



## A Web-enabled Configuration System for Interior Design

Maura Mengoni<sup>1</sup>, Damiano Raponi<sup>1</sup> and Roberto Raffaelli<sup>2</sup>

<sup>1</sup>Università Politecnica delle Marche, [m.mengoni@univpm.it](mailto:m.mengoni@univpm.it), [d.raponi@univpm.it](mailto:d.raponi@univpm.it)

<sup>2</sup>Università degli Studi eCampus, [roberto.raffaelli@unicampus.it](mailto:roberto.raffaelli@unicampus.it)

### ABSTRACT

Web-enabling technologies represent the next generation of design environments to design and manufacture complex systems, such as them characterizing contract furniture. In the context of web applications to facilitate and support teamwork in collaborative product development, the paper presents a CAD-based infrastructure for the 3D visualization of co-designed solutions, the on-line customization of furniture items and the creation of a shared relational database of products, architectural scenes and knowledge-based rules guiding configuration. A double-level geometry is presented to manage 3D web representation and product structure. A use case is adopted to show main platform functionalities and possible advantages for the extensible contract furniture cluster.

**Keywords:** web-enabling technologies, collaborative product development, CAD, WebGL

### 1. INTRODUCTION

Nowadays, contract furniture (CF) represents a promising opportunity for SMEs to be competitive in the worldwide market. Contract furniture is a transversal business which aims to provide finished commodities for hospitality, retailing, as well as for stores, offices and restaurants.

The process of CF design requires the involvement of different actors (product designer, architects, supplier, etc.) that are often distributed worldwide and work on the same project to achieve a shared goal. An high level of collaboration and synchronization is required, from the integration of single “pieces of work” which are individually developed (i.e. tasks, decisions) to the combination of different working actions due to the actors’ working way (i.e. work at computers, talk to other designers or specialist, solve problems by acting on the product models) [4]. The different social background, knowledge and skills of the involved stakeholders trigger research toward the creation of adequate supporting tools for proper data visualization and management.

A previous research work [13] has outlined the main characteristics of contract furniture design and the main problems to manage such extensible and temporary cluster of involved companies, different in size, adopted information technologies, developed

products, project goals, skill, etc. Two main open issues has emerged:

- no Computer Supported Cooperative Work system is dedicated to this complex application area, whose needs regard the simultaneous support of information exchange along the furniture life cycle, the management of multiple and conflicting design constraints and process requirements and finally the creation and sharing of the whole furnishing Bill of Materials [7,8];
- the management of an high level of furnishing customization is complex as personalization depends on numerous requirements: from the needs of the target market, the standards of the country, the customer profiles, the contract typology, the design style, the target cost to the expected performance, etc. It is much more than a simple variation of the item dimension as it is extended to finishing, functions, aesthetics, etc. In addition it does not regard the single furniture item, but the whole space. As a consequence a CAD-based solution must manage configuration at different levels of abstraction. A browser-based tool could enable users to visualize and interact with complex 3D configuration [14].

Contract furniture design lacks of integrated, flexible and cost-effective supporting solutions. The Web is surely the most powerful tool to provide a light-weight and an operation system-independent platform for users to search, browse, retrieve, share and manipulate information in the context of the CF design. Most research works propose a Collaborative Product Data Model (CPDM) and a constraint-based Collaborative Design Process Model (CDPM) to facilitate the management and coordination of the collaborative design process as well as design knowledge management [20]. In other works [5-21] the Internet is leveraged to build an environment that connects distributed CAD software clients to a centralized product data server for collaborative design. Some technological difficulties emerge and require to be overcome. Firstly, large CAD files require an unbearable down-loading time over the Internet. Besides, parts are usually modeled with different software systems or even with different versions. The visualization of the CAD model would not be possible if there were no general standards for 3D graphical representation and effective transmission strategies over the networks.

In conclusion, no solution offers an integrated environment for collaborative data management, product modeling and customization, review and decision-making [13]. In this context the present research work proposes a platform which consists of four main user interfaces: one dedicated to the online product configuration and 3D visualization of the custom solution, one to the selection of the personalized items and their integration into a 3D architectural space, one to manufacturing companies to define models and design contractions and one to support the co-design of integral solutions.

The paper mainly focuses on the description of the web-enabled geometry representation and configuration system that allows 3D CAD models of architectural spaces and furniture items to be customized and shared by designers and manufacturers within the network of companies. Configuration is based on a set of knowledge-based rules that guarantee the coherence of the designed solutions against manufacturing and installation constraints.

Therefore, the main contributions are: the developed 3D-web based visualization of integral solutions and single items; the adopted infrastructure to exchange and synchronize data among the different desktop and internet based applications; the structure of the virtual prototype as a compendium of both geometric, functional, technological features for enabling companies to contribute with their specific knowledge; the developed graphic rendering algorithms which are homogenous for web and CAD-based tools; and, finally, the modality to guarantee the coherence of information flowing across the system modules by sharing models and data structures.

After the description of the architecture of the developed configuration system, a test case is

implemented to demonstrate the advantages for the final users. In fact, several stakeholders are involved, from the architect which combines design products to achieve environments with higher aesthetic value, to the company which can promote a better contextualization of the own products. Finally, the end user is advantaged by knowing about new "Made in Italy" products and modern living environments.

## 2. BACKGROUND

### 2.1. 3D Model Sharing on the Web

In order to cope with the requirements of collaborative systems on webs with limited bandwidth capability, the research community has initially investigated in innovative light-weight 3D standards and 3D streaming communications [7]. As a result some dedicated formats, such as VRML, X3D (eXtensible 3D), W3D (Web 3D) and MPEG-4 were launched.

The VRML files describe the 3D scene in terms of geometric objects, triangular meshes, operations and properties of the scene. It has the advantage of being written in text format. The major disadvantage of VRML is the dimension of the files since no compression has been built in. The download and display speeds are not satisfied.

X3D is a major upgrade from VRML aiming to more light-weight representation. It is being developed under the Web3D Consortium's ([www.web3d.org](http://www.web3d.org)) standardization process. Nodes in X3D are represented in XML tags so as to take full advantages and potentials of XML on the Internet. Moreover X3D incorporates numerous 3D techniques including advanced rendering and multi-texturing, NURBS surfaces and IEEE Distributed Interactive Simulation (DIS) networking.

