Function Structure and Object-Oriented Virtual Prototyping Framework

Jen-Chi Wu, Ming C. Leu, and Xiaqing (Frank) Liu
Missouri University of Science and Technology, Rolla, MO 65409

Abstract
The paper proposes a virtual prototyping framework based on the function structure and object-oriented design methodology. This virtual prototyping framework integrates system architecture diagram, virtual components, and operational scheme to construct a virtual prototype. The system architecture model, which is from the function structure and object-oriented design approach describes the interactions and defines the interfaces between virtual components. A virtual component consists of 3D graphical representation, dynamic behavior codes deriving from the object model of the function structure and object-oriented design approach, and their interface. The operational scheme defines the operations the virtual product performs. By leveraging the design result from the function structure and object-oriented design methodology, our virtual prototyping framework is effective and efficient.

1. INTRODUCTION
This paper describes a virtual prototyping framework based on the function structure and object-oriented design methodology. The function structure and object-oriented design approach incorporates functional structure from engineering design and OO methods from modern software design for a more complete and robust product design method. This design method uses a function structure model for design and analysis of product functionalities and information flows. It then forms the product modules with the modular design techniques for product architecture. From these modules, an object model for representing the physical structure of a product is constructed. Finally, the object model and the information flows from the function structure are integrated into the system architecture model.

Virtual prototypes are basically 3D visualization models in digital forms with dynamic behavior simulated by software programs [1-3]. The modern software is mainly developed with the Object-Oriented Design and Programming technology[4, 5]. Since the function structure and object-oriented design methodology contains functional model, object model and system architecture model, the construction work of the virtual prototype becomes efficient by integrating and leveraging on the results from the design methodology.

Based on the system architecture diagram and object model from the function structure and object-oriented design methodology, the virtual components, operational scheme, and virtual product can be developed and integrated in the prototyping process. The system architecture diagram is the component's overview of the product/system in physical domain, and is constructed based on the information flow and object model from the design methodology. The virtual components are the product/system components with dynamic behavioral simulation in digital forms. The dynamic behaviors of a virtual component can be simulated according to the correspondent object from the object model in the product design methodology. The operational scheme is the statement to define the behavior and operations of the virtual product. A virtual product is the product/system in digital forms. The 3D visualized and interactive virtual prototype can be built by integrating the virtual components, the system architecture diagram and the operational scheme.

In this paper, the function structure and object-oriented design methodology is first introduced. The discussion of the virtual prototyping framework based on this design methodology then follows. Finally, an implementation example for the explosive ordnance robot system with our virtual prototyping framework is demonstrated.

2. PROPOSED VIRTUAL PROTOTYPING FRAMEWORK
The product design world consists of four domains, which are customer domain, functional domain, physical domain and process domain as shown in Fig. 1. [6]. In this world, the function structure and object-oriented design methodology can integrate a function structure in the functional domain, an object model in the physical domain, and a system architecture model, which analyzes the interactions and define the interfaces between object components.

Figure 1. Four Domains of Product Design World

2.1. Function Structure and Object-Oriented Design Methodology
The function structure and object-oriented design methodology consists of three operational models, which are function structure, object, and system architecture models. Its operating structure is shown in Fig. 2.

The function structure model begins with constructing a black box model to represent a product as a functional system. The black box model contains three types of inputs and outputs: material, energy, and signal flows. Material has properties of
form, mass, color, etc. Examples include solid, liquid, gas, workpiece, and other physical objects. Energy refers capability to do work such as electricity, hydraulic, magnetic, and mechanical forces, and so on. Signal is an event, message, or some form change, which can carry or convey the information. Examples include commands, status, image, sound, etc. After the black box model is developed, the function structure modeling process follows. This modeling process contains five phases of activities, i.e. Phase 1: Develop process descriptions as activity diagrams, Phase 2: Formulate subfunctions through task listing, Phase 3: Aggregate subfunctions into a refined function structure, Phase 4: Validate the functional decomposition, and Phase 5: Establish and identify product architecture and assemblies [7]. A function structure can be created with either a top-down or a bottom-up approach. The function structure model provides detailed relationships and interactions between functions by connecting and analyzing the functions with material, energy, and signal flows.

