UML Notation

Jim Fawcett

CSE681 – Software Modeling and Analysis

Fall 2017

Diagrams for Architecture

- **Activity Diagram** (high level program behavior)
  Shows activities a program carries out
  - Which activities may be executed in parallel
  - Which activities must be synchronized for correct operation
- **Package Diagram** (package structure of program or system)
  - Enumerates all software packages
  - Shows calling dependencies between packages
- **Module Diagram** (packages in a subsystem – focused on one responsibility)
  - Enumerates modules
  - Shows calling dependencies between modules

Design Documentation

- Class Diagram
  - Shows classes that are used in a program along with their relationships
- Sequence Diagram
  - Illustrates the timing of important messages (method invocations) between objects in the program.
- Structure Chart **(not UML)**
  - Shows calling relationships between all the functions in a package or module.
- State Diagram
  - Illustrates how a program navigates through its states.
- **Data Structure Diagram,** Ad-Hoc Diagram **(not UML)**
  - Presents the layout and relationships between important pieces of data in the program.

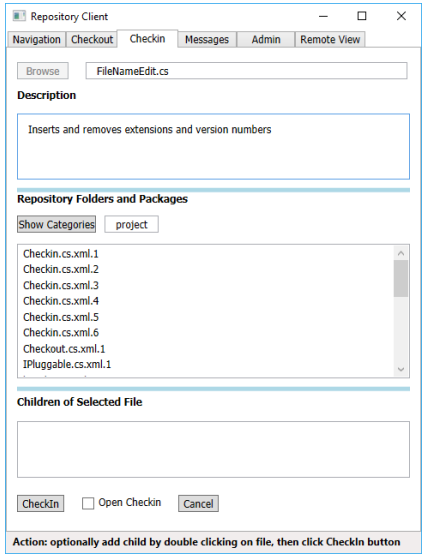
Diagrams for Requirements

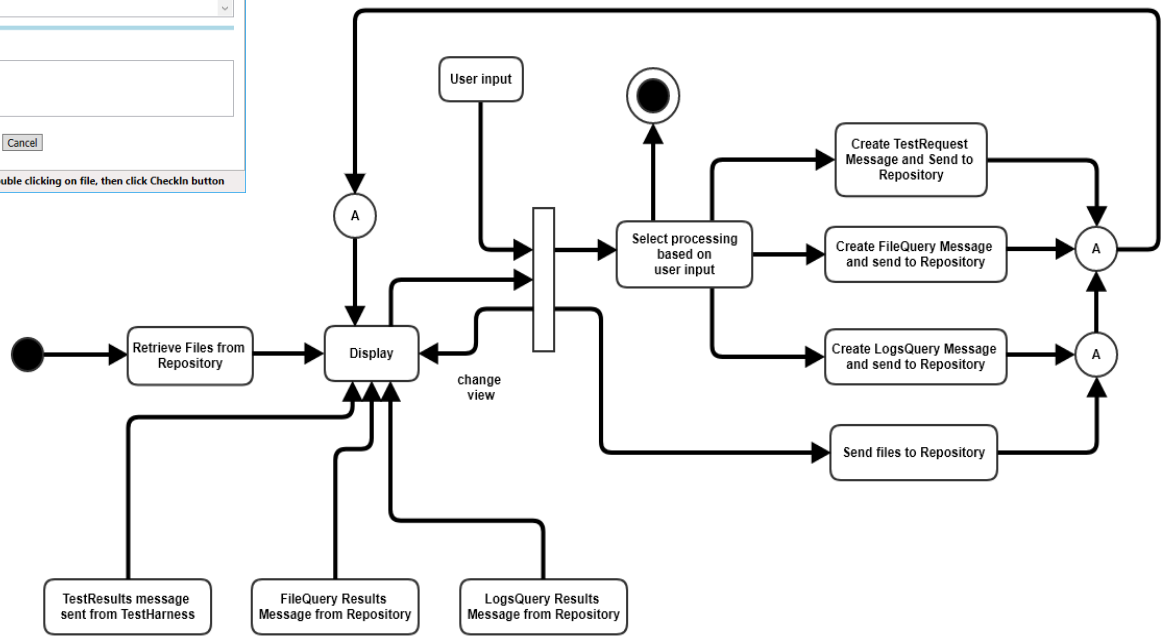
- **Context Diagram**
  - Shows how program interacts with its environment
- **Data Flow Diagram**
  - Represents requirement processing and the information flows necessary to sustain them.

Activity Diagram

Table of Contents

- Each of the blocks represents a specific processing activity.
- Start and stop activities are explicitly shown
- Synchronizing bars indicate timing constraints:
  - No output activity can start until all of the input activities have completed.
  - Multiple output activities indicate tasks that can run in parallel.



