Product Design Data Analysis by Unified Modeling Language (UML) Diagrams

Sharareh Teimouri *

** Architectural and Civil Engineering Faculty, Islamic Azad University-Zanjan Branch*
Zanjan, Iran, sht_id2000@yahoo.com

Abstract: As a standard language in the software design, UML is used as a new analytical procedure in the industrial design. In software engineering, this formal specification language has been well accepted, due to having a group of symbols representing the concepts and final objectives. In industrial design, this unified modeling language has some advantages such as coordinating research groups together by creating some floating modules, and planning a framework to do the tasks. UML diagrams make a widespread analysis on system's functions, activity and statement. Under these circumstances, UML diagrams seem to be useful in both analysis and design stages of the product design and information analysis. Indeed, UML diagrams show how well different parts of a system work together, and how users interact with the final product. UML diagrams help industrial designers analyze the collected data being used in their product designs. Moreover, as the UML graphs show a product from different points of views, it can be used by different groups. This proposed method would be advantageous in system abstraction and modularization in terms of yielding a graphic view of gathered information, and analyzing input data altogether.

Key words: Data Analysis, UML, UML Diagrams, Visual Modeling Language.

1. Introduction

As the scale of design processes increases, as well as their knowledge intensity and organizational complexity, traditional approaches to process design may no longer suffice (Ernest van Aken, 2005). Thus, making a new procedure in analyzing information seems to be essential. Although, the structure of communication as scaffolds for knowledge construction has never been measured directly, the language-based communication has been argued to play a principal role (Dong, 2005). Unified modeling language or UML (Booch, 2001) is a formal and widely used method mainly in software engineering and management applications due to its potential in system analysis, abstraction and modularization. UML diagrams can be used as an effective tool in information analysis in industrial design. In this method, attempts are made to precisely identify the objects with all their specifications. Indeed, design strategies based on UML methodologies have been investigated in the literature. UML is used as an educational methodology in education of Industrial Design (Hu, 2007), or a container terminal (Bielli, 2005), propose a modeling tool to describe and simulate systems to manage and operate it.

This paper shows how modules can float between parallel diagrams and how main parts and subparts of a product can be classified. UML diagrams coordinate analysis of different parts and functions in a given system. By analyzing various parts of an ordinary machine and its states and functions, UML diagrams can be extracted, determining characteristics and relations between objects during the design procedure. The purpose of a class is to declare a collection of methods, operations, and attributes that thoroughly describe the structure and behavior of objects (Booch, 2001). Other diagrams such as "activity diagrams", "use case diagrams", "state charts" facilitate to figuring a whole approach to system totality and its function.

2. Data Analysis

Unclear process of information analysis in industrial design makes a need to improve the traditional ways of working in design (Cross, 2000). Indeed, due to massive and complex design-processes (Ernest van Aken, 2005) traditional approaches are not sufficient to solve all design problems. Every good complex project requires good design research (Manzini, 2009), but without a well accepted analysis method and a plan to classify them, all of gathered information seems as a pile of data without any use. Information analysis phase is considered as a central part to the integrated creative design process models (Howard, 2008). Actually, visual instruments are the best tools to organize and analyze gathered information (Manzini, 2009), and modeling methods are areas which their requirements can be thoroughly fulfilled by the visual instruments.

A standard analyzing modeling methodology classifies information and shows attributes, structures, behaviors and relationships of components. Abstraction, hierarchy, encapsulation and modularity (Papajorgji, 2003) are some significance in object oriented patterns. In Object oriented methodology each object is a model of the real world and can be physical, conceptual, or software (Papajorgji, 2003). Due to physical identity of industrial products, this paper takes into account only physical objects such as a mini dishwasher on which the proposed method has been applied in the experimental results section.

The following sections present how UML class and other diagrams as an association model can be created for a product, and how they analyze information. In addition, they present how composite diagrams make an industrial modeling language, and facilitate information transition between different projects.