![Figure 2. The Operating Structure of Function Structure and Object-Oriented Design Methodology](image)

When the function structure is developed, the modular design techniques can be applied to form the modules of the product architecture. Two modular design techniques can be used for the module identification, the basic clustering method and module heuristics method. The basic clustering method uses a function structure to cluster subfunctions into modules based on the “natural” or intuitive groups that depend on each other and/or can be solved together. On the other hand, the module heuristics method identifies modules from the function structure using three strategies: dominant flows, branching flows and conversion-transmission modules [5]. These modules identified from the modular design are later converted into the objects in the object model.

The object model is based on High Order Object Modeling Technique, which is an object decomposition tool with structure analysis for the high order and primitive objects. Primitive objects can be readily understood by the product designers and developers, and they require no further analysis. High order objects, however, need to be decomposed. An object has attributes, and it operates methods to fulfill functional requirements in the functional domain [8]. Though the methods and functions can be different, they are treated identically in the design methodology for integration purpose. Moreover, a high order object encloses its component objects and their relation.

To create an object model, a context object diagram is first derived. This diagram is a top-level object model, which defines the scope of product and describes the interactions between the top-level product object and the external objects. This top-level product object, which is a high order object, is then decomposed into subobjects to fulfill or implement the subfunctions in the modules from modular design. If these subobjects are not primitive objects, both the corresponding modules and objects should be further divided into lower-level modules and objects. The procedure continues until all the high order objects have been decomposed into primitive objects.

The system architecture model is a technique for analyzing the interactions and defining the interfaces between objects. Analyzing the interactions refers to identify and group the external and the internal information flows into public or private object attributes. Defining the interfaces, on the other hand, refers to what attributes are required for what object methods and what is the best fit to integrate the interactive objects in term of physical structure.

When developing a system architecture model, two questions are asked. The first question is “Can the object in the system architecture model fulfill the functions in the corresponding module from the function structure model?" The second question is “Can the interactive objects integrate together in term of physical structure?" The purposes of the process are to ensure that all the functional requirements in functional domain have been fully supported by the objects in physical domain, and all the objects will fit well in term of physical structure.

A level-by-level development procedure of this framework is shown in Fig. 3. It begins with a black box model (BBM) in the functional domain. In the physical domain, a top-level object model (Context Object Diagram) then is constructed. After that, a top-level system architecture model is developed based on the top-level product object and the black box model.

After the system architecture model is created, the design team will review these top-level models. If the product object is a high order object in the context object diagram, the black box model needs to be decomposed into level-1 function structure to analyze its functionality. By applying the modular design to the level-1 function structure, the modules for product architecture form. The top-level product object then are decomposed into level-1 subobjects based on these modules.

Once the level-1 function structure and object models have been developed, the level-1 system architecture model can be constructed. The function structure, object, and system architecture models at this level will again be analyzed and reviewed by the design team. The process continues until all the models reach their bottom level, where all objects in the system architecture models become primitives.
2.2. Virtual Prototyping Framework

The virtual prototypes are basically 3D visualized and interactive models in digital forms constructed and controlled by software application. The modern software is mainly developed by Object-Oriented Design and Programming technology. Since the design results from the function structure and object-oriented design methodology contain functional structure, object and system architecture models, the construction work of the virtual prototype can leverage and integrate the results from the design methodology.

The proposed virtual prototyping framework begins with creating the object model and the system architecture model using function structure and object-oriented design methodology. The system architecture model then is used to describe the interactions and define the interfaces between virtual components. Object model, on the other hand, is converted into dynamic behavior codes for the correspondent virtual components.

A virtual component is the product/system component in digital form, which consists of 3D graphical representation, dynamic behavior code, and their interface. To construct a virtual component, its 3D graphical representation from CAD model will be created and saved with the STL formats first. These STL files of the virtual components then are imported and reconstructed into 3D graphical model again in the virtual reality software application. Furthermore, the object for this virtual component is converted into programming codes to simulate its dynamic behaviors. In addition, the interface between the 3D graphical model and the behavior code is analyzed and defined. Finally, the 3D graphical model, behavior code, and interface are integrated together as one entity to form the virtual component.

After the virtual components are created, they are assemble together based on the system architecture model to form the 3D graphical model of the virtual product. However, all the interactions and interfaces between the virtual components should be carefully analyzed and reviewed to ensure a conflict-free design.

Once the graphical model of the virtual product is constructed, the operational scheme needs to be defined for further development of the virtual prototype. The operational scheme is the statement to define the dynamic behavior and operations of the product/system.

