CASE TOOLS

System analysis and System Design

Akhil Jain and Shubham Jain

Table of Contents

1. Feasibility Study

1.1. Understanding the Problem

In this section, we tried to understand the purpose of software.

The Computer Aided Software Engineering (CASE) tool allows Software Engineers to create simple unambiguous easily maintainable and reproducible software designs. CASE tools enable software engineers to abstract away from the entanglement of source code, to a level where architecture and design become more apparent and easier to understand and modify. This tool allows automated building of software design and automatically creates the data dictionary.
It helps in automation of various activities of system development and management processes, hence increases productivity of the development team.

1.2. Functionality

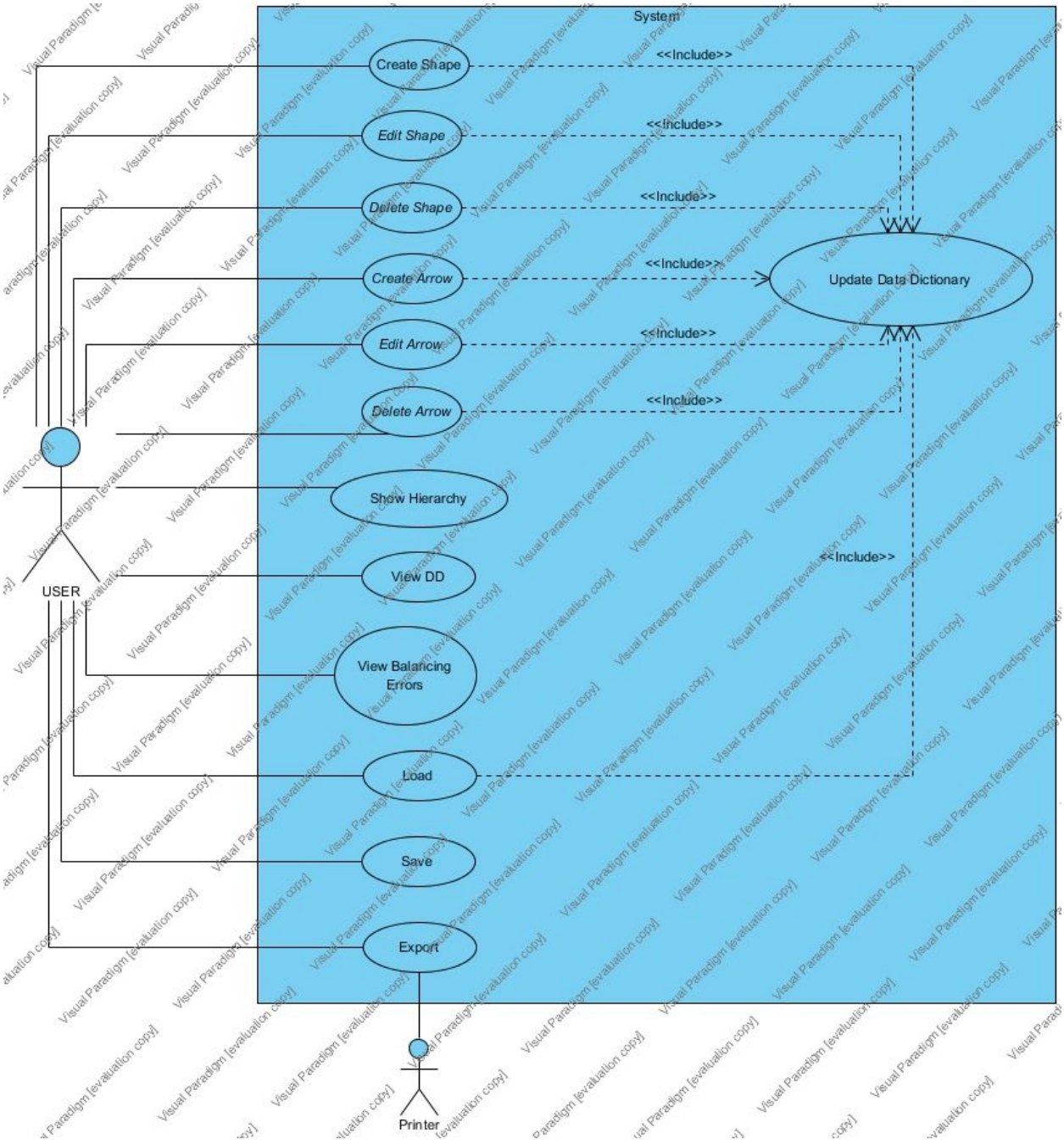
Structured Analysis:

The CASE tool supports a graphical interface and the following features

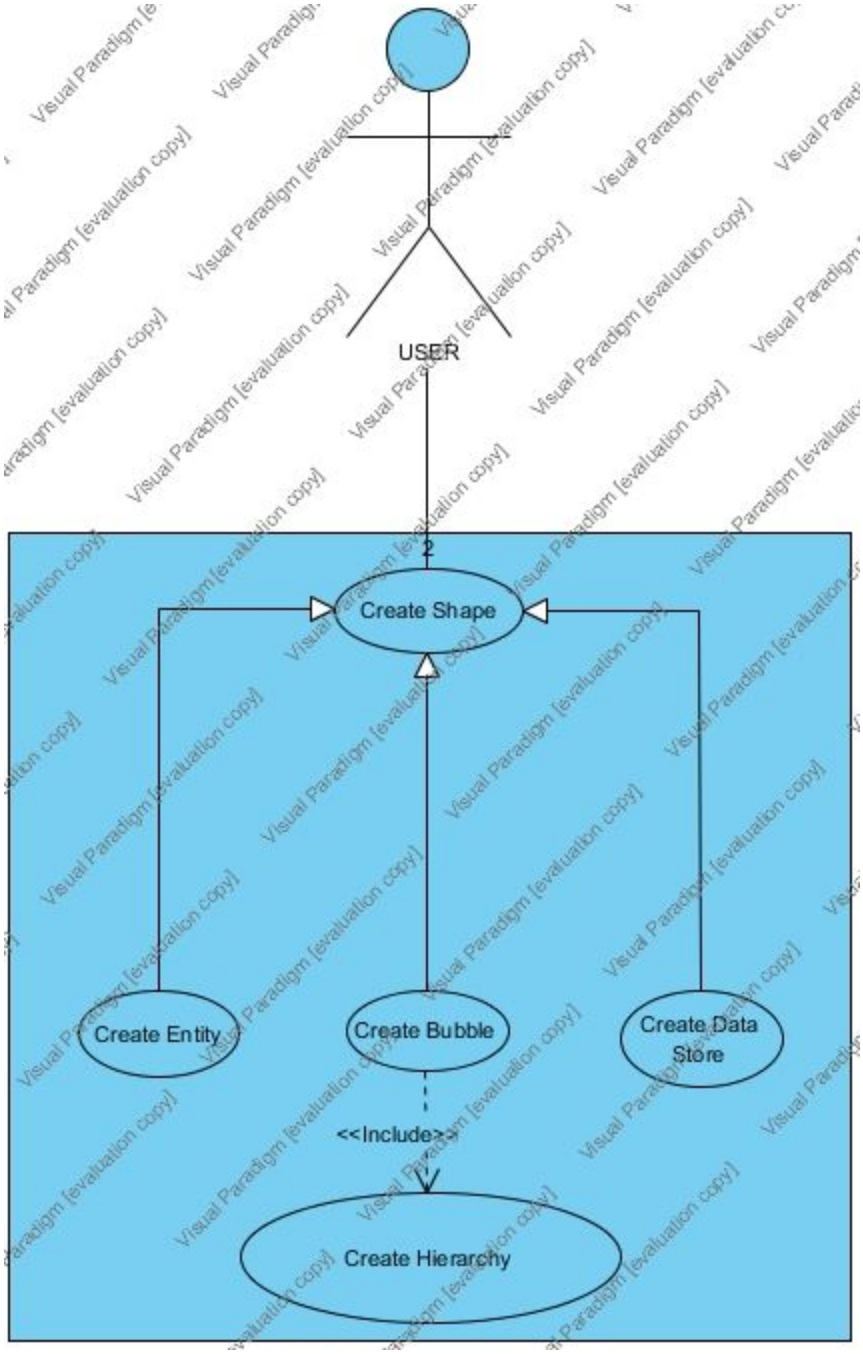
- The user can draw bubbles, data stores, and entities and can connect them using data flow arrows.
- Supports editing the data flow diagram.
- The user can create the diagram hierarchically.
- The user can determine balancing errors whenever required.
- The software creates the data dictionary automatically.
- Supports printing the diagram on a variety of printers.

Structured Design:

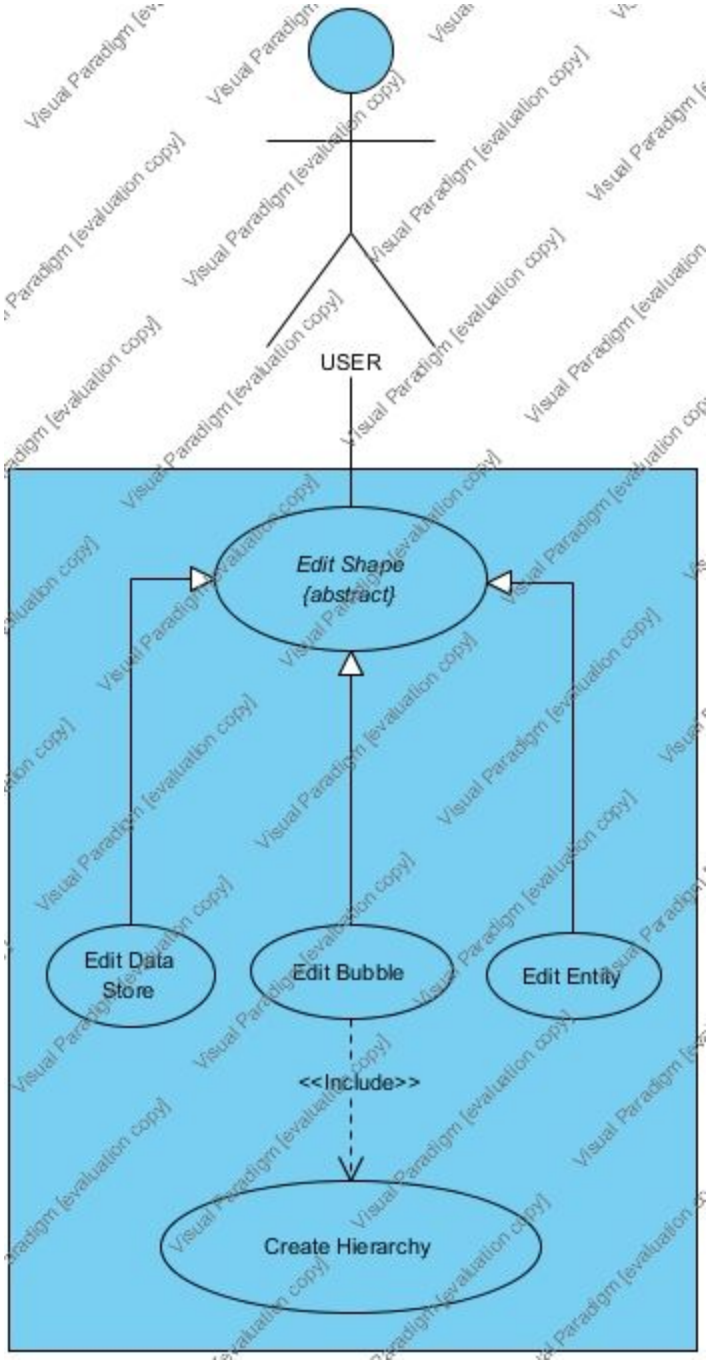
- The user can draw modules, control arrows, and data flow arrows. Also, a symbol for library modules is provided.
- The user can associate a module with some bubbles of the DFD. It can check if all the bubbles are assigned to some module and also whether each module is assigned at least one bubble.
- The user can modify his design.
- Clicking on a module shows its internal organization.
- The user can save his design and can load previously created designs.