3. Unified Modelling Language

Design knowledge is a collection of different cognitive artifacts including tools to help understand the state of objects and implement design ideas (Manzini, 2009). When data of a project are collected, there is a need to arrange them by a standard analyzing system. Unified Modeling Language as a standard modeling language has mentioned characteristics to describe and visualize object oriented systems based on UML diagrams. In this diagrammatic system, data, behavior and function are accounted as the attribute, operation and responsibility of an object. For instance, Simplified Flow Diagram (SFD) and Process Flow Diagram (PFD) with a set of specifications describe the model elements in the defined diagrams (Gabbar, 2003). Modeling as a language of designing (Sener, 2007) provides means to describe the behavior of a situation or system to a given level of accuracy (Gabbar, 2003). Like a special modeling language, UML is based on object oriented methodologies and analysis. Because of having a collection of different diagrams for different purposes, and due to its potential in system abstraction and modularization, UML is a highly abstract language (Hu, 2007) as well as formal and widely used method (Taylor, 1990) mainly in software engineering and management applications. It is used in analysis and design phase in industrial design.

A model is a set of collected data diagrams. UML as a set of graphical notation makes abstracted graphical models of every real product. This diagrammatic model shows all data and entities, and determines all relationships between different parts in a system by its easy to understand visual method. UML as a visual modeling language is a communication tool to contact other designers and involved groups and it provides an overall insight of the design procedure, which promotes mutual understanding and cooperation between different groups in a design project. It enables everyone to read, interpret and understand all its diagrams. In this paper, it is presented how a product as a sample is shown by UML diagrams and how UML have been employed as an assistant tool in analysis and modeling phase in industrial design and detecting its probable drawbacks.

3.1 Object Diagram

As a modeling language, UML can provide a diagrammatic and language base model of a system analysis by its diagrams. UML data flow diagrams analyze and model systems rapidly and entirely, like what is done with actual models in the real world. Analysis derives the actual behavior from the synthesized structure (Howard, 2008). Visualized format of UML diagrammatic layout makes all parts of a system perceived; that is, by UML diagrams, we have a diagrammatic model of a system. We describe our example by objects, classes and other diagrams as well, and provide a whole simulated model of a system or a product.

An object is an entity with a well-defined boundary and identity that encapsulates state and behavior, constituting an instance of a class (Booch, 2001). Every object has characteristics such as state, behavior and identity (Papajorgji, 2003). It can be shown in the UML object diagram (Fig.1). In the right side of the diagram, there is a general kind of dishwasher products, and in the left side, we can find the special case such as a mini dishwasher.

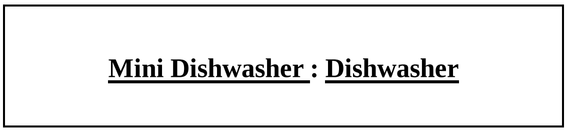

Figure.1 Sample of UML object diagrams

All objects, subsets and classes can be introduced and accepted as modules. Once a product is designed and developed, components of the product are observed to have other potential uses (Stone, 2000). Modules can be diagrammatic or physical structures. They are smaller parts of a system created independently and used in other systems. There is no need to any change to adapt them to other products. Thus, various combinations of them can create new objects and modules to transfer to various concepts. Likewise, there is no need to redefine the concepts which are already defined. Modules allow products to be easily updated in the future and present the opportunity to share them across products (Stone, 2000). Therefore, there is no need for repeating the concepts by the designer. This method can also prevent redesigning of the similar modules to other products and these modules can float among parallel projects. In object oriented method, an object may be a combination of some components. Therefore, in design process, changes are performed from details to whole.

3.2 Class Diagram

UML Class diagram is used for a wide variety of purposes, and as a central analysis diagram of object oriented design provides static structure diagrammatic model, and shows the fundamental constructions and interactions of a system such as inheritance, aggregation, and association. This diagram is applicable to all objects in a system such as places, actors, events and so on. A class is the descriptor for a set of objects with similar structure, behavior, and relationships (Booch, 2001). An object as a class is shown with a rectangular and three main partitions of it: Class name, Attributes and Operations (Fig.2).