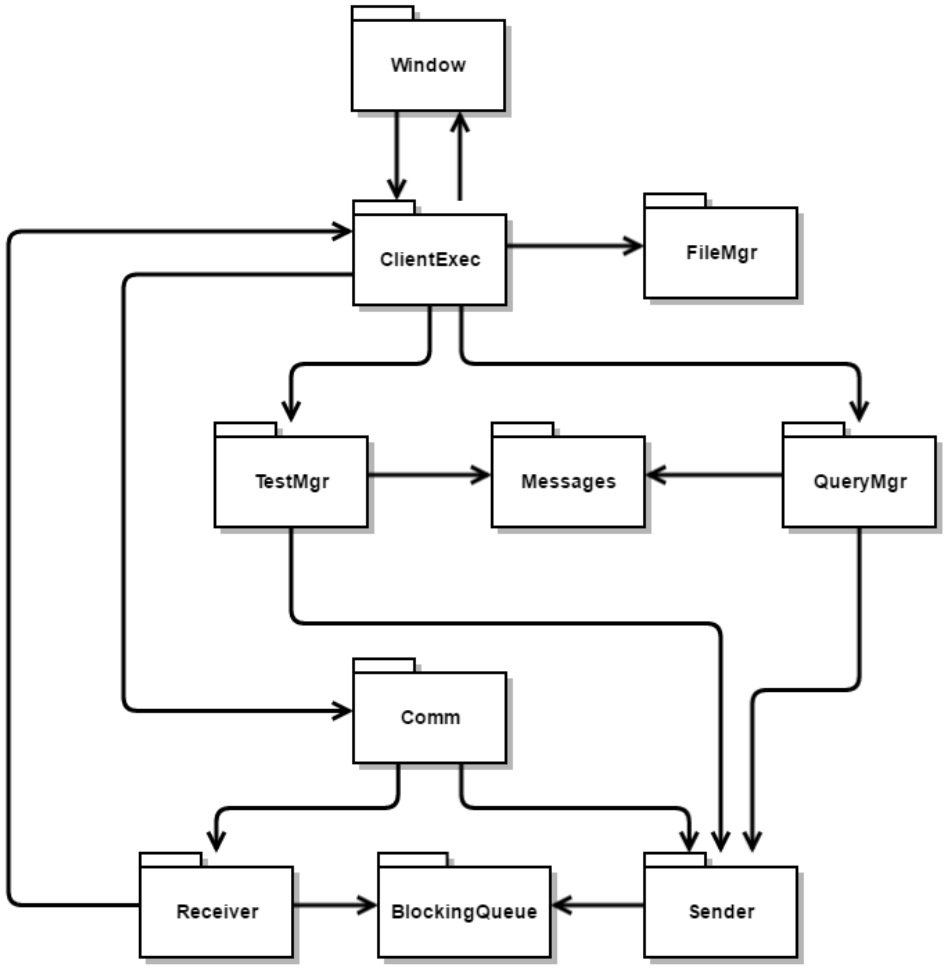


Collaboration System Client Activities

Package Diagrams

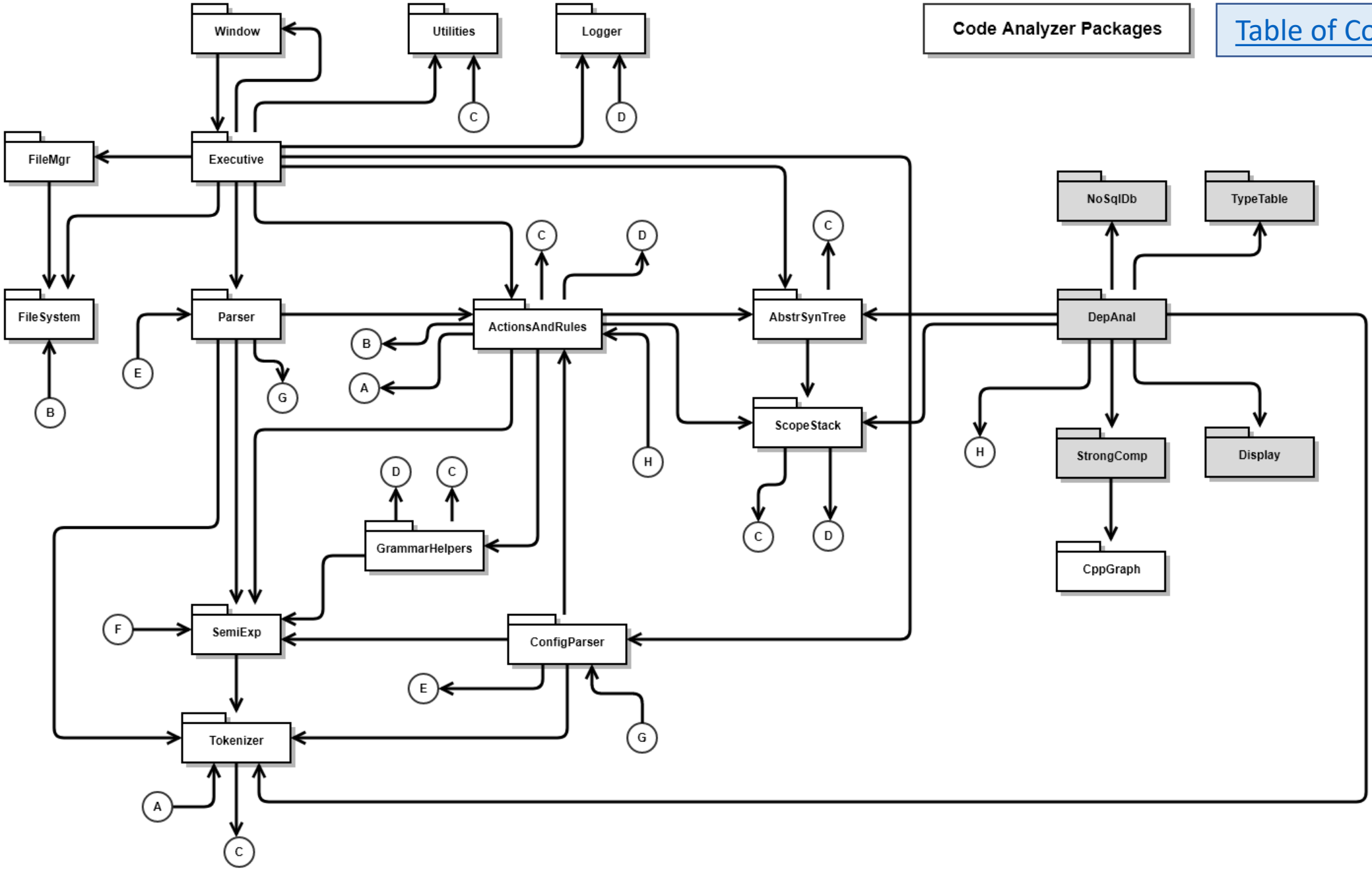
Table of Contents

- Enumerate each of the packages in a program or module.
- Show calling relationships with one-way directed lines.
- May show module boundaries.



