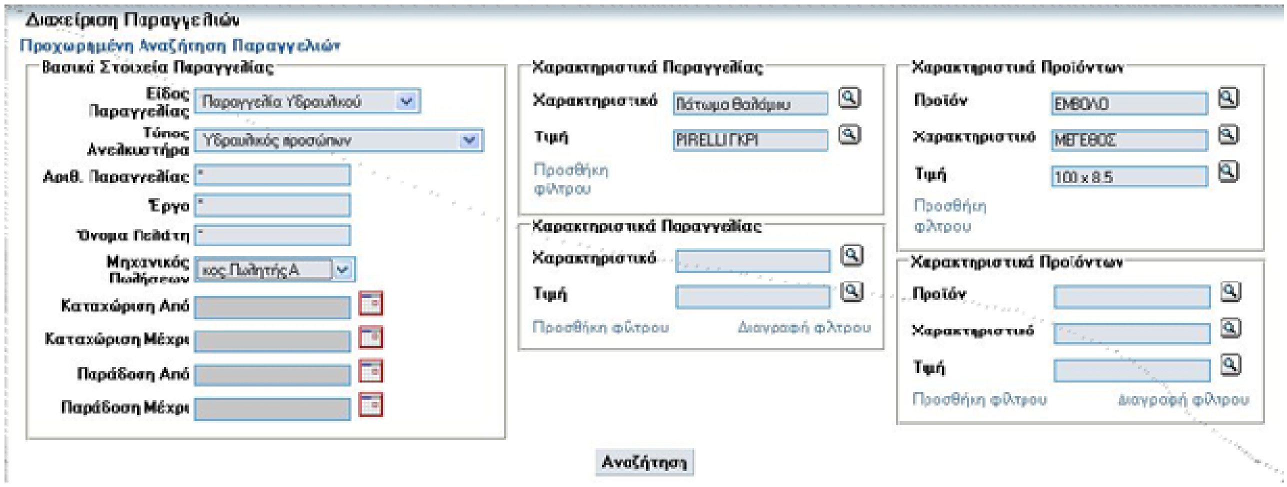

Fig. 2. Advanced search for orders.

- Document management. The whole procedure of an order may involve a significant amount of documents such as reports, faxes, and emails. Therefore, it is essential, for proper organization of the company, to manage those documents. Furthermore, it is necessary to be able to export the orders into a printable file format such as pdf. All those mentioned concerns are satisfied by the application.
- Customer Relationship Management. The application supports the notion of roles. A role, essentially, is an employee or customer class. For instance, “administrator” and “salesman” are two roles with different privileges. The administrator may introduce new rules while the salesman can only promote the status of an ordering.
- Integration with external systems. The application is designed in such a way that it is able to cooperate and integrate with 3rd party applications. It supports external document management systems, the Crystal Reports reporting system and a service that designs the actual lift mechanical design ready for production.

The system uses Java Server Pages (JSP) technology that provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server and platform independent.

Moreover, it uses JavaScript, which is a scripting language most used for client-side web development. Some indicative parts of the user interface are depicted in Figs. 1 and 2, related to product management and search for orders respectively. It should be noted that the user interface is mainly in Greek, but it can be converted in any other language.

4.3. Automated CAD Design

A problem that comes from providing an online point of sale for custom products is the lack of preview.

The client doesn’t know what he is purchasing, unless he meets with a sales representative who can explain the details. In this case however the whole process of online sales becomes meaningless.

In order to reduce the turnaround time it takes to provide the clients with designs that they can interact and provide feedback on, we created a component that can create accurate, print quality designs of the end product in the enterprise standard DWG format.

The Software platform on which the development was performed is the widely popular AutoCAD, which is not parametric, but the appropriate software (created from us) was added.

The software natively supports the ObjectARX API [7] which provides a detailed mapping of the inner workings of the platform. Applications that are created using this API are treated as points of extensibility or plug-in and can be invoked inside the user interface of the software. They are also expected to interact with the user in order to produce results. In order to allow the communication of the CAD software with the outside world we implemented a Remoting engine.

This is a commonly found design pattern in applications based on the .net framework and allows the remote invocation of commands. The communication is made over a private TCP channel, and is used for the transport of objects.