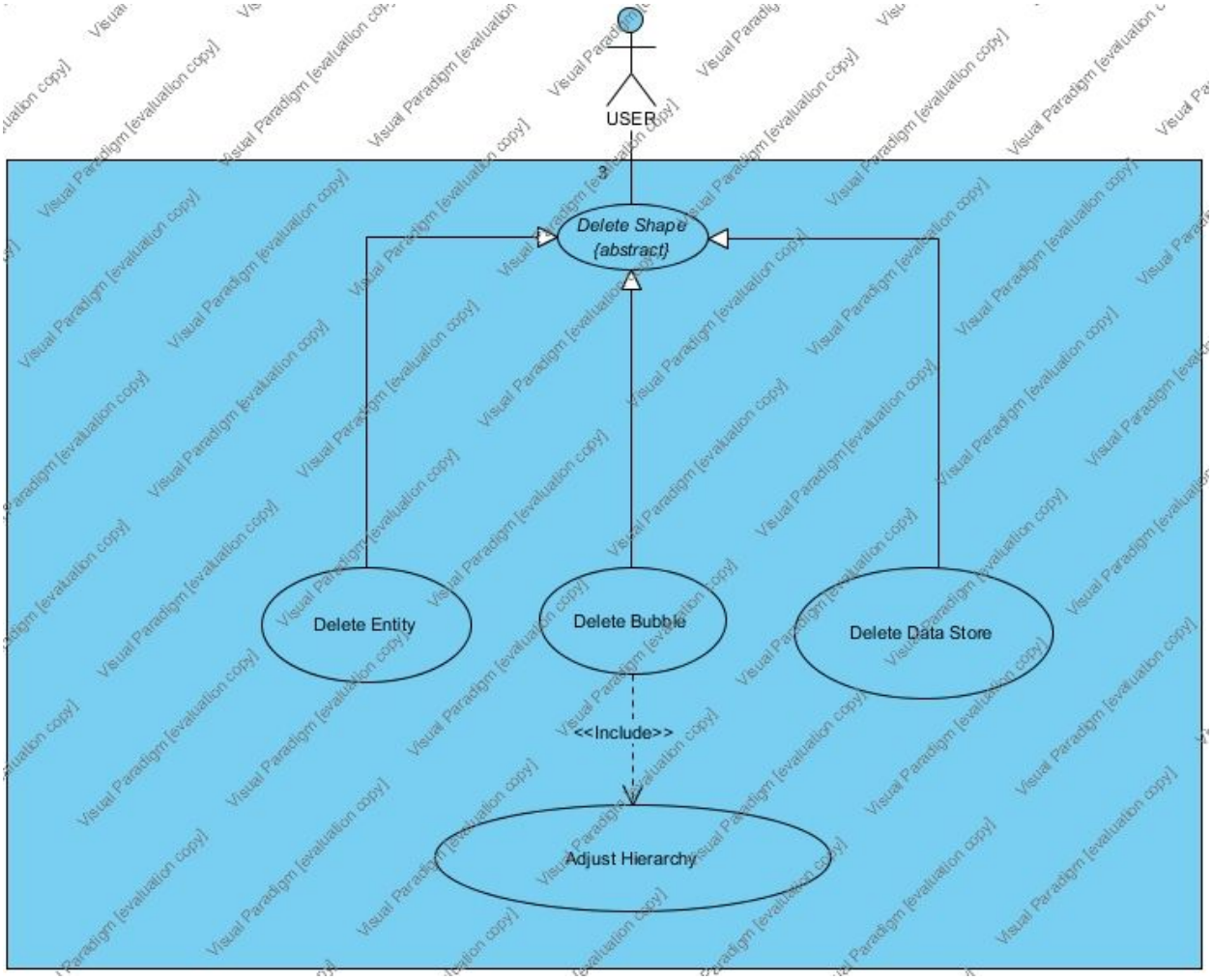


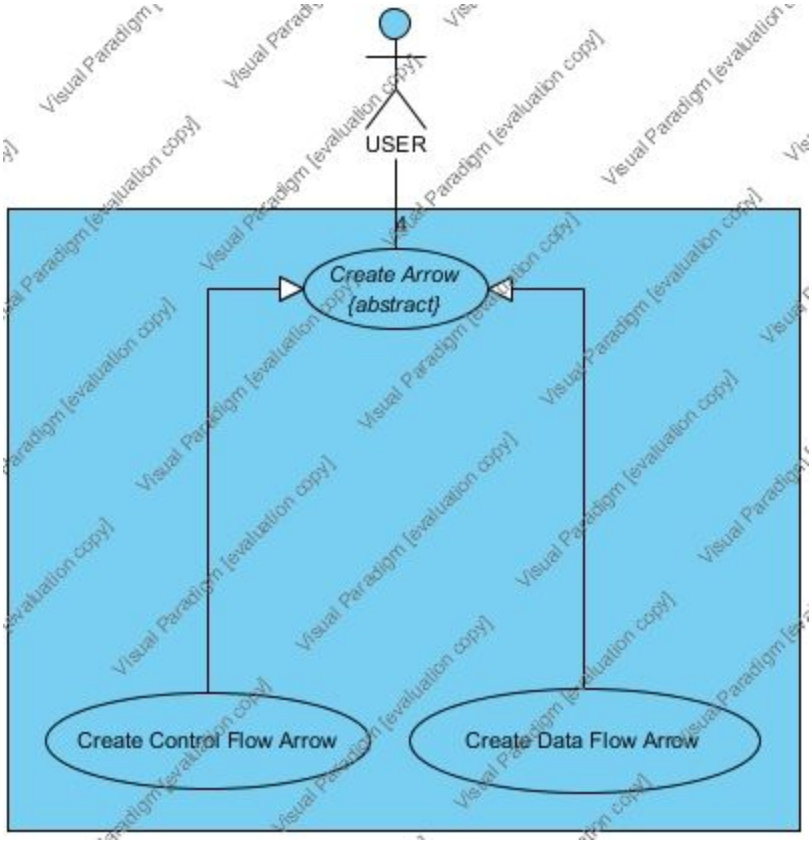
Use Case Diagram