MPEG-4 has been proposed to define a binary compression format to encode VRML in binary representation. As the major objective of adopting MPEG-4 is for multi-media applications, MPEG-4 supports media mixing and audio composition, and it can easily mix with rich forms of media, video and audio for multi-media collaborations in a Web-enabled environment.

Behr et al. [4] provided an overview of the available 3D web technologies with a division in plug-in, i.e. Java3D, O3D, and plug-in-less solutions, i.e. Canvas3D or other solutions that use 2D browser tools to simulate a 3D pipeline. VRML and X3D languages are actually developed to describe 3D scenes in terms of geometry, material and illumination, while their rendering inside a web browser requires the installation of specific plug-ins or applets. Examples are the Active X technology which operates on Microsoft's web browser and operating system, or other components, such as a Java Applets which run on different platforms [11].

Experiences of architectures to share 3D contents via web browser can be found in Visintini et al. [18] and in OSG4Web [3, 15]. The first consists in a browser

plug-in that uses an OpenSceneGraph library that support Level-of-Detail (LOD) objects and can load different 3D model formats. Among them, Java3D is a well-known high-level programming API for visualizing and manipulating 3D models. Java3D has rich set of 2D and 3D objects and behaviors and it encompasses 3D geometry compression and LOD. A binary geometry compression format is utilized both as a run-time in-memory format, as well as a storage and network format. Finally Java3D supports a wide variety of file formats to accommodate many vendor-specific CAD formats, interchange formats and VRML.

These latter peculiarities are particularly useful for large scale visualizations. The drawback of these purpose developed technologies is the need to install external components necessary for its functionality. To solve this issue, the WebGL approach was developed and it is expected to grow considerably in the future [6, 10]. WebGL is an API developed to extend the capability of JavaScript language in order to control the graphic card from the most widespread browsers (Firefox, Chrome, Safari). It is based on the OpenGL ES 2.0 standard to be supported also by embedded and portable devices.

Actually many graphics libraries rely on WebGL. For instance, Google O3D is a graphic API expressly designed to create interactive 3D application within a browser [12]. It was conceived as a browser plug-in, but one year later it was turned into a JavaScript library based on WebGL.

## 2.2. Configuration Systems in the Furniture Design

Generally speaking, given a set of components with certain properties and ports connecting them, configuration can be defined as the activity of finding a solution in combining such components while fulfilling a set of constraints [9, 17]. The set of constraints that restricts product variants is related to technical limitations, economic factors and production processes [16]. Narrowing the scope to the CF design, configuration systems are intended as proprietary or open-source platforms, which allow 2D and 3D environments to be created and furniture items positioned and rendered. They need to be easy-to-use, intuitive and low cost.

Web-enabled configuration systems allow data to be easily shared through the web. The user imports 3D model from some shared libraries. Some examples of existing systems are SweetHome 3D (<http://www.sweethome3d.com/>) and DomusPlanner (<http://www.domusplanner.com/>). However, such systems poorly support the technical configuration of the products, its structure, rules and constraints, so that they are suitable for the final customer but not expert users such as architects or manufacturers.

Other CAD-based configuration systems include dedicated commercial configuration systems or

open-source platforms. They implement functionalities to handle CAD models, configure them in a certain environment and to generate complete BOM. They adopt a single company perspective, so they cannot support collaboration and co-design within an extended network. Some examples are Metron (<http://tesysoftware.net/>), 3CAD evolution (<http://www.3cadevolution.it>) and Mobilia (<http://mobiliasoftware.com/>).

An comprehensive analysis of the configuration systems lead to include the BIM (Building Information Modeling) technology, which include common commercial systems such as ArchiCAD (<http://www.graphisoft.com/archicad/>), Revit (<http://www.autodesk.com>) and Allplan (<http://www.nemetschek-allplan.eu>) [1]. They implement functions to generate a CAD model of an architectural building characterized by its geometry, spatial relationships, information about materials and building elements, costs and project schedule [2]. They support the interior design but cannot support the extensive collaboration among several companies and stakeholders.

In conclusion, the analysis of the available technologies has highlighted the lack of solutions dedicated to the contract furniture. In particular, web-enabled configuration platforms are desirable to manage technical features, product variants, the whole environment configuration, rules, real-time collaboration and high-quality rendering. A knowledge approach leverages the development of products which can be configured in several variants. Built-in rules allow the designer to create configurations which respect the user needs, permit multiple alternatives to be quickly tested, and avoid design errors caused by the lack of technical knowledge.

## 3. APPROACH AND PLATFORM DESCRIPTION

This section presents a CAD-Based infrastructure to support the 3D visualization and configuration of furniture items, the creation of a shared relational database of products and integral solutions, the definition of design rules and, finally, the configuration of the whole architectural spaces by selecting and positioning furniture items according to the pre-defined rules. Before detailing the proposed system architecture, the adopted product structure and the geometrical representations necessary to meet the CF requirements are described.

### 3.1. Product Structure and Geometry Representation

An original representation is introduced to manage the geometry, the appearance and the configuration rules of the products. Fig. 1 depicts the main elements of the representation.