Code Analyzer Packages

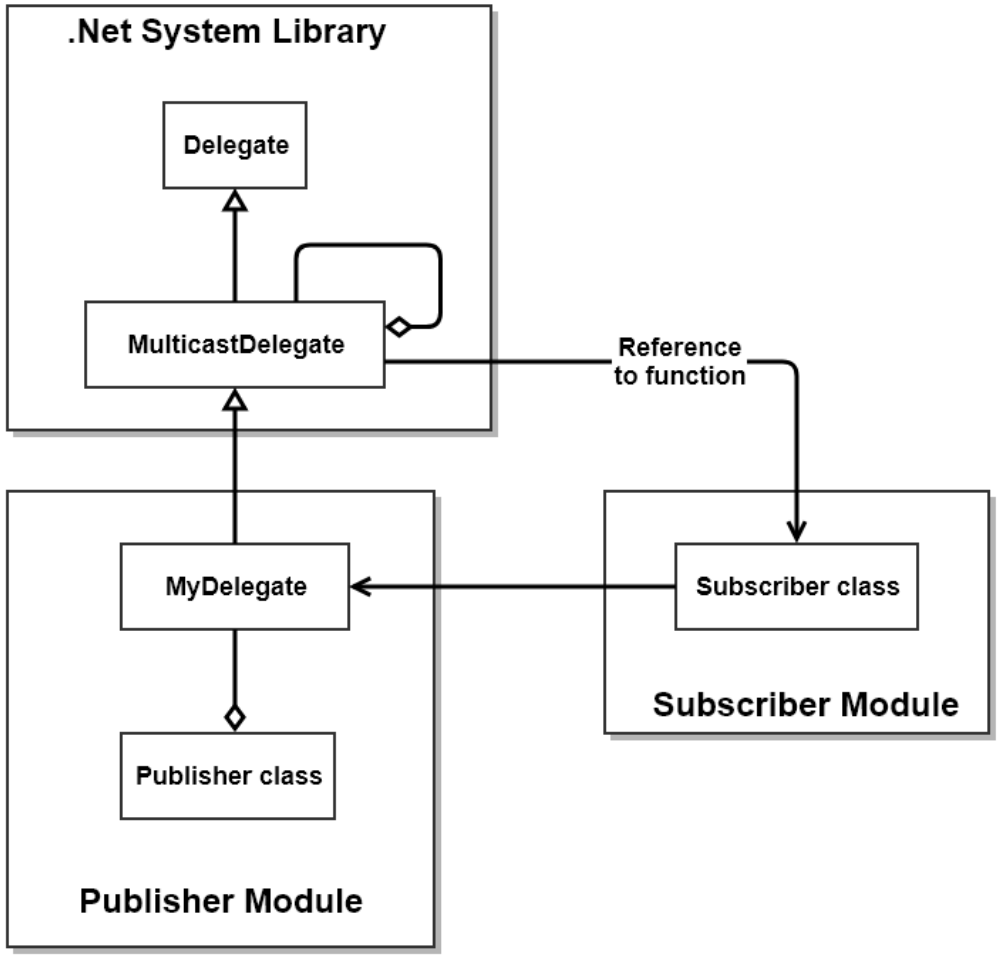
Table of Contents



Class Diagrams

Table of Contents

- Show each class with a named rectangle.
- Show class relationships:
  - Inheritance with a line beginning with a triangle attached to the base class.
  - Composition with a solid diamond attached to the composer
  - Aggregation with a hollow diamond attached to the aggregator
  - Using with a directed line from the user to the used class.
- May show module boundaries.
- May show methods and members