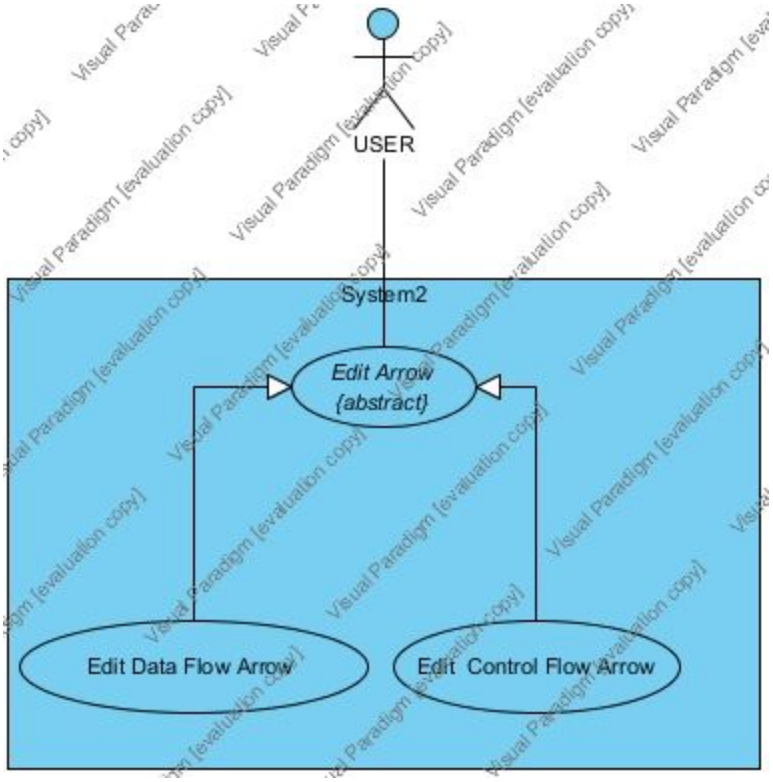

USER
2
Create Shape
Create Entity
Create Bubble
Create Data Store
<<Include>>
Create Hierarchy