Product is here intended in a broad and transverse sense, including an entire environment, a group of

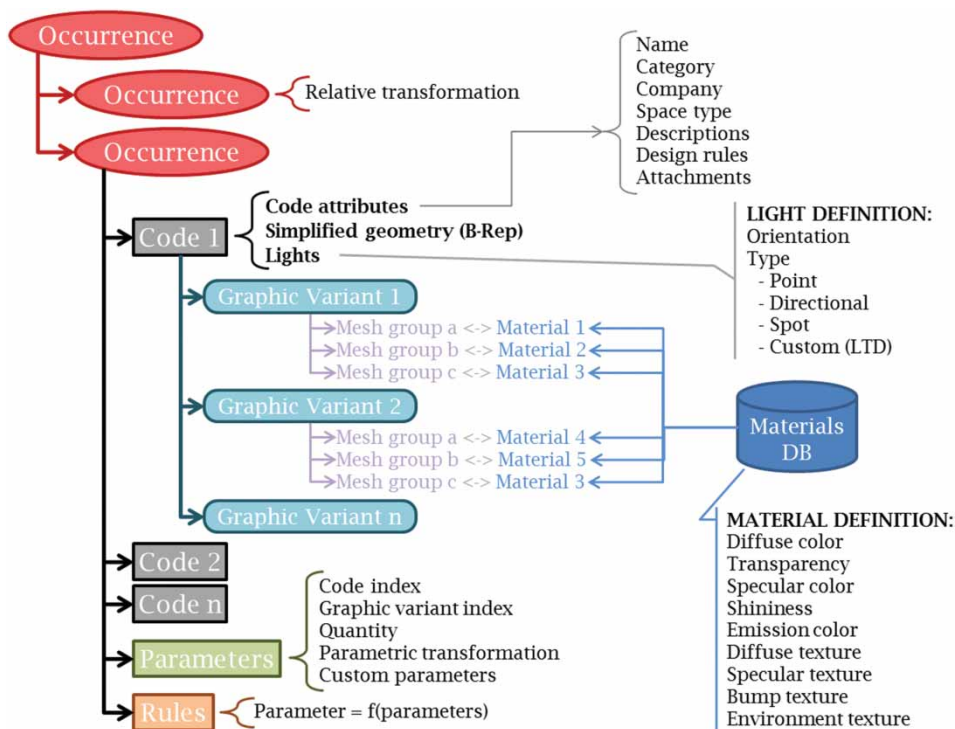


Fig. 1: Representation of the product: structure, variants, graphics, rules.

products (e.g. kitchen), single product instances (e.g. dishwasher) and product inner structures (e.g. drawers of a cabinet). The product is captured by the concept of *occurrence*. An occurrence defines the relative orientations of its child occurrences. The whole ambient is therefore a hierarchic structure constituted by a tree of occurrences. Such hierarchy allows pieces of configuration to be realized separately, combined and reused.

Each occurrence contains the definition of a product through a list of codes. A *code* is an instance of a physical product as it appears in the company repositories (e.g. Enterprise Resource Planning or Product Lifecycle Management systems). It includes the item geometrical representation, its non-geometric attributes and embedded light definitions. The different codes act as variants of the occurrence. Only one code is active at a time.

The geometry of a code is defined on two levels. The first one is a render mesh whose vertices include

texture mapping coordinates. The second one is a hidden simplified B-Rep analytic representation to be used as the basis for geometric functionalities such as mating, part measuring and positioning. Fig. 2 shows the two representations in case of a bathroom sink. A mesh capturing the details of the bowl and the aesthetic features provides the render geometry. A simplified geometry consists of boxes corresponding to the sink overall dimensions and a hole as a reference for the tap. The advantage of such duality is to have the possibility to implement low-cost functionalities for precise part positioning, mating and measuring of the distances.

The memory request due to the use of a simplified geometry in addition to the rendering model is negligible. In fact, the bounding box or similar elementary shapes are suitable for the majority of the cases to provide sufficient geometric elements, such as faces, edges and vertices, to mate and measure the developed composition. The B-Rep representation of

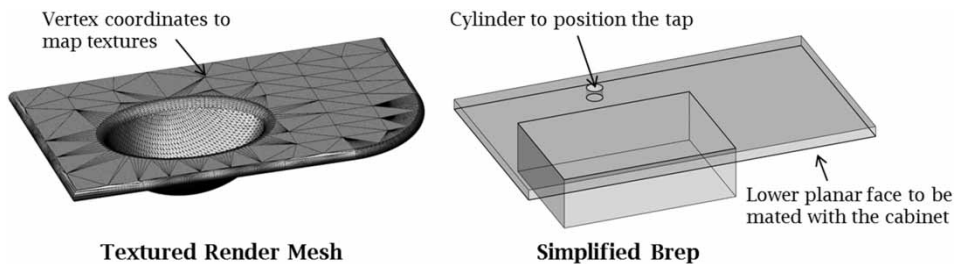


Fig. 2: Example of geometric representation of a bathroom sink.



such simple shapes is very light and does not waste resources of the computer, realizing an acceptable compromise.

The geometry is input through Step files for the B-Rep geometry and Vrml files for the render mesh. The two formats have been chosen for being diffused and flexible. In fact, step files contain the definition of solid geometry in terms of geometric and topological data. They can be created with the majority of commercial CAD systems. On the other hand, the rendering model requires meshes, texture mapping, material and lights definition. Vrml files include such data which can be in such a way transferred from other software system. However, the proposed product definition includes more extended characterization of the rendering properties, such as custom lights, bump mapping, shadows, etc . . . . Such attributes are defined by the developed user interfaces and saved in a proprietary format.

As defining a new product, three cases are distinguished:

- **Vrml + Step files:** The Vrml contains the mesh definition, texture mapping, material attributes and the textures images. The Step provides for the simplified geometry. The bounding boxes of the two representations are checked to be coherent each other. The Vrml is created by a system which has functionalities for advanced material definition and texture mapping, for instance 3DS Max by Autodesk.
- **Only Vrml file:** in this case the simplified B-Rep is computed from the bounding box of the mesh.
- **Only Step file:** in this case it is assumed that the geometry is detailed, for instance it comes from a CAD model of the design department. Functionalities are provided to compute the mesh and the texture mapping from the B-Rep. The B-Rep is then substituted by its bounding box.

### 3.2. Configuration Rules

The product exhibits several degrees of configuration. The first one is given by the codes which basically represent different geometric solutions. Secondly, each product variant may then have in turn several *graphic variants*, e.g. different materials combinations to be applied on the same geometry. The set of meshes which are used to render the object is split in *groups*, i.e. portions of the geometry with the same material definition. A graphic variant maps material definitions to each mesh group. Finally, the geometry can be modified by anisotropic scaling.