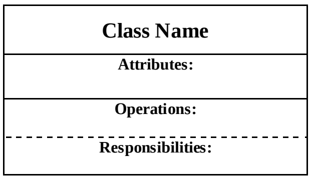

Figure.2 Standard UML class diagram

Class name is a standard name and must be written at the head part of the class diagram. Attribute is the second part of a class diagram that describes the range of values of a property (Booch, 2001). Attributes represent the structure of a class and they may be found by examining class definitions, the problem requirements, and by applying domain knowledge. An operation is a service that can be requested from an object of a class to affect the behavior (Booch, 2001). On the other hand, the behavior of a class is represented by its operation. They may be found by examining interaction diagrams. In the UML diagram group, a lot of classified information might be acquired from a class diagram. Responsibilities show a class tasks in a system. Attributes and operations can be assigned by public (+), private (-) and protected (#) signs to change their accessibility from other classes. Here is an application of a UML class diagram on a mini dishwasher machine (Fig.3).

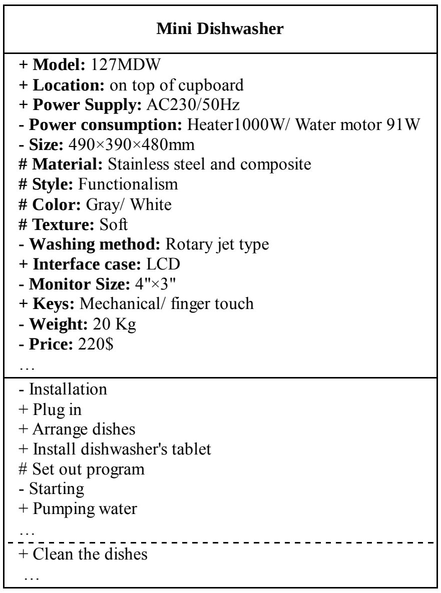

Figure.3 Sample of UML class diagrams on a mini dishwasher machine

The model is concerned with describing the intension of the class (Booch, 2001). Besides, UML diagrams show a subset model as an association class model. As modules, its classes and subclasses can be used in similar projects as well.

3.3 Use Case Diagram

A use case is a pattern of behavioral diagrams of a system and it provides a visual image of user's needs. It is a sequence of relationship of actions performed by an actor in a system dialog box. The main purpose of a use case diagram is to show what functions are performed for which actors, and what roles of an actor are in a system, and what the relationship is between actors and actions in a system. Use cases are represented by ellipse and the actors are represented by stick figures (Fig.4). It describes a set of use cases enclosed by a system boundary, communication (participation), associations between the actors (stick man) and the use cases, and generalizations among the use cases (Liverani, 2004).

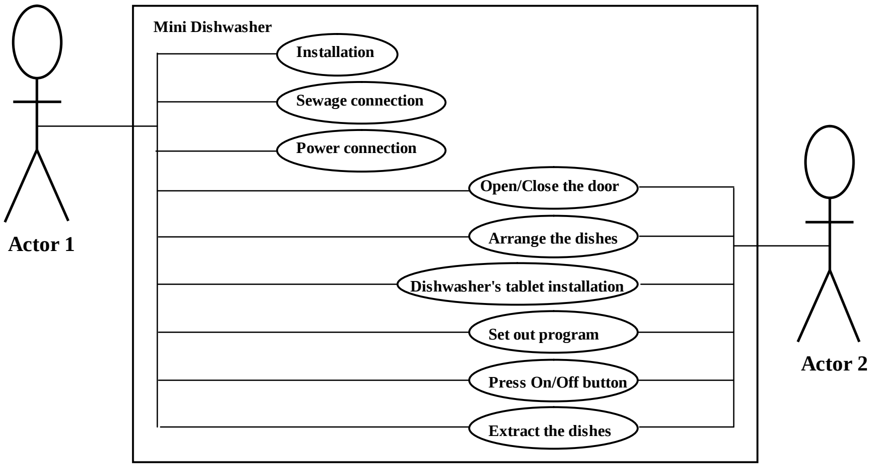

Figure.4 Sample of UML use case diagrams

3.4 State Diagram

A statechart diagram can be used to describe the behaviour of instances of a model element such as an object or an interaction (Booch, 2001). This diagram like a dynamic behaviour model shows the possible states of an object as a sequence of given events occurring in a system. It consists of states, transitions, the start and end points. States are shown by rounded rectangles to show the name, state variables and activities (Fig.5). State variables as the characteristic of a state refer to entry and exit time. Activities are the actions that executed on a state. Transitions are occurred between states without any changes in them and they are shown as a solid line with an arrowhead.