The dynamic behavior and operation of the virtual prototype is represented and analyzed with the hierarchical state transition model, which is a hierarchy of high order and primitive state transition diagrams. The hierarchical state transition model consists of a network of object states, and a set of transitions between these states. Its notations are shown in Fig. 4.

The double-box entity on the left hand side of Fig. 4 represents a high order state, which needs to be decomposed into a set of substates in the analysis process. The box on the right is a representation of a primitive state, which is an abstraction of only attributes' values of an object, and no decomposition is needed. The transition includes a trigger and an event. An event takes an object from one state to another. A trigger is the precondition of an event.

With the hierarchical state transition model, the operational scheme of the virtual prototype is properly defined. The operational scheme should best show and tell the capabilities and benefits of the product; in other word, to catch the essence of the product.

Finally, the graphical model of the virtual product and the operational scheme are incorporated into the 3D visualized and interactive virtual prototype. A 3D interactive and visualized virtual prototype and its enclosed essential components are shown in Fig. 5. The detailed development procedure of this virtual prototyping framework is given in Table 1.

The first step in constructing the framework is to come up with a top-level (level-0) black box model in functional domain, object model in physical domain, and system architecture model. The design team analyzes, verifies, and reviews these top-level models to ensure the function flows and the product scopes are properly defined.

At step 2, level-1 function structure, object, and system architecture models are created. The design team analyzes, verifies, reviews level-1 function structure model, object model, and system architecture diagram, and checks if all the objects in the system architecture model have become primitive objects. At step 3, if all the objects become primitive objects, the function
structure and object-oriented design is completed, move to step 4; otherwise, continue the process for the next-level decomposition until all the objects have become primitive objects.

At step 4, create virtual components by integrating 3D graphical models, dynamic behavior codes, and interfaces based on the object model. At step 5, develop the operational scheme represented with the hierarchical state transition model to analyze the dynamic behavior of the virtual prototype. At step 6, construct the 3D interactive and visualized virtual prototype by incorporating the virtual components, the integrated system architecture diagram, and the operational scheme.

### Table 1: Development procedure for the Function Structure and Object-Oriented Virtual Prototyping Framework

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Function Structure</th>
<th>Object Model</th>
<th>System Architecture Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>1.1 Create level-0 black box model.</td>
<td>1.2 Create level-0 context object diagram.</td>
<td>1.3 Create level-0 system architecture model.</td>
</tr>
<tr>
<td></td>
<td>1.4 Analyze, verify, and review top-level function structure, object model, and system architecture diagram.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>FS</td>
<td>2.1 Create level-1 function structure model based on the level-0 black box model and apply modular design techniques to form modules.</td>
<td>2.2 Create level-1 object model based on the modules formed at step 2.1.</td>
</tr>
<tr>
<td></td>
<td>2.4 Analyze, verify, and review level-1 function structure model, object model, and system architecture diagram, and check if all the objects have become primitive objects.</td>
<td></td>
<td>2.3 Create level-1 system architecture model based on the object model and information flows of function structure.</td>
</tr>
<tr>
<td>Step 3</td>
<td>If all the objects have become primitive objects, the function structure and object-oriented design is completed, and moves to step 4; Otherwise, continue the process for the next-level decomposition until all the objects have become primitive objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>Create virtual components by integrating 3D graphical model, dynamic behavior code, and interface based on object model.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>Develop the operational scheme represented with the hierarchical state transition model.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>Construct the 3D interactive and visualized virtual prototype by incorporating the virtual components, the integrated system architecture diagram, and the operational scheme.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3. APPLICATION EXAMPLE

To illustrate the Function Structure and Object-Oriented Virtual Prototyping Framework, a virtual prototype of an Explosive Ordnance Disposal (EOD) Robot System is developed using the procedure in Table 1.

At step 1, a black box model ‘Disarm Bomb’ as shown in Figure 6 is first generated as the top-level (level-0) function structure model since the prime function of the robot system is to disarm the bomb. This function model then serves as the functional requirement in functional domain.

![Figure 6. Black Box Model at Level-0](image)

Once the black box model is created, a context object diagram for the robot system is built as the top-level object model shown in Fig. 7. The operator requests services from the robot system and the robot system provides services to the operator. The design goal of the robot system is to disarm bomb. The bomb can be deposited into the disposal container. The technical support team provides product services to the robot system such as maintenance and repair represented by a relation object named the supporting mechanism.