The engine is set to execute on the startup of the CAD software and listens for commands from other applications. When a connection is made, the calling application passes the necessary arguments and waits for the output file.

The design process is parametric. This means that the final drawing is a product of constraint resolution to the input arguments.

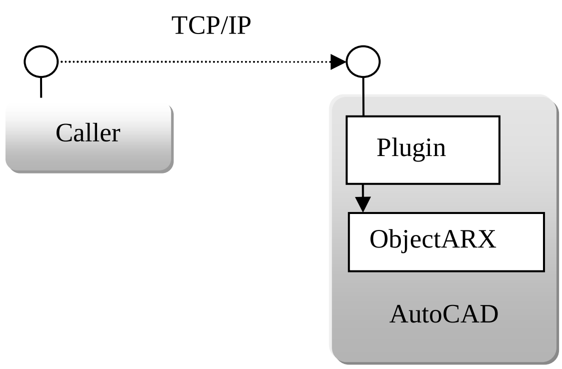

Fig. 3. End result of an elevator design.

These can be either dimension of the design or reusable design blocks as defined either by the needs of the structure or the requirements set by the client. In Fig. 3 we can see the end result of an elevator design.

Unfortunately the CAD software available today is not created with the intent to scale on the server side, and does not meet the necessary requirements for this task.

The component we created is expected to run for very large spans of time, with provisions for Quality of Service, verification and granular tracking of the actions it performs.

Issues that arise as part of this relationship include failures on the part of the CAD software, lacks of detailed API that can be used to track actions and internal or suppressed failures, and lack of support for externally controlled programmability.

To overcome this issue and enable the application to scale arbitrarily and independently of future context we wrapped it as a Web Service.

An issue that is immediately apparent is that of timeouts. Most major WS application stacks cannot issue synchronous requests and wait for the results for more than a small amount of time.

To overcome this each request, upon placement, is assigned a token which can be then used to poll on the progress or failure of the process. Some Quality of Service calls are also provided including estimated time left.

Eventually the caller may use it to acquire the result drawings. Table 1 synopsizes the most important calls, while Fig. 4 displays a common use case for the WS.

Due to the limitations of the CAD software working as a server, the majority of the application’s time is spent in waiting while non-concurrent processes are running. In order to maximize the availability of the component, we have encapsulated the CAD software in a single user interface, which is then wrapped by a queued WS. Each single user service is then registered in a pool and encapsulated by the queued service.

Another issue than was encountered is the need in part of the CAD software for administrative privileges on the host system, which would in turn require the same privileges for the web server that supports the WS. This is generally ill advised since it would allow for shatter attacks [8]. That means that security exploits that are found on the web server are always critical since they can perform anything an administrator can do.

An advantage of the architecture we propose is the gradual limiting of the privileges on the services, as the attack surface and exposure to the outside increases.

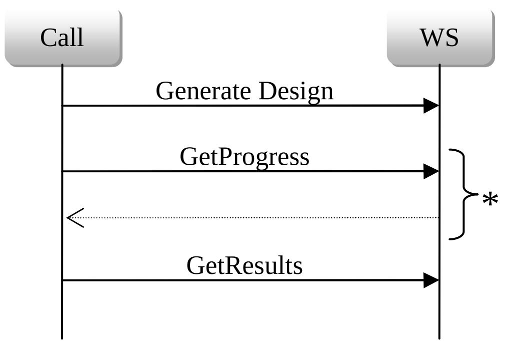

Fig. 4. Use case.

Table 1

Most important calls

GenerateDesign	Registers a design creation in the respective queue and returns a tracking token.
GetProgress	Monitors the progress of the request that has been assigned a token. Results can indicate waiting in queue, partial progress, completion or failure.
GetRequestsBefore	Returns the number of requests that are pending for completion before the request in question.
GetEstimatedTime	Returns time estimation about the completion of the request in question.
GetResults	Returns an address for a directory that contains the resulting files in various vector and raster formats.

4.4. Workflow orchestration

The workflow itself is orchestrated using the BPEL language [9] for web service orchestration. A synopsis of the workflow is presented in Fig. 5:

The user attempts to place an order, the system decides on the ramifications that his choices have to component selection and pricing.