Parsing Facility

Table of Contents

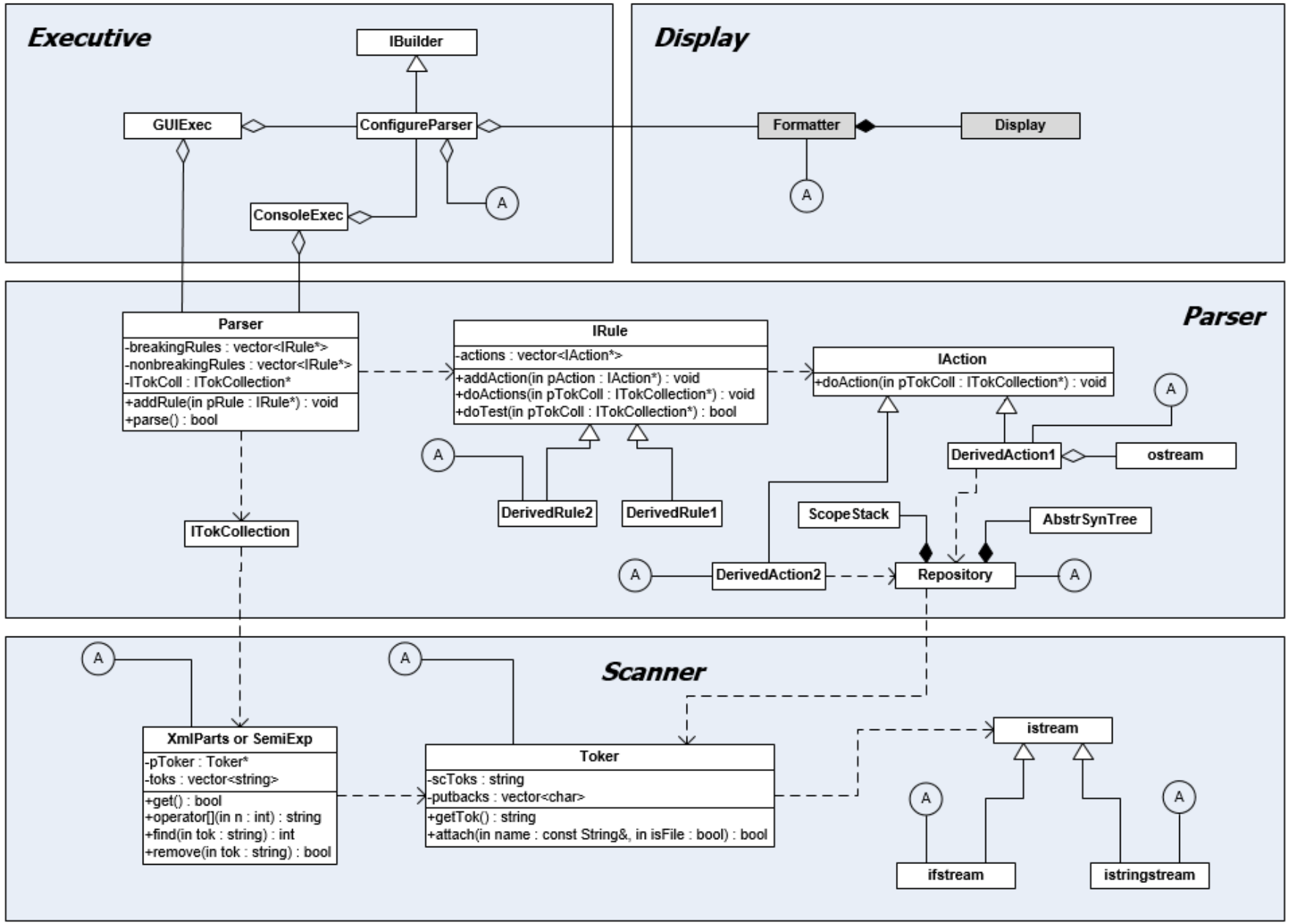
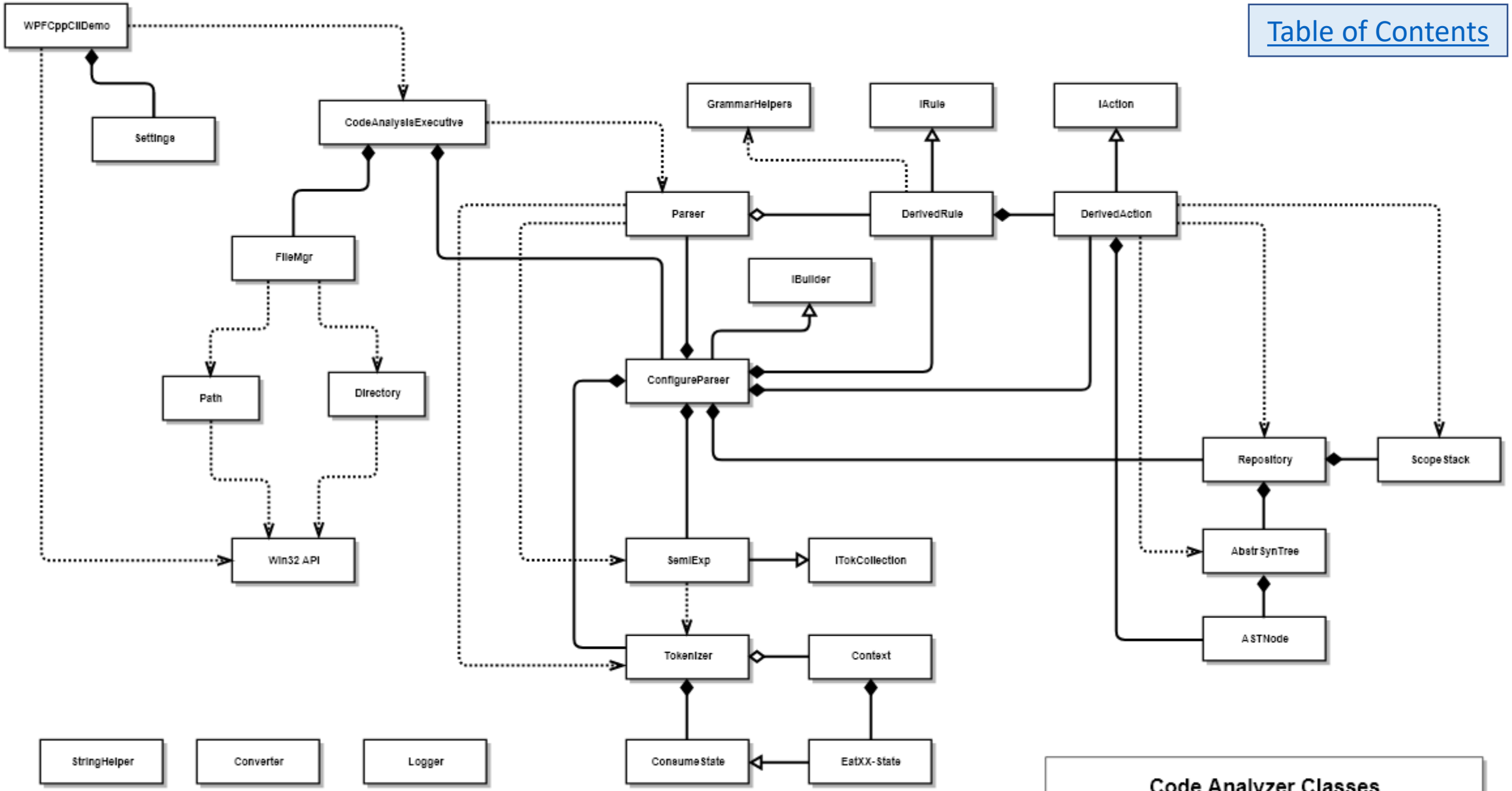