A set of parameters is introduced to control the status of the occurrence, such as the current code, the status of the geometry and the appearance. A parameter is a variable which can assume numeric or textual

values in a certain domain. Rules operate on these parameters to implement configuration logics. Parameters can be freely defined by the user. Some of them are reserved and created by the system, such as:

- **Variant:** an integer controlling the active code index;
- **GraphicVariant:** an integer controlling the active graphic variant index;
- **Index:** an incremental read-only index to distinguish repetitions of the same code in the whole configuration;
- **Quantity:** an integer value causing copies of the occurrence to appear in the configuration. If the value is zero the occurrence is eliminated from the configuration;
- **DeltaX, DeltaY, DeltaZ, RotX, RotY, RotZ:** numeric values to transform the occurrence by a parametric roto-translation;
- **ScaleX, ScaleY, ScaleZ:** parameters controlling the anisotropic scaling of the occurrence;
- **Width, Depth, Height:** Dimensions of the bounding box of the code geometry. Such values are read-only.

Rules are defined as expressions in the following form:

$$P_i = f(P_0, P_1, \dots, P_j, \dots, P_N) \text{ where } j \neq i \quad (3.1)$$

Every parameter is computed as function of the others as in equation 3.1. The adopted syntax is standard and recalls MS Excel formulas (examples in equation 3.2). An exclamation mark is used to separate an occurrence name from the parameter name. The inference engine is able to detect parameters dependencies and rearranges the order of the rules to solve them.

$$\text{Handle!Quantity} = \text{SELECT}(\text{Cabinet!Variant}; 2; 3; 4; 1; 0)$$

$$\text{Handle!Spacing} = \text{SELECT}(\text{Cabinet!Variant}; 360; 180; 180; 0)$$

$$\text{Handle!DeltaZ} = -\text{Handle!Spacing} * \text{Handle!Index} \quad (3.2)$$

Each parameter can be exposed thanks to a specific flag. If exposed, a parameter becomes usable as configuration input option. The user can change the value to obtain different product definition thanks the recalculation of the rules and consequent updating of the other parameters.

Finally, rendering materials and lights are defined with a good level of complexity. The material includes standard parameters such as diffuse color, specular color, transparency, etc . . . , and maps for diffuseness, transparency, reflectiveness and bumping. Light description comprises ambient, directional, spot lights options but also custom illumination diagrams from ELUMDAT files (\*.ltd) [19]. Lights can be defined in the environment or be geometrically associated to products such as lamps.

### 3.3. Real Time Rendering

The scene rendering relies on the parallel computing capabilities of the PC graphic cards (GPU). The requirements for high quality rendering, such as per-pixel shade computation, shadows projection, bump mapping, custom lights definition, is achieved thanks to the programming capabilities of the GPU.

The rendering behavior of a graphic card is controllable through the Shading Languages, which allow developers to customize the fixed rendering pipeline thanks to some shading algorithms, i.e. *Vertex Shader* and *Fragment Shader*. In the present work, the OpenGL Shading Language (GLSL) [10] is exploited to reach the desired high quality rendering effects on both desktop applications and web pages.

A Vertex Shader is a piece of code which runs for every vertex given to the GPU. The purpose is to transform each vertex's 3D position in virtual space to the 2D coordinate at which it appears on the screen also computing the depth value for the Z-buffer. Fig. 3

```
void main(void) {
    mat4 modelView = MVMatrix * XformMatrix;
    vec4 ecPosition = modelView * MCVertex;
    gl_Position = PRMatrix * ecPosition;
    diffuseTextureCoord = TexCoord0;

    if (EnableLighting)
    {
        ecPosition3 = (vec3(ecPosition)) / ecPosition.w;
        ecNormal = vec3(modelView * vec4(MCNormal, 0.0));
        if (EnableNormalize)
            ecNormal = normalize(ecNormal);
    }
}
```

Fig. 3: Example of GLSL code from a Vertex Shader.

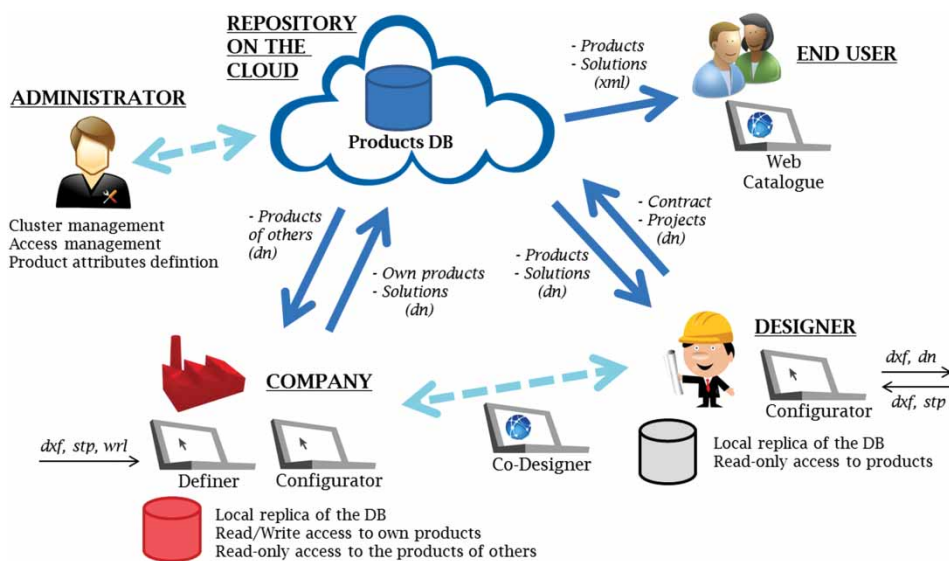


Fig. 4: System architecture, user roles and types of exchanged data.

reports an example of GLSL code for a Vertex Shader. On the contrary, the Fragment Shaders runs for every pixel on the screen and basically compute color and other attributes.

The adopted GLSL code can be used on both desktop and web applications. In fact, the OpenGL APIs allows activating interactions between the CPU, where geometric data are managed, and the GPU. On a web page, the advent of the WebGL and HTML5 technologies has given the possibility of controlling the GPU through the JavaScript [14].

