1. Introduction

Many formal specification techniques exist for modeling different aspects of software systems and it is difficult to find a single notation that can model all functionalities of a complex system clearly and precisely from the collections of syntax specifications and formal specifications notation are often extended and combined for modeling large and complex systems. In recent years, Formal Methods Integration has been a popular research topic. In the context of combining state-based and event-based formalisms, a number of proposals have been presented by Pressman (2005). The general observations on these works are that various formal notations can be used in an effective combination if the semantic links and establish these links. The semantic/syntax integration of those languages would be a consequence when the semantic links are precisely defined. Unlike UML, an industrial effort for standardizing diagrammatic notations, a single dominating integrated formal method may not exist in the near future.

The reason may be partially due to the fact that there are many different well established research work e.g., VDM forum. Another reason may be due to the open nature of the research community, which is different from the industrial 'globalization' community. Regardless of whether there will be or there should be an ultimate integrated formal method (like UML), diversity is to be the current reality for formal methods and their integrations. Such diversity may have an advantage on different formal methods and their combinations may be effective for developing various kinds of complex systems. The best way to support and popularize formal methods and their effective combinations is to build a widely accessible, extensible and integrated environment. The supply chain management provides an important infrastructure for a promising environment for various formal specification and design activities because it allows sharing of the difficult tasks and is to resist many good new proposals for extending UML a clear consequence and drawback of pushing a single language for modeling all software systems of various design models and provides hyper textual links among the models. The success of the Semantic specification and formal specification may have profound impact on the supply chain environment for formal specifications. Work is in co-operation easily. Many formal tasks like model reusing and model redefining can be achieved automatically or semi-automatically. This work only demonstrates an approach on how to build a semantic specification environment for supporting, checking, extending and integrating various formal specification languages. Furthermore, based on this semantic environment, specification comprehension (Queries for review/understanding purpose) can be supported. As the preliminary work this work also demonstrates how the traditional techniques can assist formal specification and design process.
Data flow diagrams models the flow of data into and out of an information system. It shows the processes that change or transform data. It shows the movement of data between processes and represents a system as a network of processes which transform data flowing between them.

Entities may be people, departments, other companies, other systems. These are called sources if they are external to the system and provide data to the system, and sink if they are external to the system and receive information from the system.

Components of a DFD:
1. Process:
   - The name of a process should describe what the process does and for what purpose.
   - It must have at least one input and at least one output
   - Labeled with verb + object (e.g. “print invoice” or “add customer”) (e.g. in the hierarchy below, none of the processes are primitive) or labeled more generally (e.g. “customer maintenance” or “warehouse reports”)

2. Data Flow:
   A data flow represents data in motion, moving from one place to another in the system. It may consist of many individual, related pieces of data that move together to a common destination name each data flow using a noun or noun phrase
   eg. Customer order
   - The name of a data flow should describe the details about the customer.
   - The name should include as much information as possible about the data flow.
   - It must originate from and/or lead to a process (this means that entities and data stores cannot communicate with anything except processes –as it takes a process to make the data flow)
   - It can go from process to process, but that does imply that no data is stored at that point
   - It can have one arrowhead indicating the direction in which the data is flowing
   - It can have 2 arrowheads when a process is altering (updating) existing records in a data stores.

3. Data Store: A data store represents a collection of data flows. Each data store has a unique name describing the contents of the data store. It may represent many different types of physical locations of data. It may be a temporary or a permanent repository of data different notations

4. External Agent: It is an entity with which the system communicates and which is outside the scope of the system eg. An outside organization or individual, another department or another system, who interacts with the system under consideration.
• It is a source if it is an origin of data coming into the system.
• It is a sink if it is a destination of data leaving the system.
• It is outside the system and defines its boundaries.
• It may be both a source and a sink what a sink does with data it receives from the system and how a source produces data which it inputs to the system are outside the boundary of the system and are not shown on the data flow.

**Definition 1:** A Data Flow Diagram consists of:
- A process is an activity or a function that is performed for some specific business reason;
- A data flow is a single piece of data or a logical collection of several pieces of information;
- A data store is a collection of data that is stored in some way;
- An external entity is a person, organization or system that is external to the system but interact with it.