Then the precise characteristics of the design are calculated and the drawing process is called. Finally the client receives the complete order with all the necessary details.

An advantage of the core components of the project being implemented or exposed as Web Services is the ability to repurpose each part of the system by applying workflow transformations.

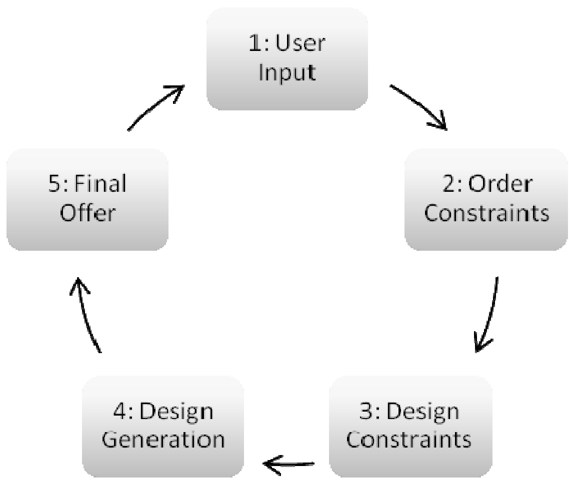

Fig. 5. Workflow orchestration.

These can affect the order and flow of steps taken to produce the outcome, and can be optionally extended using BPEL4People [10] to incorporate human interaction in the workflow as required by the business's future needs.

5. CONCLUSION

Given the increasing consumers' interest in the online product customization and ordering the construction companies invested in the development of online services that cover these needs. However, implementing such services impose certain challenges.

The current methodologies enable only limited product customization while the human factor is more than necessary to perform the ordering, design and pricing. In order to counter strike this situation we developed an architecture that enables fully automated product customization, ordering and pricing with minimal human intervention.

We successfully tested this approach in the real world by implementing it to support hydraulic elevators. Our efforts encourage the further extension of the existing system in various mechanical products.

REFERENCES

[1] J. Hyvrinen, Z. Turk, E. Balaton, C. Ebert, A. Gehre, P. Katranuschkov, A. S. Kazi, K. Kurowski, E. Petrinja, V. Stankovski, *State of the art and market watch report*, Interoperability of Virtual Organizations on a Complex Semantic Grid (InteliGrid), Project funded by the European Community under the "Information Society Technology" Programme Contract, IST-2004-004664.

[2] X. Chen Li and Shuming Gao, *A Web services based platform for exchange of procedural CAD models*, Computer supported cooperative work in design, 9th International Conference, CSCWD 2005 Coventry, UK, Vol. 1, May 24–26, 2005, pp. 605–610.

[3] M. P. Carroll and C. M. Hawkins, *Web based analysis*, ACM Symposium on Solid and Physical Modeling, 7th ACM symposium on Solid modeling and applications, ACM press, Saarbrücken, Germany, June 17–21, 2002, pp. 220–225.

[4] K. D. Bouzakis, A. Vakali, G. Andreadis and E. Karapidakis, *Manufacturing automation of a workpiece using XML*, 2nd International Conference on Manufacturing ENgineering (ICMEN), Kassandra Chalkidiki, October 5–7, 2005, pp. 741–750.

[5] K. D. Bouzakis, A. Vakali, G. Andreadis, *Development of a Web-based System Providing Communication between Designers and Manufacturers*, 2nd International Conference on Manufacturing ENgineering (ICMEN), Kassandra Chalkidiki, October 5–7, 2005, pp. 759–766.

[6] http://www.ilog.com/.

[7] Autodesk Inc., *ObjectARX Api Specification*, http://www.autodesk.com/objectarx , 2003.

[8] T. Close, A.H. Karp, M. Stiegler, *Shatter-proofing Windows*, HP Labs Whitepaper, 2005.

[9] BPEL Working body, *BPEL4WS 1.1 Specification*, IBM DeveloperWorks, 2003,

[10] Active Endpoints, Adobe, BEA, IBM, Oracle, SAP AG, *WS-BPEL for People Extension*, 2005.