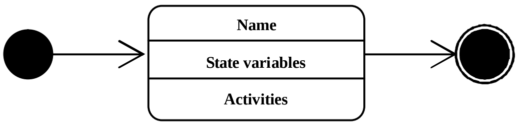

Figure.5 A standard UML state diagram

A state is a condition during the life of an object, or an interaction during which it satisfies some conditions, performs some actions or waits for some event (Booch, 2001). A transition as a relationship between two states and a future state of objects is shown by a solid line with an arrowhead symbol. In other words, this diagram is a graph of massages sent from one object to others. Every state diagram must have the start and end point (Fig.6).

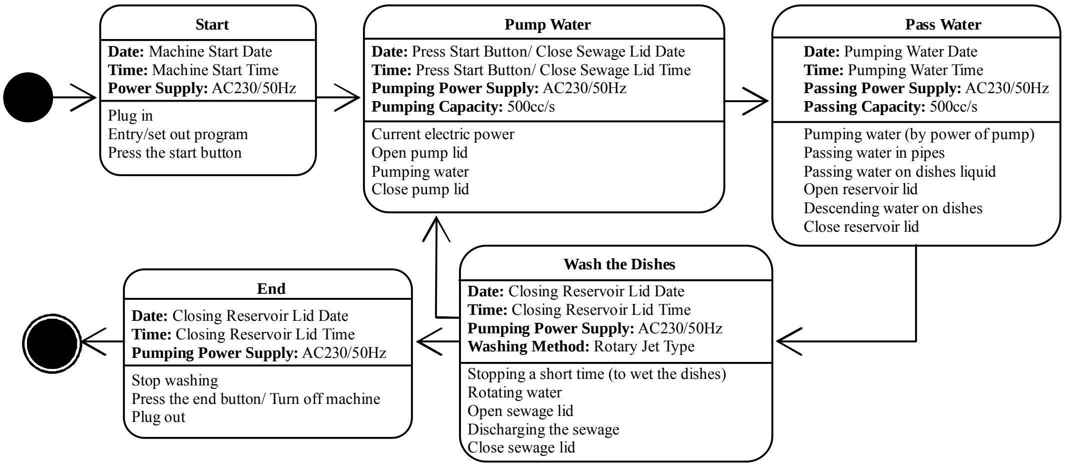
Start
Date: Machine Start Date
Time: Machine Start Time
Power Supply: AC230/50Hz
Plug in
Entry/set out program
Press the start button
Pump Water
Date: Press Start Button/ Close Sewage Lid Date
Time: Press Start Button/ Close Sewage Lid Time
Pumping Power Supply: AC230/50Hz
Pumping Capacity: 500cc/s
Current electric power
Open pump lid
Pumping water
Close pump lid
Pass Water
Date: Pumping Water Date
Time: Pumping Water Time
Passing Power Supply: AC230/50Hz
Passing Capacity: 500cc/s
Pumping water (by power of pump)
Passing water in pipes
Passing water on dishes liquid
Open reservoir lid
Descending water on dishes
Close reservoir lid
Wash the Dishes
Date: Closing Reservoir Lid Date
Time: Closing Reservoir Lid Time
Pumping Power Supply: AC230/50Hz
Washing Method: Rotary Jet Type
Stopping a short time (to wet the dishes)
Rotating water
Open sewage lid
Discharging the sewage
Close sewage lid
End
Date: Closing Reservoir Lid Date
Time: Closing Reservoir Lid Time
Pumping Power Supply: AC230/50Hz
Stop washing
Press the end button/ Turn off machine
Plug out

Figure.6 Sample of UML state diagrams

3.5 Sequence Diagram

UML sequence diagram is a model of usage-base scenario explaining the usage of a system. This diagram expresses time ordering (Papajorgji, 2003) and illustrates a real indirect relationship (Baudry, 2005) between events and objects. A sequence diagram focuses on the massage interchanging between a number of objects (Yang, 2007) to perform a task and chronological sequence of scenario base models to show the arrangement of processes operation. Sequence diagrams as the state diagrams are dynamic models of systems.