### 3.4. System Architecture

A major contribution in the field of the furniture configuration is given by the proposed architecture for the distributed model sharing (Fig. 4). The products are stored on a remote repository in a server on the cloud. Information retrieved in the system database flows across four different platform modules mostly

through a proprietary file format (\*.dn) which embed the described representation scheme.

Data sharing modalities, i.e. type of information and access, are subjected to the specific user role as follows:

- **Administrator:** it has full access to the system and the responsibility of the managements of the shared data. Such role is in charge of the cluster leader company;
- **Company:** it loads its own products to the server by the Definer module. It has also read-only access to other company products and it can use them for sample compositions. A local replica of the DB is provided to allow off-line operability. Synchronization is allowed once the user connects to the Internet and logs with its specific role;
- **Designer:** it refers to architects and other specialists which use the platform (Web Catalog, Configurator and Co-Designer modules) to design contract environments. The designer role allows products to be downloaded to a local DB and the configured solutions to be updated and shared;
- **End user:** a generic user which browse the Internet by the Web Catalog module searching for furniture solutions.

More specifically the features of the system implementation and the user interfaces of the software modules are here described:

- **Web Catalog:** a web-based virtual marketplace promoting in an appealing way the products offered by different manufacturers. The software allows user identification, authentication and tracking to achieve a complete profile. Once logged, the user can view a rich catalog of products and assess all predefined variants by a high quality rendering. The catalog is written in ASP.NET code to produce dynamic Web pages, applications and services and adopts WebGL for rendering interactive 2D and 3D graphics of product models, scene and solutions within any compatible web browser;
- **Configurator:** a configuration tool able to support the 3D configuration of both products and architectural space according to a set of knowledge-based rules and best practices. Items are positioned in the environment from the Virtual Catalog by following manufacturing guidelines, technical constraints and geometrical relationships among objects in space. The system is interfaced and synchronized with the Virtual Catalog to keep the selected products and solutions ever updated. The Configurator is implemented on the .NET Framework adopting object-oriented computer languages and exploits the

programming capabilities of the graphic card through the shading language to realize 2D and 3D photorealistic rendering.

- **Definer:** a management tool to enable companies uploading products and solutions into the catalog, preset the product variants (e.g. colors, finishing, functions, features) and add technical documentation suited for the different furniture stakeholders (e.g. contractors, designers, architects and end-users). For each item the company provides the 3D model, indicate existing or customizable characteristics, specify possible accessories or auxiliary functions, define customizable features and allowed range of modification, add installation and configuration constraints and finally attach technical information such as manuals and data sheets. The module is developed by the same implementing technology of the configuration system and is synchronized with the platform-shared database.
- **Co-Designer:** a collaborative design tool to allow designers to interact with company technicians in case some special product requirements are needed.

From a technical point of view, the synchronization among the server, the definer and the configurator systems is allowed by remote connection to database through the OLE DB standard while files are transferred through the FTP protocol. The communication between the server and the Web Catalog is based on the standard HTTP protocol. The geometric data are transmitted from the server to the client side in XML format, parsed by JavaScript code and temporary stored by the web page DOM.

#### 4. AN APPLICATION TEST CASE: THE SERVICED APARTMENT

The described platform is completely developed and tested thanks to the DesignNET initiative, an Italian project promoting innovation and Made in Italy lifestyle in the context of companies operating in the CF sector.

The application of the proposed platform to a developed test case is described in this section. It concerns an application in a typical example of CF, i.e. a serviced apartment. It is a type of furnished apartment available for short-term or long-term stays, which provides amenities for daily use.

Ten SMEs were involved in the project which was led by a designer who developed the concept and coordinated the work of the companies. The apartment is composed of four main areas: the kitchen, the living room, the bedroom and the bathroom as shown in Fig. 5. In these areas companies have worked together to integrate their products considering functional, aesthetic and dimensional constraints.



Fig. 5: Geometric model of the studied serviced apartment.

#### 4.1. Products Definition through the Definer Software

As a first step, companies uploaded some items into the platform thanks to the Definer tool. The software was installed as a desktop application in each company that used it to define the products and populate a local DB.

A new product can be created with or without geometry providing a unique name and identifier. The choice of creating a product without geometry is to group more articles under a wider definition. The geometry is loaded from CAD models in STEP and/or in VRML formats as described above. Product is then populated with attributes, variants, different dimensions, components and colors.

For example, a cabinet has two or more drawers with different handle options and is equipped with oven, hob, and combine various finishes. Product attributes are given by: name, space type, company name, category, line, code, description, notes, attached technical documents, preview, market target, contract type, pricing, product type and the visibility (Fig. 6). The visibility attribute set the availability of the product across the platform, i.e. the Web Catalog, the Configurator, both or none. Also some tags are added to classify the product on the basis of specific design features such as eco-sustainability, hygiene, multi-sensoriality and safety.

The product instance is enriched with new codes, i.e. variants, or with occurrences, i.e. child parts. Variants and child occurrences are defined from scratch or are reused from other products. Graphic variants are then associated to codes.

For instance, five codes and five occurrences are defined for the cabinet. Codes activation and occurrences visibility, as well as dimensions and orientation in the space, are controlled by the parameters and rules which determine valid combinations.

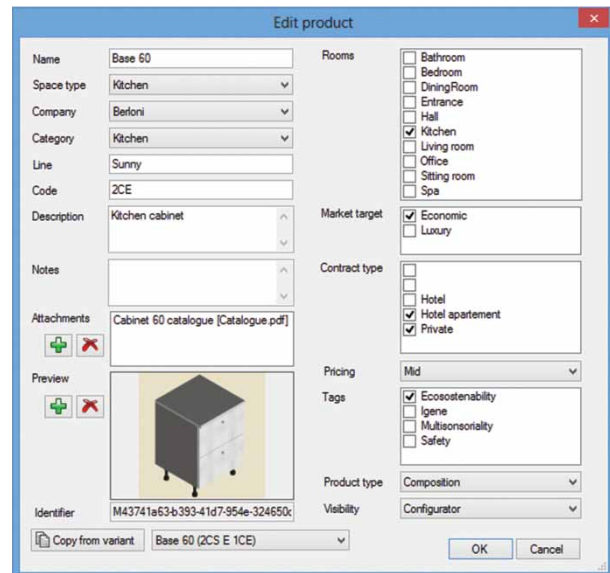


Fig. 6: Non geometrical attributes of a product definition in the Definer software.