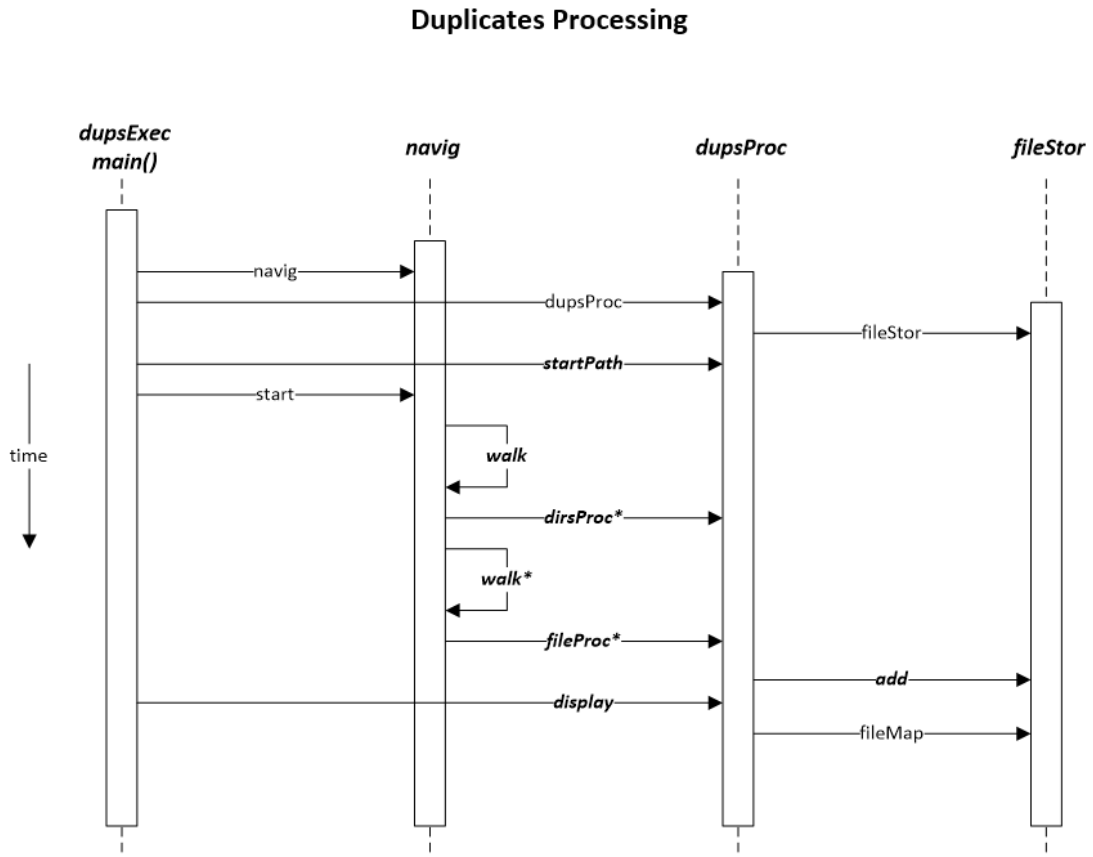
Table of Contents



Sequence Diagram

Table of Contents

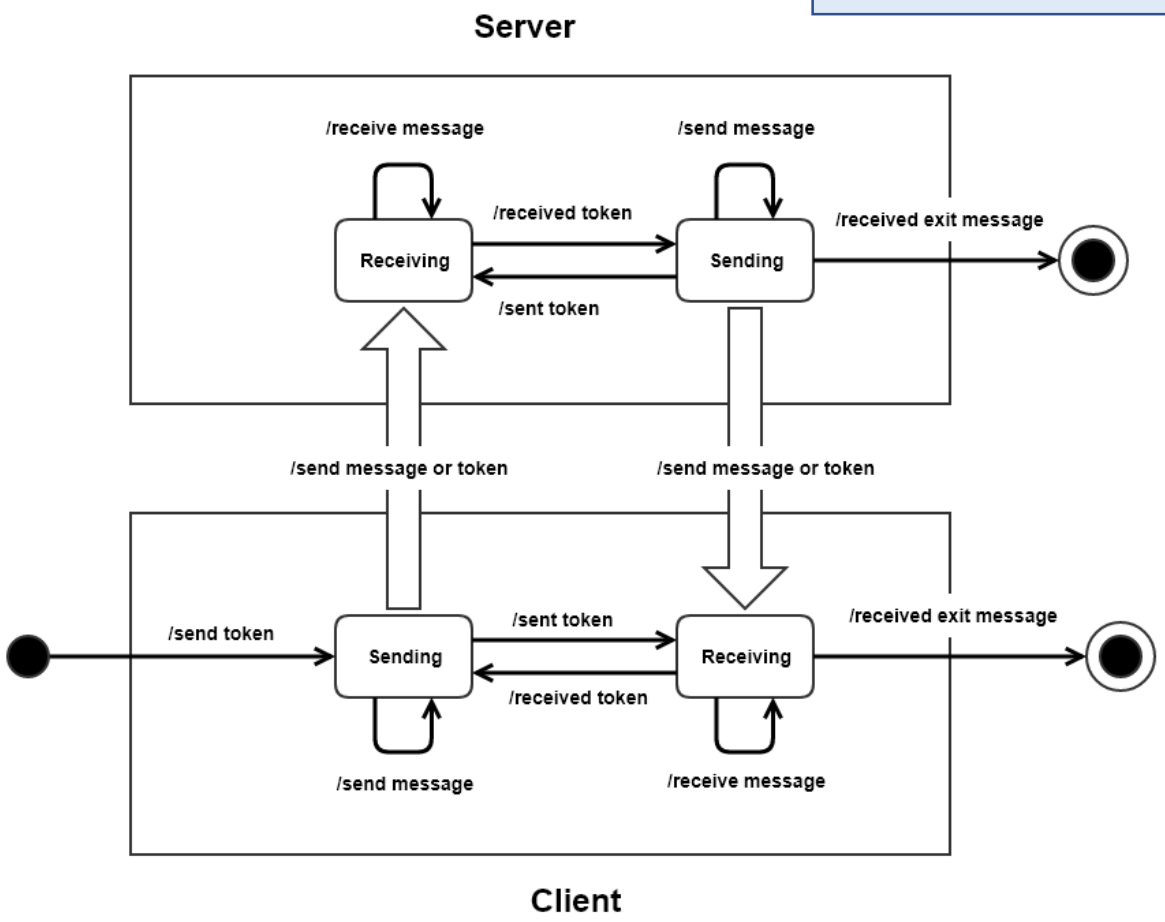
- Represents conversations between objects in a program.
- Each horizontal line is a method invocation. Text is the name of the method called.
- Increasing time flows downward in the diagram.
- Multiple calls are shown with an asterisk "*".
- Each bar represents lifetime of the named object.



State Diagram

Table of Contents

- Shows transitions between processing states.
- May have entrance and exit transitions.