A sequence diagram has two vertical and horizontal dimensions (Booch, 2001). Parallel vertical lines show the current situation of processes for different objects, and horizontal arrows as the engine of sequence diagram represent the message exchanging between different instances (Fig.7).

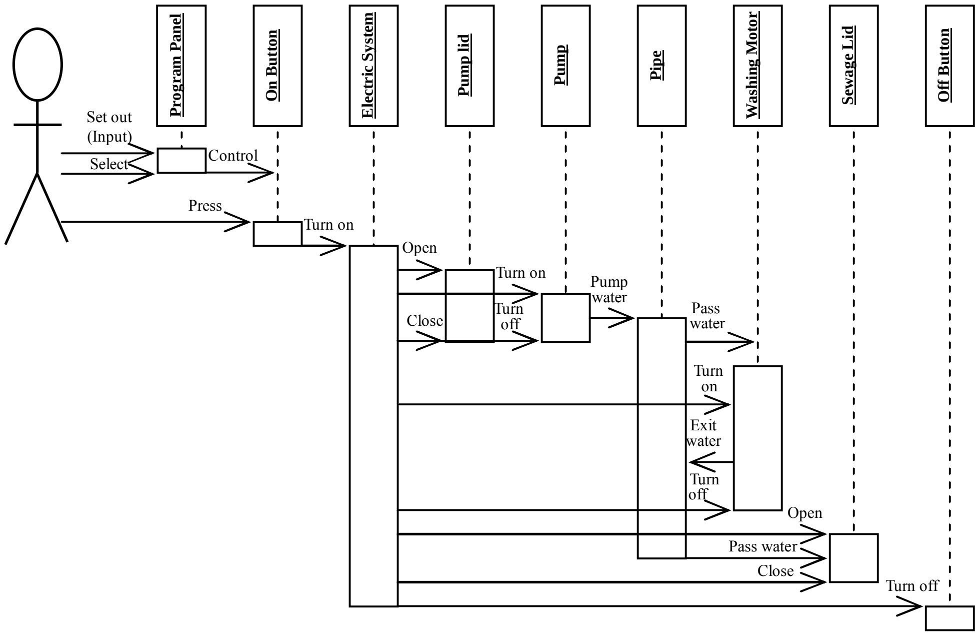

Figure.7 Sample of UML sequence diagrams

Since they have the ability to be understood easier, they are used to get a better grip of an interaction scenario for an individual designer or for a group that needs to achieve a common understanding of the situation (Refsdal, 2008).

3.6 Activity Diagram

An activity graph is a variation of a state machine in which the states represent the performance of actions or subactivities, and the transitions are triggered by the completion of the actions or subactivities (Booch, 2001). It is used for modelling of the behaviour of use cases, objects or operations. An activity diagram is a special case of a state diagram in which all (or at least most) of the states are action or subactivity states (Booch, 2001). As a flow of data, it presents the system's activity process and provides an overall control on every system. Like state diagram, it must have start and end points. Rounded rectangles and arrows are used to show activity and transition of activity (Fig.8).