Additional parameters are added to fulfill special need of customization. Some parameters are then chosen to be exposed to the user of the Web Catalog and/or the Configurator.

In particular, for the cabinet test case four custom parameters are added in the parent occurrence: one for the type of handle, one for the wood color, one for the finishing of the work plane and one for the type of the hob. Another parameter is added to the handle child occurrence to set the spacing between the handles.

At the end, eighteen color combinations, eight types of handle, ten finishes of the work plane and two types of hob are defined (Fig. 7). Nine rules are



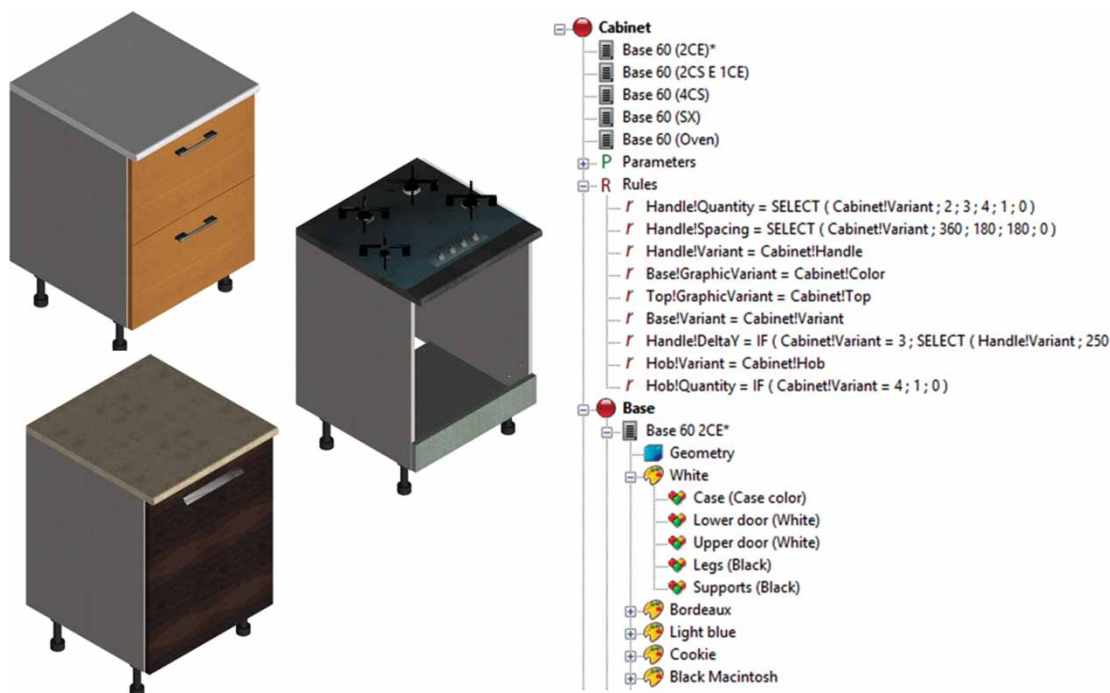


Fig. 7: Variants of the cabinet (left) and tree of configurations from the Definer software (right).

input to manage the quantity and the spacing among handles, the code of the active handle, the graphic variant of the base, the graphic variants of the work plane, the presence and the variant of the hob.

The product definition is then completed with the installation constraints. They relate to the maximum distance to the power socket and the request for contact to the wall.

Once ended the product definitions, the company can log it into the server and update the server DB to share the documents across all applications.

#### 4.2. Product Sharing through the Web Catalog

As new products are uploaded on the server, they are accessible from the Web Catalog website. Products are classified according to categories, i.e. kitchen, bath, living, wellness. The catalog has free access to allow products to be publicized. The private user can browse the pages, view the 3D rendering, change the configurations, download documentation and contact the producer by email. A search function is also provided on the basis of criteria such as company name, contract type, pricing.

The members of the DesignNET network, i.e. companies and architects, have a special login to the Web Catalog. They can open a workspace where personal data are organized in two main containers: “my projects” and “my collaborative sessions”. The first one contains the developed solutions uploaded to the server, while the second one traces the co-design sessions held with the company technicians

to solve specific design problems. Collaborative sessions can be triggered on a specific product searched in the catalog or from the items used for a developed solution.

#### 4.3. Products Usage in the Configurator Software

The Configurator is used by an architect or a designer to create a configuration with a wizard composed by five main steps. The Configurator makes use of the products downloaded from the server after synchronization with the local DB. The configurator provides read-only access to products so they can be used and modified in the limits fixed by the companies in the previous definition phase.

The tool is here described taking the Serviced Apartment as example. A new configuration starts with the definition of some project attributes such as: the name of the project, the type of the environment (i.e. hotel apartment, commercial space, etc ...), the target market, the location, the pricing, the dimensions of the environment, the number of the rooms. The geometry of the environment is loaded from a STEP file.

In the second step the designer details the model of the Serviced-Apartment adding some tags to detect the rooms. He also input key-points for indicating the position of windows, doors, electrical socket and drainages. In the third step the designer places items in the environment. Drag and drop functionality are provided to easy the task of choosing an item from

the available product tree and put it in the graphical area. The item can then be configured by choosing suitable values from the exposed parameters: variant, dimension, color and other options.

The system allows the products to be exactly positioned in several ways, such as snapping, mating and alignment of faces and edges, or by imputing precise translation and rotations. Measurement and alternative rendering options are available to ease the positioning task. Items are positioned in the environment

by following manufacturing guidelines, technical constraints and geometrical relationships among objects in space. Referring to the cabinet case, the designer selects the variant, i.e. with two drawers or with oven and hob, the color of the wooden doors, the type of the handles, the finishing of the work plane and the type of the hob. When selecting the variant with oven and hob, the system alerts the user with the design guideline of positioning the product close to an electrical socket.