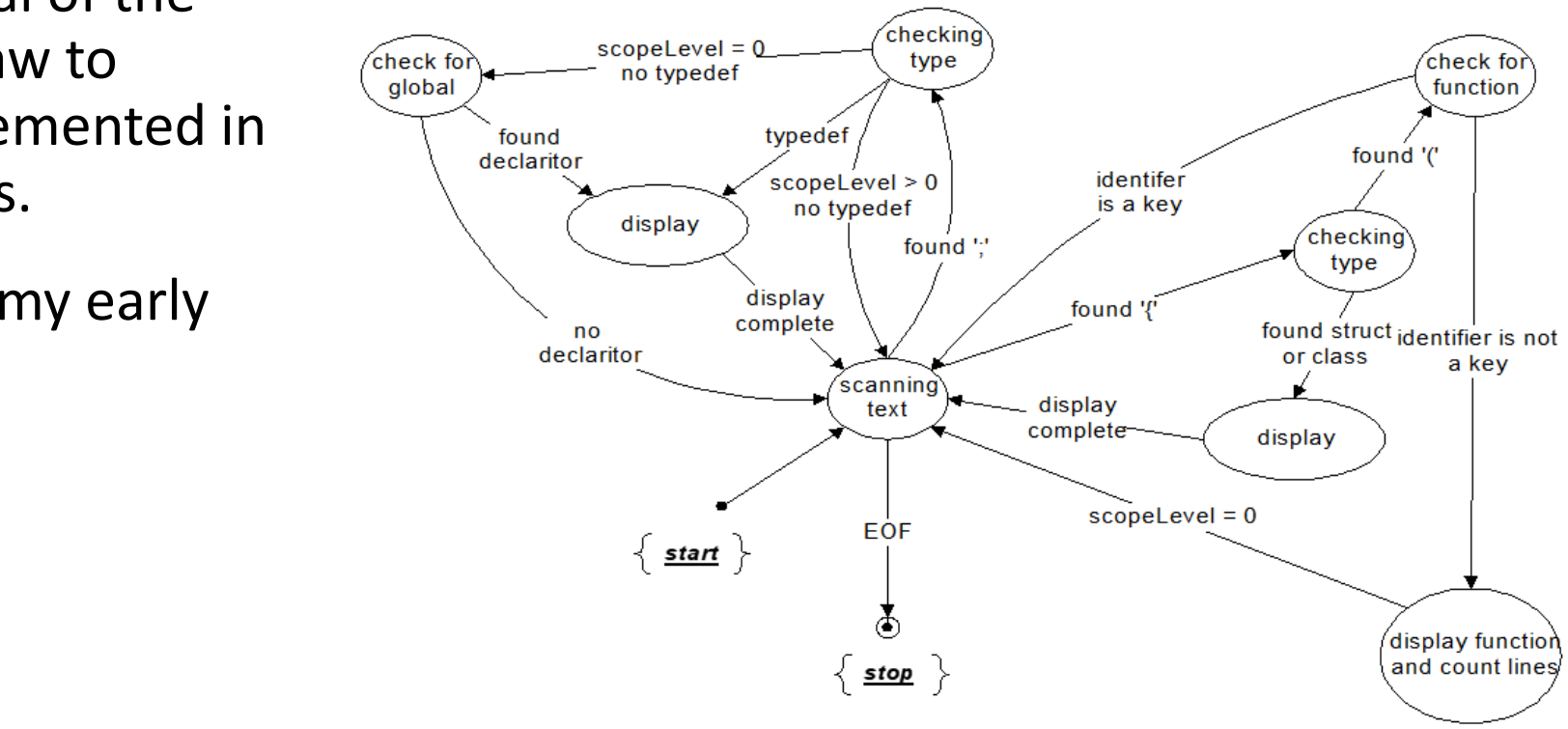


State Chart - Socket Bilateral Communication

Tokenizer State Diagram

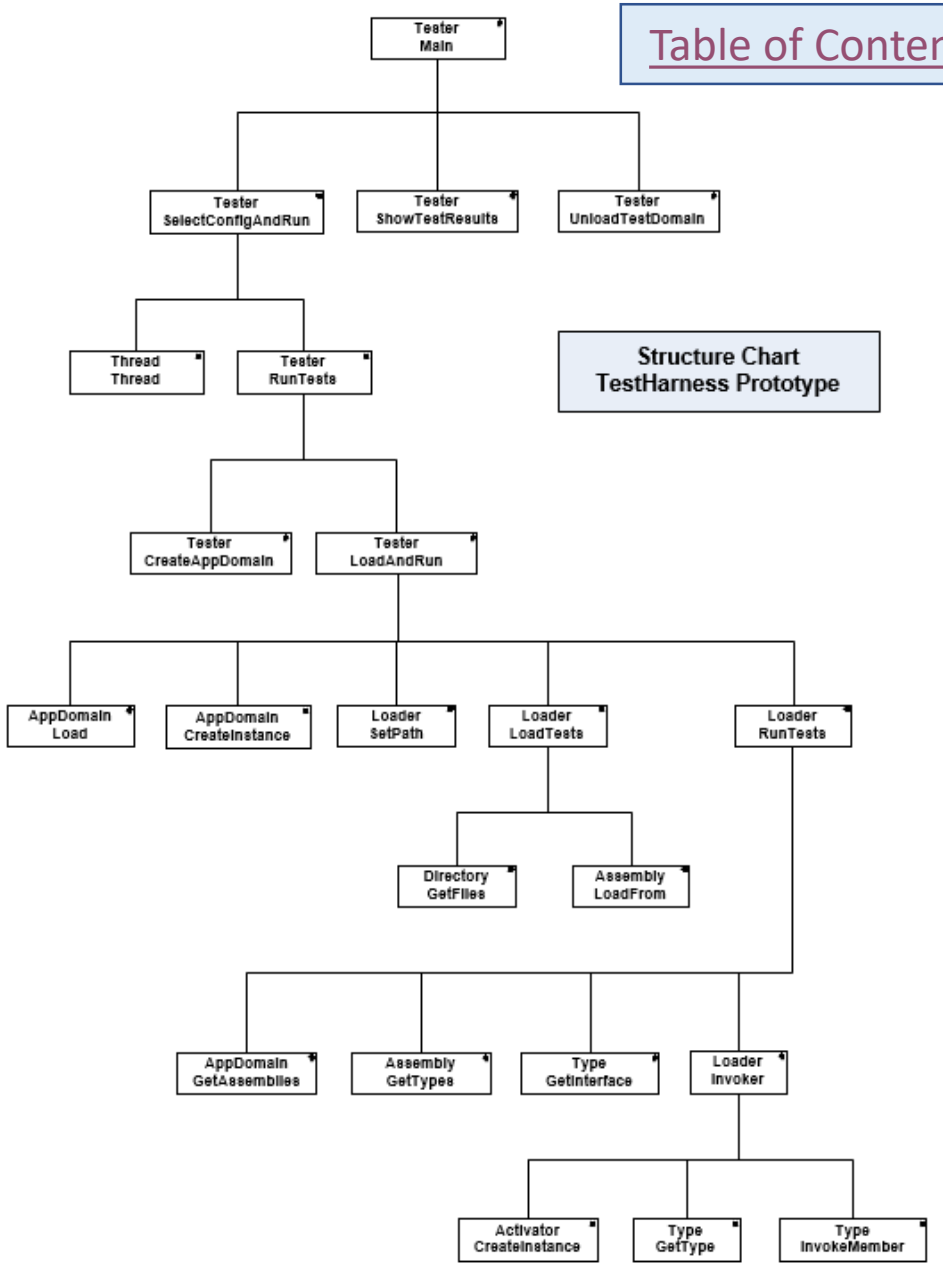
Table of Contents

- This example is typical of the diagrams you will draw to illustrate states implemented in one of your programs.
- It documents one of my early tokenizer designs.



Structure Chart (not UML)

- Shows function calling relationships.
- Used when there is a deep nesting of calling relationships.
- Used infrequently, but it is the best way to understand deep calling relationships.



Ad-Hoc Diagram

Table of Contents

- Ad-Hoc
  - Latin phrase “for this”
  - “designed for a specific problem or task”.
  - Both quotes from Wikipedia.
- A diagram made to suit one particular purpose.
- This diagram represents a parallel-pipelined software structure I used for code analysis.