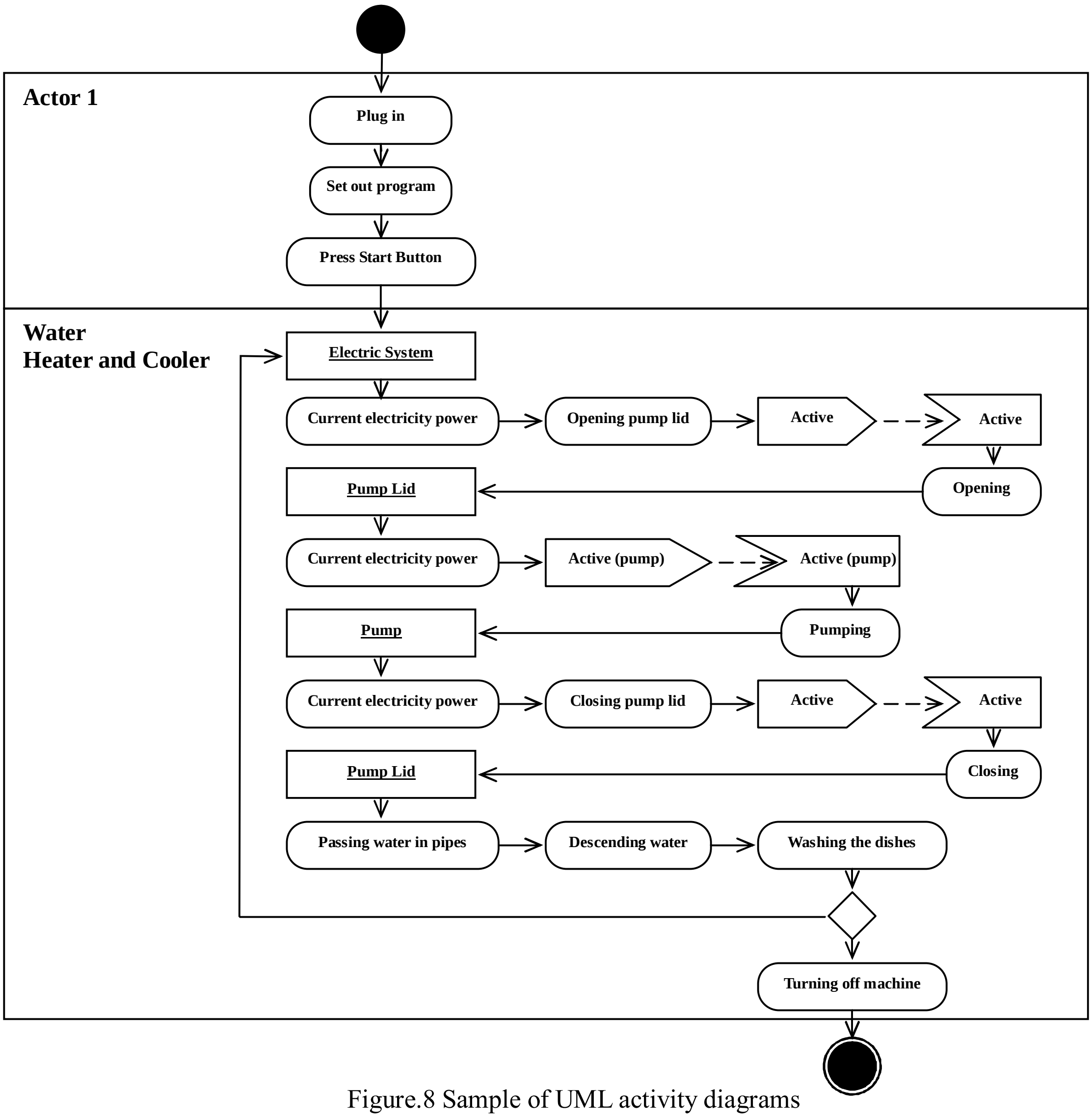

Figure.8 Sample of UML activity diagrams

3.7 Collaboration Diagram

Although, UML sequence diagram and collaboration diagram express similar information but they are different. Collaboration diagram shows the organization of the objects participating in an interaction. Collaboration diagram is used to define class behavior (Papajorgji, 2003). This diagram is used to define the outcome of an object in a scenario and represent the modeling and interactions between objects in a system. By using this kind of diagrams, the role of objects can be determined in a data flow diagram of every use case events and the classes' responsibilities as well. Objects and actors are described by links and messages to show how they are related and how they are interact with each other. The links as the sent massages show the relationship between objects and they are shown by a line between two objects (Fig. 9). The diagrams grow to be more and more complex as long as the number of objects and messages increases.