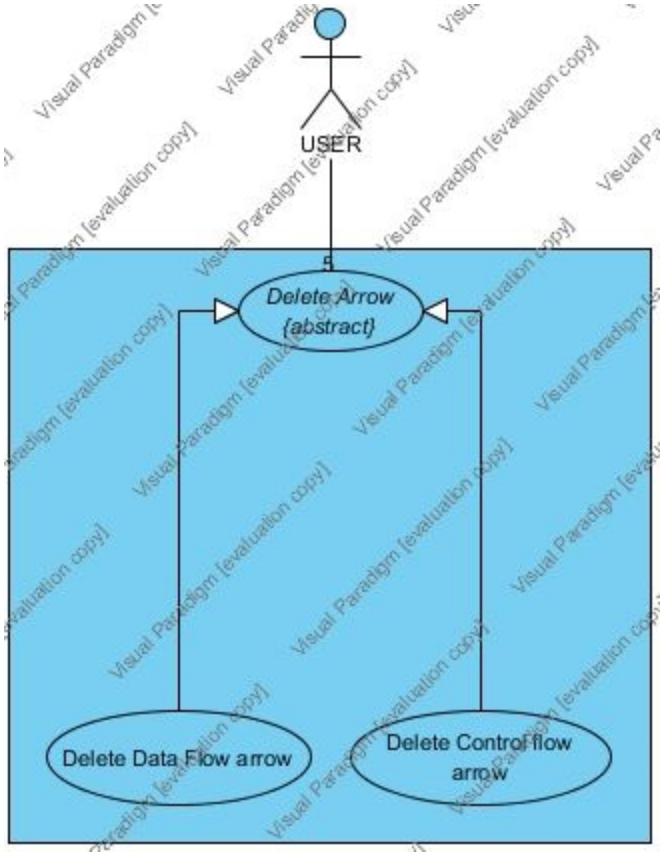
\

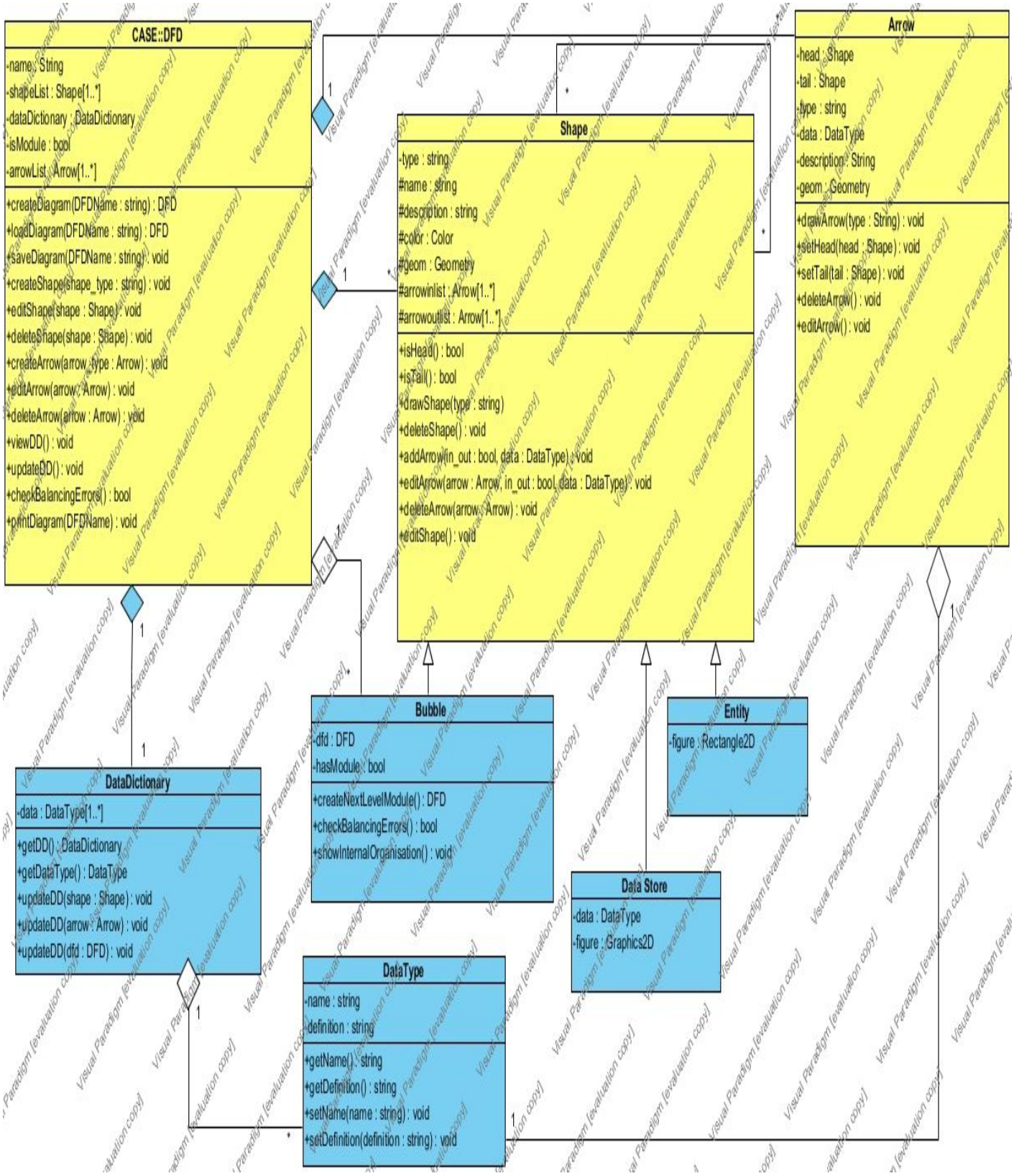

USER
Edit Shape
{abstract}
Edit Data
Store
Edit Bubble
Edit Entity
<<Include>>
Create Hierarchy


USER
Delete Shape
{abstract}
Delete Entity
Delete Bubble
Delete Data Store
<<Include>>
Adjust Hierarchy