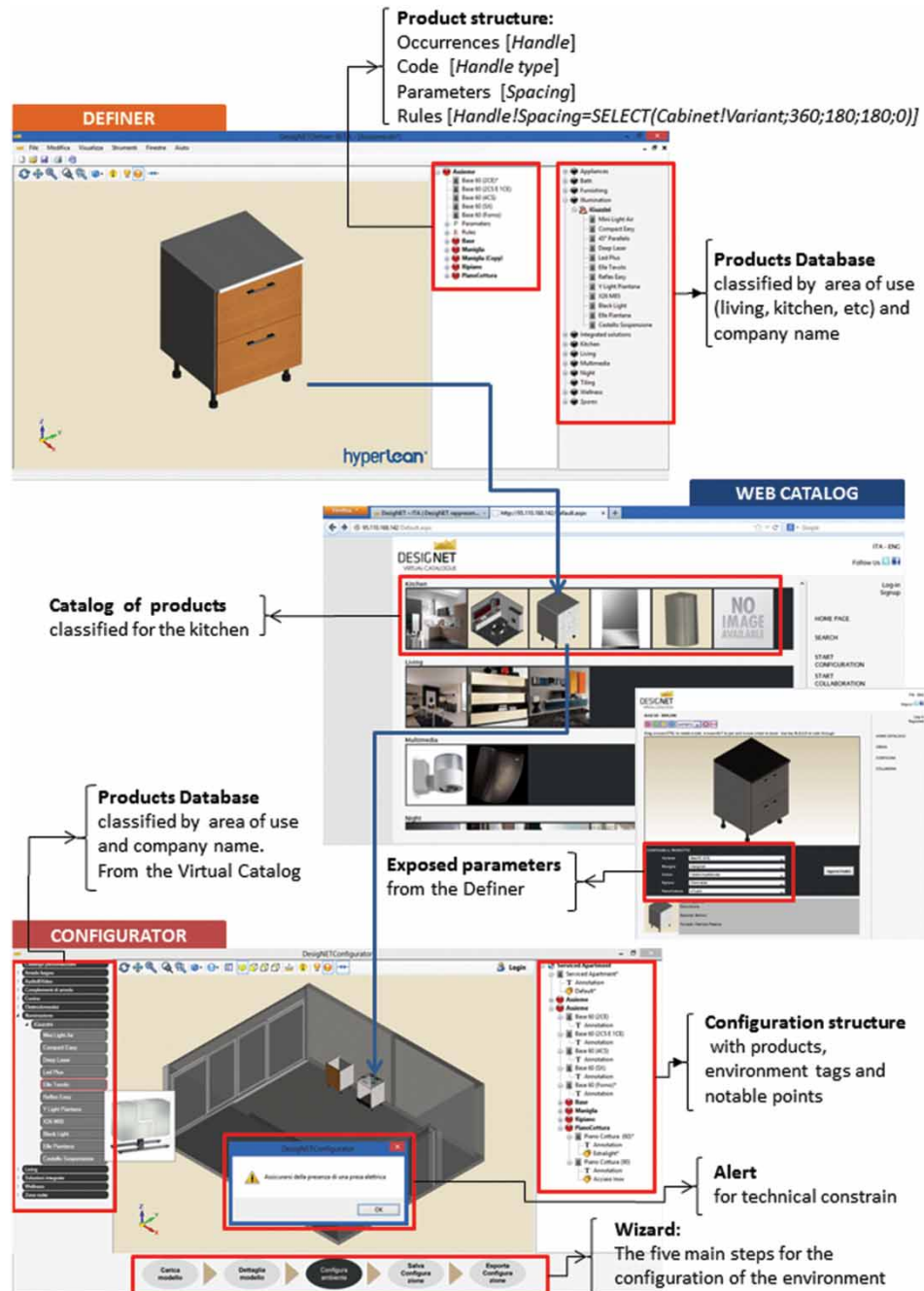


Fig. 8: The test case, from the product definition to the configuration of a Serviced-Apartment.

The fourth step takes care of saving the work locally and in the remote server. If saved in the cloud, the configuration is available in the workspace into the Web Catalog for sharing and/or launching web collaborative sessions.

The fifth and last step provides tools to export configuration data. Six options are available:

- export the 3D configured environment to a VRML file
- export the plant of the environment to a 2D DXF file
- generate a BOM of the products in the environment
- generate a printable report with the data of the project, the data of the products and the installation guidelines
- export the data sheet and the other attachments of the products
- saving the 3D model of the products in STEP or VRML format

The exported data is useful to share the configuration among the stakeholders and to provide technical information to trigger the procurement or production of the selected products. Fig. 8

#### 4.4. Results

System experimentation is still at a preliminary level due to the large amount of work to be done to populate DB and formalize the implicit knowledge of all involved stakeholders in implemented rules.

The project of a Serviced-Apartment was really useful to identify the main system potentialities and the drawbacks to be overcome. Users' feedback is promising. All actors, from manufacturers to designers recognized the importance of the platform to share project evolution and items customization from the preliminary concept development. The integration with company ERP and the interoperability with most of the adopted CAD increase system acceptability and easiness-to-upload items, define possible variants, create custom solutions and integrate products from different manufactures with a set of shared technical and geometrical constraints, The graphic rendering of the virtual Catalog allows companies to present their offers in an amazing and attracting way and final users to intuitively assess their choices.

The adoption of two levels for geometry representation, in the Definer and in the Configurator, results to be useful to switch between a render mesh to view the products in a realistic image and a B-Rep representation necessary to position parts and other geometrical functionalities. Data exchange and synchronization latency and hence system efficiency appear to be proper enough for the applications identified in contract furniture. The configuration

rules of the products are a key aspect to guarantee the technical validity of the developed solutions and make possible to integrate items from different manufactures without design errors.

#### 5. CONCLUSIONS

A CAD-based configuration system exploiting web-enabling capabilities is presented. Implementation results are described in detail focusing on how the system is able to share and synchronize information among different platform modules and how data coming from different sources is kept updated.