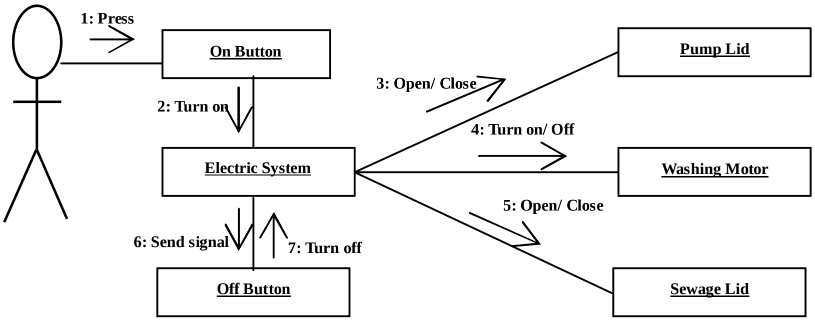

Figure.9 Sample of UML collaboration diagrams

3.8 Component Diagram

All involved groups in a project are able to get a physical view of system's structure using a component diagram. A UML component diagram is a graph of components connected by dependency relationships (Booch, 2001) between a system's classes and elements. This diagram as a modular diagram clarifies the organization and dependencies of a system and encapsulates information. Encapsulation separates the implementation of objects behavior from its public interface. It is called "information hiding" (Papajorgji, 2003).

A component diagram includes elements as the components, interfaces and connections. The component icon of this diagram is a rectangle with two little rectangle on the left side. The interface can be shown as a rectangle or a small circle connecting to a component by a solid line (Fig.10). To show additional expressions, notes can be added to each UML diagram by a folded corner rectangle. This diagram has only a type form, not an instance form. To show component instances, there is a need to use a deployment diagram (Booch, 2001).

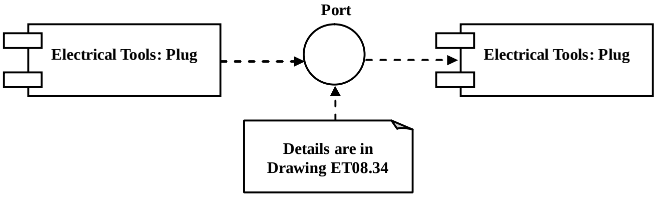

Figure.10 Sample of UML component diagrams

3.9 Deployment Diagram

UML deployment diagram shows a static and physical view of a system. This diagram is mostly discussed in the software design rather than the industrial design; because, it mostly discusses on the hardware structure of a system. Deployment diagrams show the configuration of run-time processing elements, software components, processes and objects executing on them (Booch, 2001). It is represented by the node loading information or includes component instances. On the other hand, the component diagram can be shown in a dependency relationship with a node. A solid line shows the communication between two nodes (Fig.11).

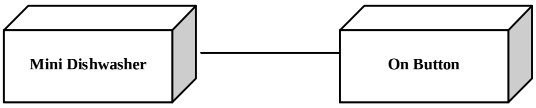

Figure.11 Sample of UML deployment diagrams

4. Common Language

A language should facilitate communication between end-users, decision makers and engineers by being intuitively understandable by all these groups (Refsdal, 2008). UML diagrams such as "activity diagrams", "use case diagrams", "class diagrams", "state charts" and "message sequence charts" are useful for both groups of product designers and software engineers to exchange their ideas and concepts to reach a common understanding (Hu, 2007) and it can be used in the similar manner in the industrial design. UML and its diagrams as a visual and graphical modeling language perform a communication role to contact designers and involved groups together. Accordingly, one of the main advantages of this language is the possibility of teamwork in a project. All related groups can work in a team, exchange information from the analyzing process, use the visual analysis in a team, and also understand the ultimate goal of the designing and analyzing information process. UML can coordinate research groups together by creating some floating modules and planning a framework to perform the tasks.