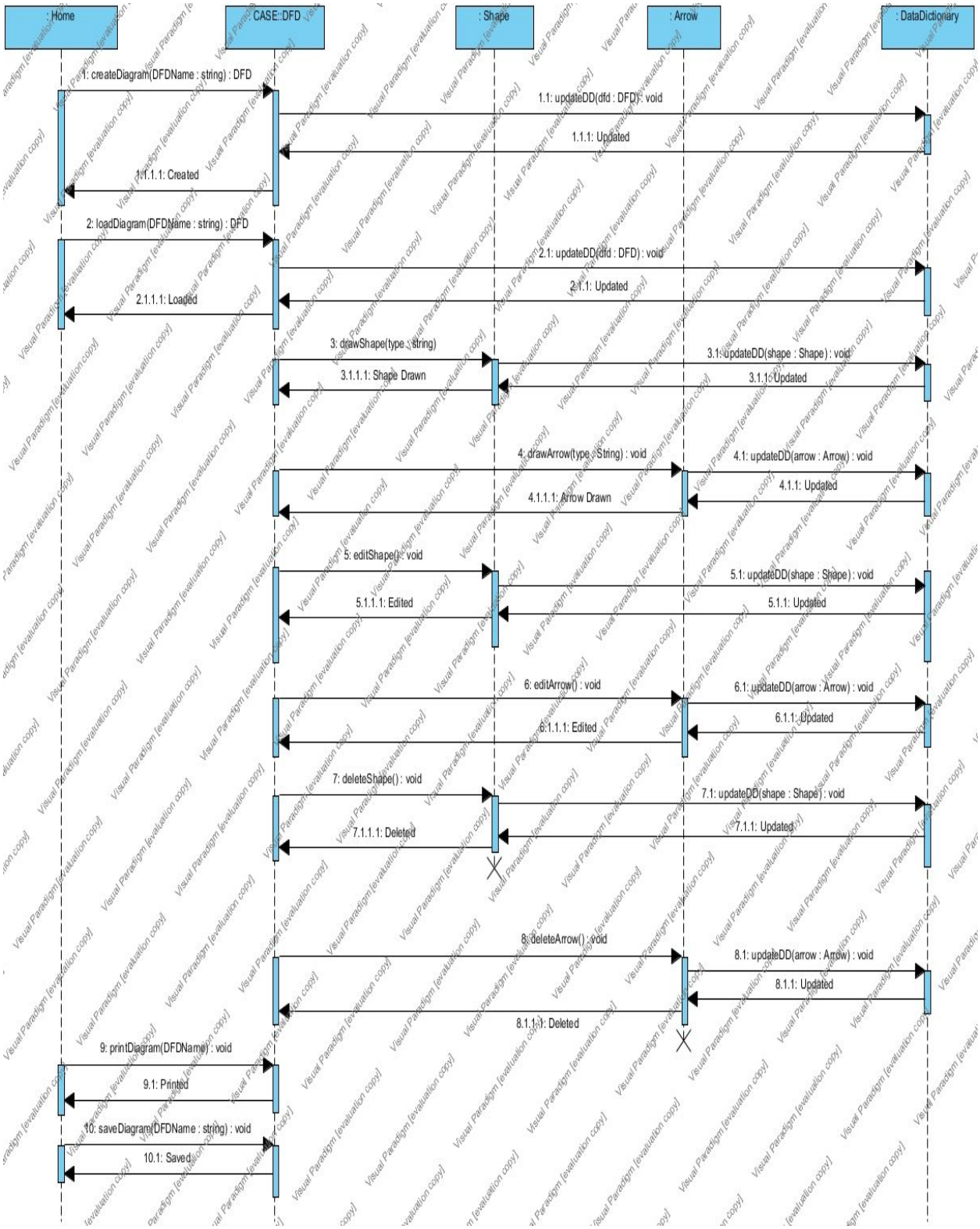

USER
Create Arrow
{abstract}
Create Control Flow Arrow
Create Data Flow Arrow


USER
System2
Edit Arrow
{abstract}
Edit Data Flow Arrow
Edit Control Flow Arrow

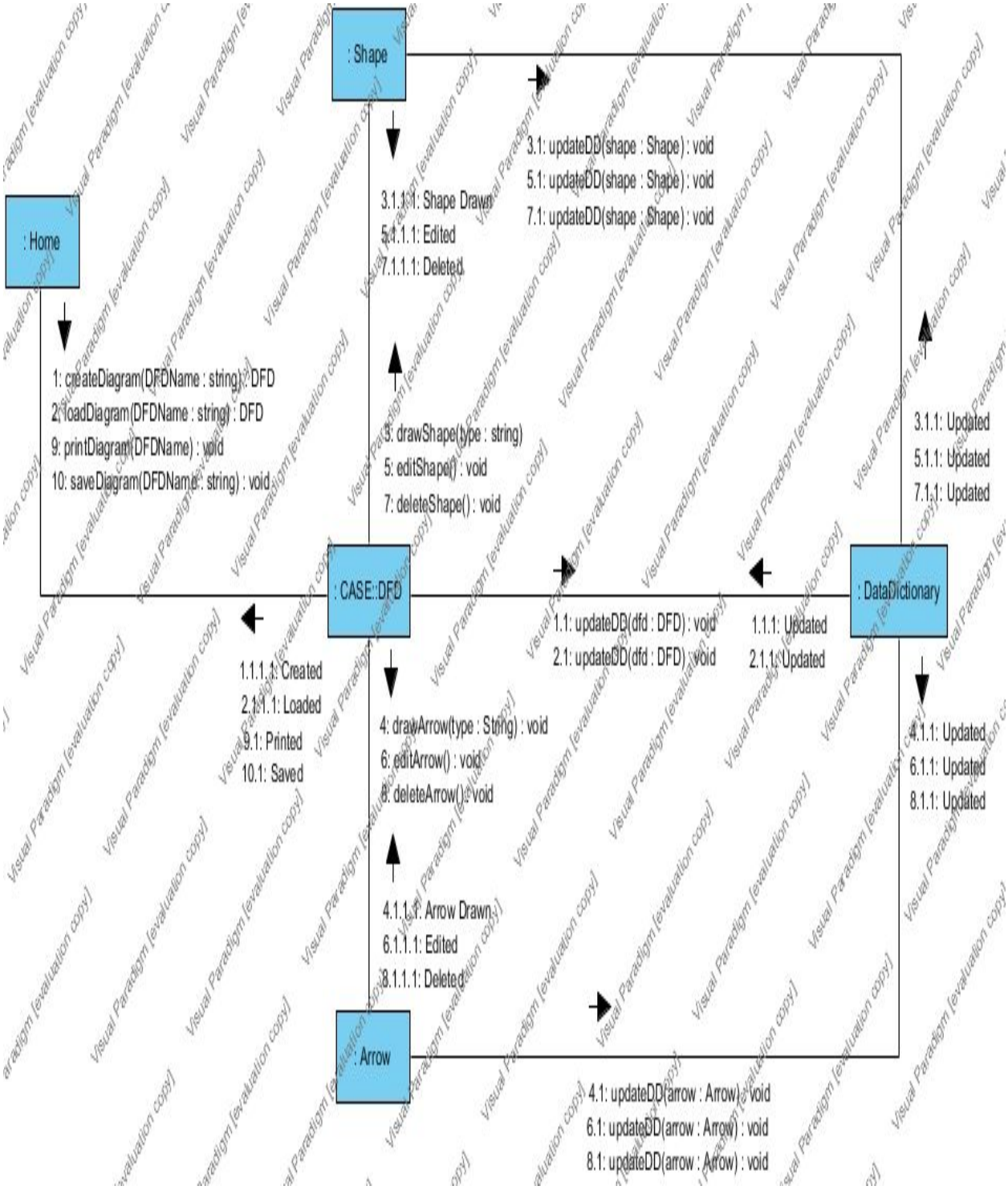

USER
Delete Arrow
{abstract}
Delete Data Flow arrow
Delete Control flow
arrow



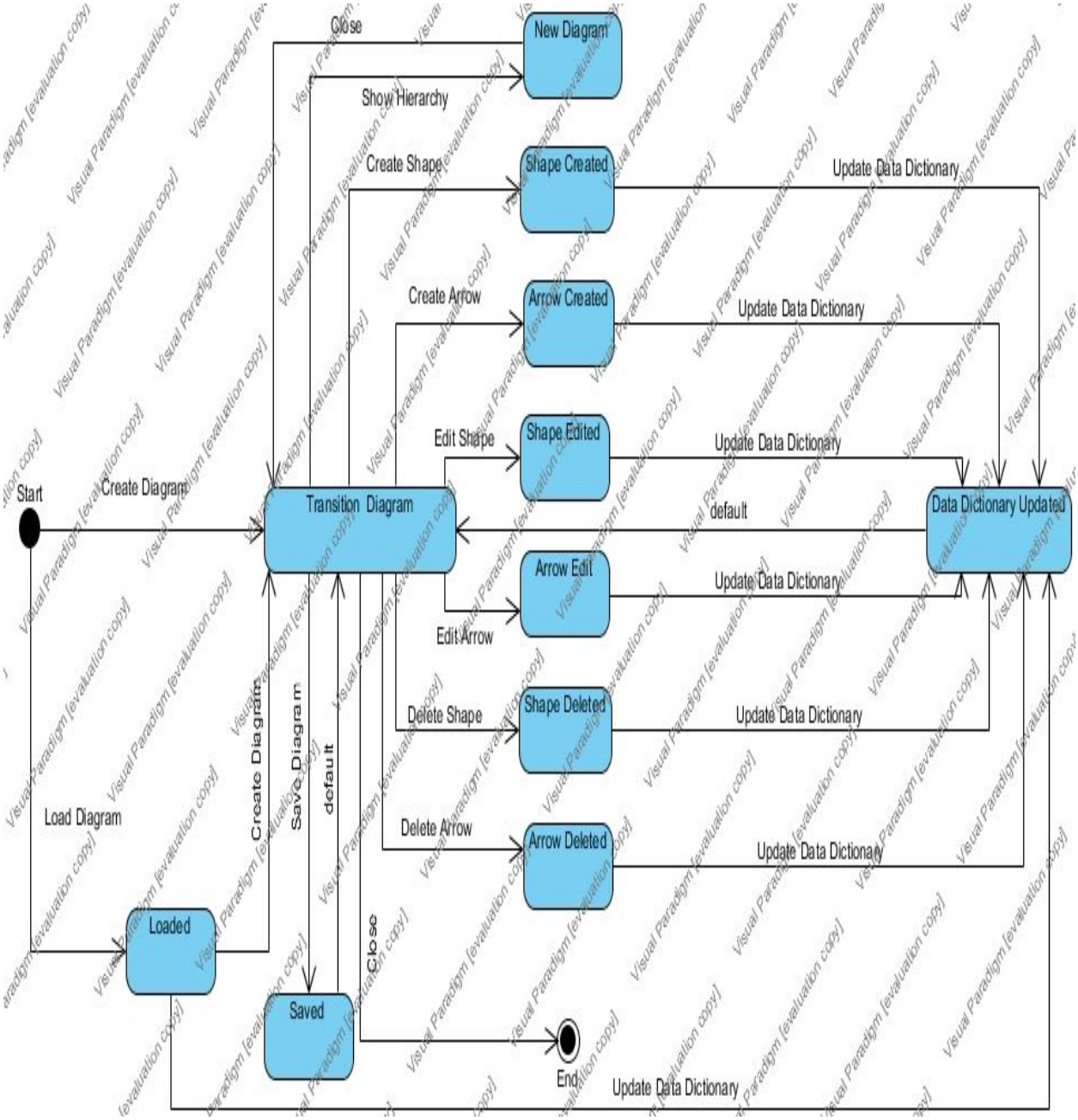
Class Diagram



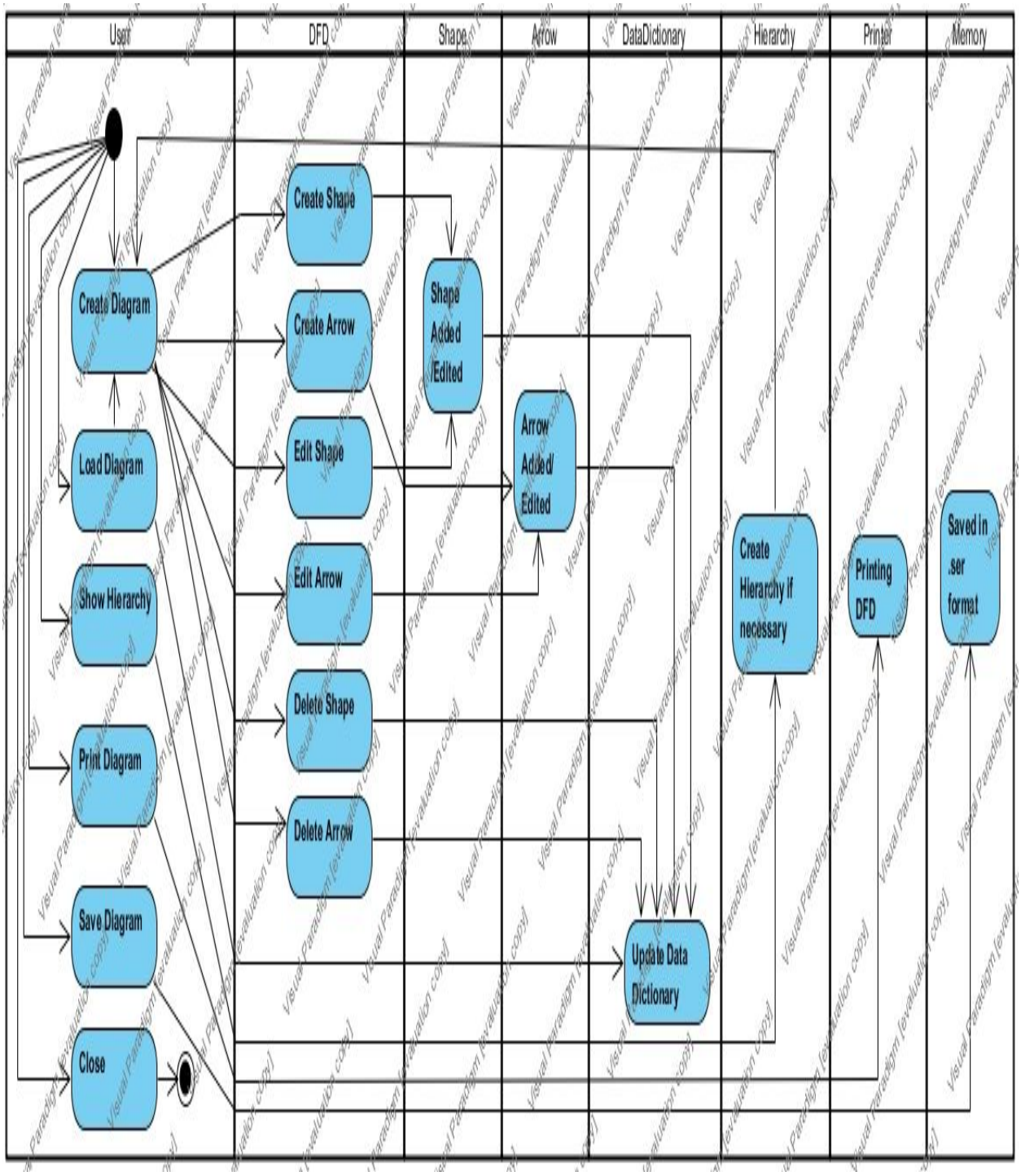
Sequence Diagram



Collaboration Diagram



State Chart Diagram



Activity Diagram

Data Dictionary

```
 Data: = Shapes
Diagrams = [{Diagram}*, Hierarchy Diagram]
Shapes = [{Shape}*, Delete Shape Command]  /*Shape is creating a
new shape or editing a shape, and also deleting a shape*/
File Manipulation Command = [Open Diagram, Create Diagram, Save
Diagram]
Open Diagram = Command
Create Diagram = Command
Save Diagram = Command