The case study for experimentation is the design of a Serviced-Apartment. It has demonstrated the system functionalities and the flow of work across the stakeholders. The preliminary results outline the usability of the system and the achieved effectiveness in terms of tool flexibility, time to market reduction and capability to manage design conflicts. All these parameters need to be objectively measured by collecting more use cases.

#### REFERENCES

- [1] Arayici, Y.; Coates, P.; Koskela, L.; Kagioglou, M.; Usher, C.; O'Reilly, K.: Technology adoption in the BIM implementation for lean architecture practice, *Automation in Construction*, 20, 2011, 189-195. <http://dx.doi.org/10.1016/j.autcon.2010.09.016>
- [2] Azhar, L.; Hein, M.; Sketo, B.: Building Information Modeling (BIM): Benefits, Risks and Challenges, Leadership and Management in Engineering, 11(3), 2011, 241-252. [http://dx.doi.org/10.1061/\(ASCE\)JM.1943-5630.0000127](http://dx.doi.org/10.1061/(ASCE)JM.1943-5630.0000127)
- [3] Baldissini, S.; Manferdini, A. M.; Masci, A.: An information system for the integration, management and visualization of 3D reality based archeological model from different operators, *Proceedings of the 3<sup>rd</sup> ISPRS International Workshop 3D-ARCH 2009*, 2009, Trento, Italy.
- [4] Behr, J.; Eschler, Y.: X3DOM - A DOM-based HTML/X3D Integration Model, *Proceedings of the 14<sup>th</sup> International Conference on 3D Web Technology*, 2009, 127-135. <http://dx.doi.org/10.1145/1836049.1836076>
- [5] Cheng, J.; Han, C.; Law, Y.; Zhang, Y.; Kumar, B.: Web-enabled model-based CAD for design, *Proceedings of the International Conference on Computing in Civil and Building Engineering*, 2010.
- [6] Di Benedetto, M.; Ponchio, F.; Ganovelli, F.; Scopigno, R.: SpiderGL: A JavaScript 3D Graphics Library for Next-Generation WWW, *Proceedings of the 15th International Conference on Web 3D Technology*, 2010, 165-174. <http://dx.doi.org/10.1145/1836049.1836075>

- [7] Fuh, I.; Li, W.: Advances in collaborative CAD: the state of art, *Computer Aided Design*, 37(5) 2005, 571-581. <http://dx.doi.org/10.1016/j.cad.2004.08.005>
- [8] He, F.; Han, S.: A method and tool for human-human interaction and instant collaboration in CSCW-based CAD, *Computer in Industry*, 57(8-9), 2006, 740-751. <http://dx.doi.org/10.1016/j.compind.2006.04.019>
- [9] Hulubei, T.; Freuder, E.; Wallace, R.: The Goldilocks problem, *AI EDAM* 2003, 17(1), 2003, 3-11. <http://dx.doi.org/10.1017/S0890060403171028>
- [10] Khronos Group, Open standards for media authoring and acceleration, <http://www.khronos.org>, 2009.
- [11] Manferdini, A. M.; Remondino, F.: A Review of Reality-Based 3D Model Generation, Segmentation and Web-Based Visualization Methods, *International Journal of Heritage in the Digital Era*, 1(1), 2012, 103-121. <http://dx.doi.org/10.1260/2047-4970.1.1.103>
- [12] Manferdini, A. M.; Remondino, F.: Reality-based 3D modeling, segmentation and web-based visualization, *Digital Heritage*, 6436, 2010, 110-124. [http://dx.doi.org/10.1007/978-3-642-16873-4\\_9](http://dx.doi.org/10.1007/978-3-642-16873-4_9)
- [13] Mengoni, M.; Peruzzini, M.; Raffaelli, R.; Raponi, D.: A web-based platform to support contract furniture design, *Computer Aided Design & Applications*, 11(5), 2014, 533-543. <http://dx.doi.org/10.1080/16864360.2014.902684>
- [14] Moncur, R.; Jensen, C.; Teng, C.; Red, E.: Data consistency and conflict avoidance in a multi-user CAX environment, *Computer Aided Design & Application*, 10(5), 2006, 727-744. <http://dx.doi.org/10.3722/cadaps.2013.727-744>
- [15] Pescarin, S.; Calori, L.; Camporesi, C.; Ioio, M. D.; Forte, M.; Galeazzi, F.; Imboden S.; Moro, A.; Palombini, A.; Vassallo, V.; Vico, L.: Back to 2nd AD A VR on-line experience with Virtual Rome Project, *VAST'08 Proceedings of 9<sup>th</sup> International Symposium on Virtual Reality, Archeology and Cultural Heritage*, 2008, 109-116. <http://dx.doi.org/10.2312/VAST/VAST08/109-116>
- [16] Raffaelli, R.; Mengoni, M.; Germani, M.: Improving the link between computer-assisted design and configuration tools for the design of mechanical products, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 27(1), 2013, 51-64. <http://dx.doi.org/10.1017/S0890060412000388>
- [17] Sabin, D.; Weigel, R.: Product configuration frameworks - a survey, *IEEE Intelligent Systems*, 13(4), 1998, 42-49. <http://dx.doi.org/10.1109/5254.708432>
- [18] Visintini, D.; Siotto, E.; Menean, E.: 3D modeling of the St. Anthony Abbot Church in S. Daniele del Fiuli (I): From laser scanning and photogrammetry to Vml/X3D model, *Proceedings of the 3<sup>rd</sup> ISPRS International Workshop*, 2009.
- [19] Wallner, G.: Geometry of arbitrary light distributions, *Proceedings of the 3th International Conference on geometry and graphics*, 2008, Dresden, Germany.
- [20] Wang, J.; Tang, M.; Song, L.; Jiang, S.: Design and implementation of an agent-based collaborative product design system, *Computer in Industry*, 60(7), 2009, 520-535. <http://dx.doi.org/10.1016/j.compind.2009.03.001>
- [21] Zeng, J.; Chen, W.; Ding, Q.: A Web-based CAD system, *Journal of Materials Processing Technology*, 139(1-3), 2003, 229-232. [http://dx.doi.org/10.1016/S0924-0136\(03\)00225-5](http://dx.doi.org/10.1016/S0924-0136(03)00225-5)