After the design knowledge is derived from information analysis and transferred to apply by other designers and it accumulates the design knowledge that could be the starting point for producing further knowledge by other researchers (Manzini, 2009). A graphical analysis can establish a common ground to make all parts of the design process transfer information more easily.

5. Conclusions

A new standard language in industrial design can be found to analyze information and to connect all involved groups in a project with each other. Used in software systems, UML graphs show, visualize and analyze products in the industrial design. Due to the graphical and modular ability of its diagrams, it can provide a graphical view of all required information of a project; which in turn, speeds up the design process such that the quality is preserved and misunderstandings emerged from teamwork would be lessened in the product research and development (R&D). The design acts as a connection between various involved groups in the project. The engineers working on the project consider it a success because they were able to find out significant errors in design and requirements before they started to code them. UML coordinates research groups by creating some diagrams and floating modules, and by planning a framework to perform the tasks. It also paves the way for all groups to understand the aim of the design. In addition, since the UML graphs can show the design procedure from different points of view, it can be used by different groups. The proposed method would be advantageous of having the potential in system abstraction and modularization by basic principles of object orientation: abstraction, encapsulation, modularity and hierarchy. In the future, on the basis of this new language, there can be found and made special communication tools in industrial design by adaptation and combination of other software languages.

6. References and Citations

[1] Baudry, Benoit and Le Traon, Yves, 2005, Measuring design testability of a UML class diagram, Information and Software Technology 47, p. 859-879.

[2] Bielli, Maurizio, Boulmakoul, Azedine and Rida, Mohamed, 2005, Object oriented model for container terminal distributed simulation, European Journal of Operational Research, Volume 175, Issue 3, 1731-1751.

[3] Booch, Grady, Jacobson and Ivar, Rumbaugh, 2001, OMG-Unified Modeling Language version1.4, USA: Object Management Group [Online PDF]. Available at <http://www.omg.org>.

[4] Cross, N., 2000, Engineering Design Methods-Strategies for Product Design, Jhon Wiely & Sons, England.

[5] Dong, A., 2005, The latent semantic approach to studying design team communication, Design Studies 26, p.445-461.

[6] Ernst van Aken, Joan, 2005, Valid knowledge for the professional design for large and complex design processes, Great Britain: Design Studies 26, p. 379-404.

[7] Gabbar, Hossam A., Suzuki, Kazuhiko and Shimada, Yukiyasu, 2003, Plant object- orientated model formalization- case study: HDS plant design, Design Studies 24, p. 101-108.

[8] Howard, T. J., J. Culley, S. and Dekoninck, E., 2008, Describing the creative design process by the integration of engineering design and cognitive psychology literature, Great Britain: Design Studies 29, 160-180.

[9] Hu, J., Ross, P., Feijs, L. and Qian, Y., 2007, UML in Action: Integrating Formal Methods in Industrial Design Education, Edutainment, p. 489-498.

[10] Liverani, A., Amati, G. and Pellicciari, M., 2004, "Symbolic Design": A UML Based Environment for Integrated Product Development, International design conference- DESIGN.

[11] Manzini, Ezio, 2009, New design knowledge, Great Britain: Design Studies 30, p. 4-12.

[12] Papajorgji, Petraq, 2003, Introduction to Object Oriented Design and Unified Modeling Language (UML), USA: IFAS conference.

[13] Refsdal, Atle and Stølen, Ketil, 2008, Extending UML sequence diagrams to model trust-dependent behavior with the aim to support risk analysis, Science of Computer Programming 74, p. 34-42.

[14] Sener, Bahar, 2007, Rethinking digital industrial design: a mandate for virtual workshops and intelligent environments, Digital Creativity 18, p. 193-206.

[15] Stone, Robert B., L. Wood, Kristin and H. Crawford, Richard, 2000, A heuristic method for identifying modules for product architectures, Design Studies 21, p. 5-31.

[16] Taylor, D., 1990, Object-Oriented Technology: A Manager's Guide, Addison Wesley.

[17] Yang, Dong, Wu, Hongwei and Tong, Lixin, 2007, A UML-based approach for the development of shop floor control systems, International Journal of Production Research, p. 1-33.