Check Balance Error Command = Command
Get Data Dictionary Command = Command
Command(Query) = ([Check Balance Error Command, Get Data
Dictionary Command,  View Hierarchy Command])
Output = ([Balance Error, Data Dictionary])
Data Dictionary = Data Dictionary /* The Data Dictionary of the
diagram drawn*/
```

Security

File is saved in .ser format.
No specific protection in terms of encryption of data is provided.

Detailed Analysis

3.1. Global System Architecture

The overall system architecture is a 2 tier architecture which includes client at one end and the database at the other. There is no server based middle tier in the software being designed.

Operating System:	Windows 7 and above, Linux
Code Standard:	The software will be developed in Java.
Libraries:	JRE (Java Runtime Environment)

Platform Minimum system requirements:

Recommended system requirements:
Hardware Requirements:

1. Operating system: Windows XP/98 or later versions, Linux
2. Processor: 400 MHz Pentium III processor or above
3. Hard Disk space: 500MB
4. RAM: 1 GB
5. Software Requirements: JDK and JRE packages must be installed.

Report

CASE tools: Advantages

- **Help standardization of notations and diagrams**
- **Help communication between development team members**
- **Automatically check the quality of the A&D models**
- **Reduction of time and effort**
- **Enhance reuse of models or models' components**

CASE tools: Disadvantages

- **Limitations in flexibility of documentation**
- **May lead to restriction to the tool's capabilities**
- **Major danger: completeness and syntactic correctness does NOT mean compliance with requirements**
- **Costs associated with the use of the tool: purchase + training**