![Figure 7. Context Object Diagram for the Robot System](image)

After the top-level function structure and object models are constructed, a system architecture model is developed by integrating the information flows from black box model in the functional domain with the top-level product object in physical domain as shown in Fig. 8. In this model, the top-level function ‘Disarm Bomb’ is assigned as the main method of the top-level product object ‘EOD Robot System’. Also, seven attributes and
the required implementation parameters of the product object ‘EOD Robot System’ have been identified. The ‘Bomb’ and ‘Ground’ are material flows represented by bold arrows. The ‘Electricity’, ‘Light’ and ‘Moving Force’ are energy flows represented by normal arrows. The ‘Control Signal’ and ‘Feedback Signal’ are signal flows represented by dot arrows.

Figure 8. Level-0 System Architecture Model

The multidisciplinary team then analyzes, verifies, and reviews the top-level function structure, object, system architecture models to ensure:

(1) The level-0 function has been assigned as the method of the level-0 product object.
(2) All the attributes of the product object have been assigned.
(3) The implementation parameters for the method of the product object have been identified.

At step 2, level-1 function structure model is created based on the level-0 black box model. The modular design techniques then are applied to form modules. Three modules in part of level-1 function structure model are shown in Fig. 9.

Based on the modules, level-1 object model then are constructed. Two of the objects, ‘Battery System’ and ‘Video System’, are shown in Fig. 10. By integrating the information flows of function structure and the object model, level-1 system architecture model can be developed. Fig. 11 shows the system architecture model for the EOD robot system at level 1. At step 3, all the objects have become primitive objects from observing the objects in the level-1 system architecture model. The function structure and object-oriented design is completed, and the procedure moves to step 4.

At step 4, create virtual components by integrating 3D graphical model, dynamic behavior code, and interface based on object model. Fig. 12 shows the 3D graphical model and the behavior code of a track driving motor. At step 5, develop the operational scheme with the hierarchical state transition model. The top-level operational scheme of the EOD robot system contains ‘Launch’, ‘Running’, and ‘Closed’ states, which are shown in Fig. 13. This virtual prototype application begins with ‘Launch’ state then transits into ‘Running’ state through the ‘Execute Program’ event triggered by a ‘Start Click’. The ‘Running’ state becomes ‘Closed’ if the ‘Close Click’ is triggered and ‘End Program’ event happens.

```cpp
class Track_Driving_Motor {
    public:
        Track_Driving_Motor();
        virtual ~Track_Driving_Motor();
        bool Actuate_EE (float, bool);
        bool Convert_EE_to_ME (float, float);
        int Change_ME (float);
    private:
        float Electricity_Energy;
        bool Manipulator_CS;
        float Torque;
};
```

Figure 12. Track Driving Motor’s 3D Model and Class Definition
Because the S2: 'Running' state is a high order state, it is decomposed into 'Demo', 'Assembly Process', and 'Interactive Operation' substrates as the level-1 operational scheme. As shown in Fig. 14, it is a complete graph in this case since each state can transmit to all other states. The details such as the triggers and events are also shown in the graph.

All the states in level-1 operational scheme are high order states, meaning further decomposition is required. As an example, state 2.2: 'Assembly Process' is decomposed into 8 substrates, which are 'Clear Screen', 'Chassis', 'Battery System', 'Motors', 'Control Server', 'Track System', 'Arm', and 'Video System'. Their triggers, events, and state transition sequences are shown in Fig. 15.

At step 6, construct the 3D interactive and visualized virtual prototype by incorporating the virtual components, the integrated system architecture diagram, and the operational scheme. Fig. 16 shows the virtual prototype of the EOD robot system.

4. CONCLUSIONS

We have presented an effective and efficient virtual prototyping framework by leveraging on the function structure and object-oriented design methodology. With this framework, a multidisciplinary team of mechanical, electronic, and software engineers can co-design and co-analyze the product and its components from the perspectives of function, object, and system architecture. Also, the correspondent object is converted into behavior codes, and integrated with the 3D graphical model to form the virtual component. Furthermore, the operational scheme represented with the hierarchical state transition model is created to analyze the dynamic behaviors of the virtual prototype. Finally, the 3D interactive and visualized virtual prototype is constructed by integrating the virtual components, system architecture, and the operational scheme.

5. ACKNOWLEDGEMENTS

We acknowledge ISC for supporting this research.

6. REFERENCES