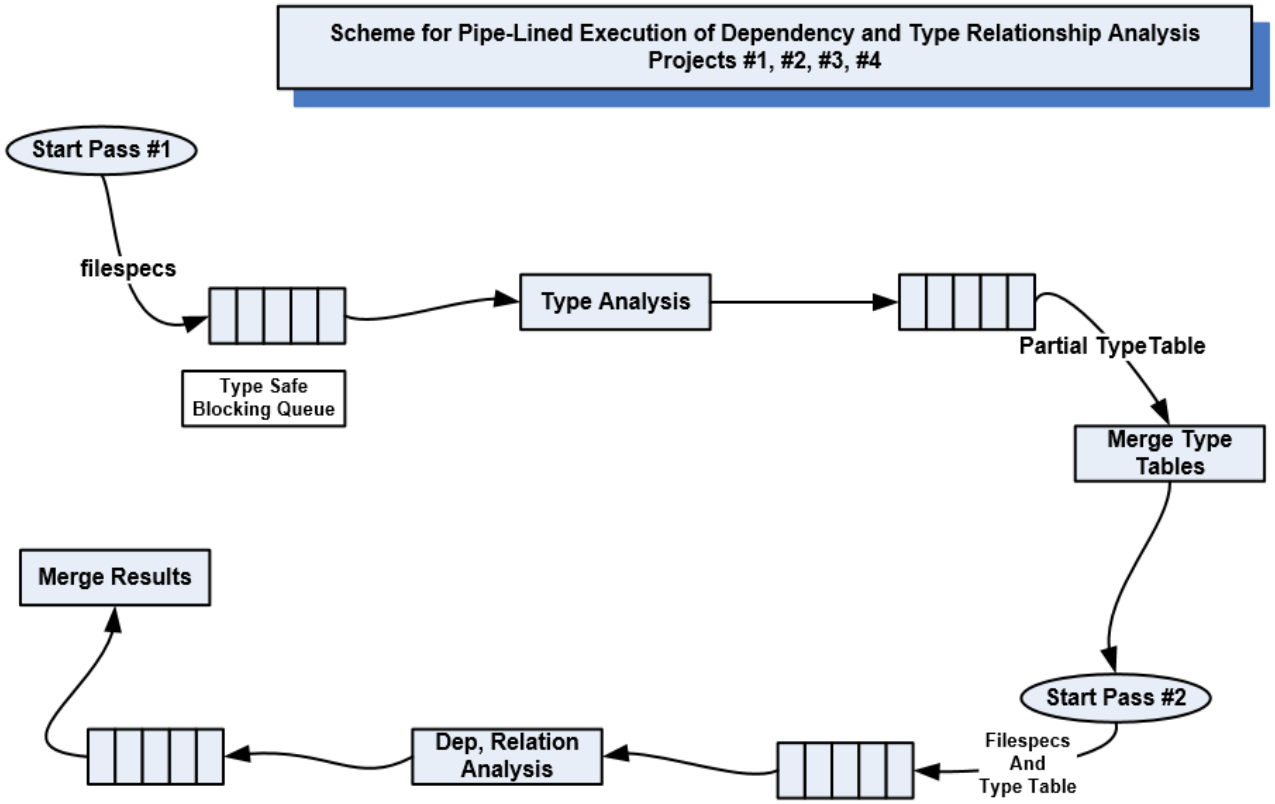
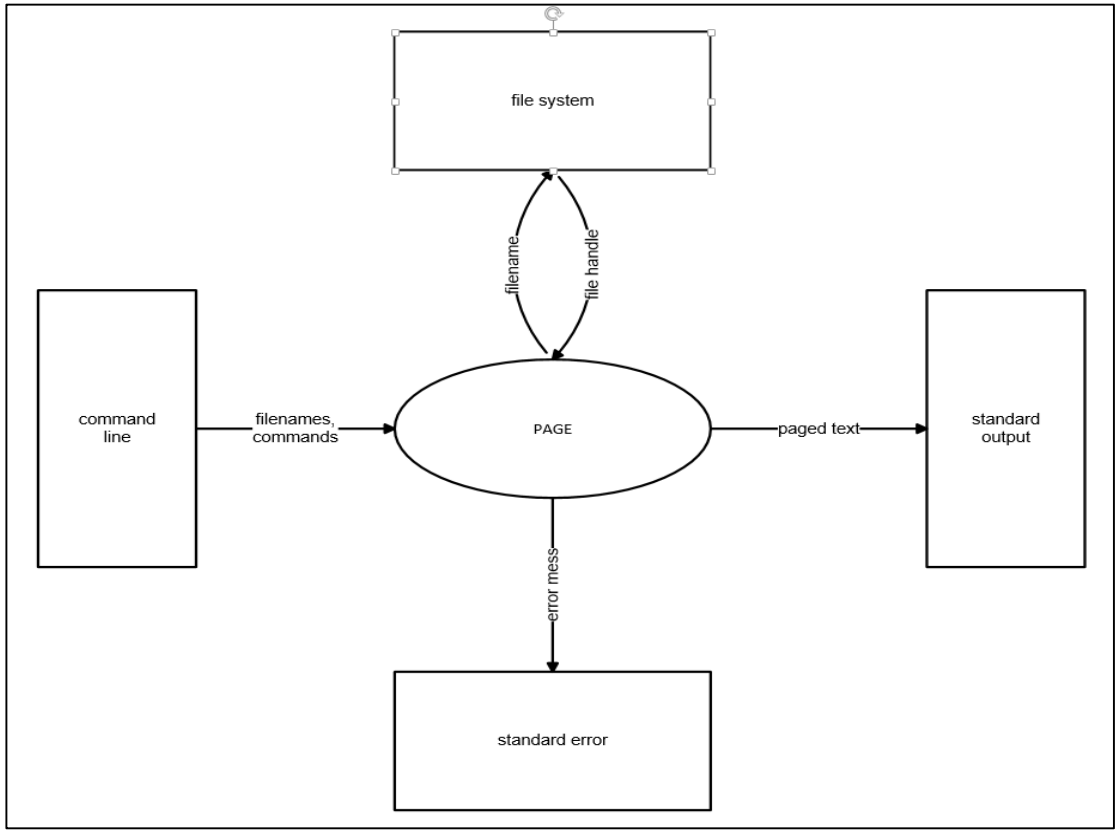


Table of Contents

Diagrams for Requirements

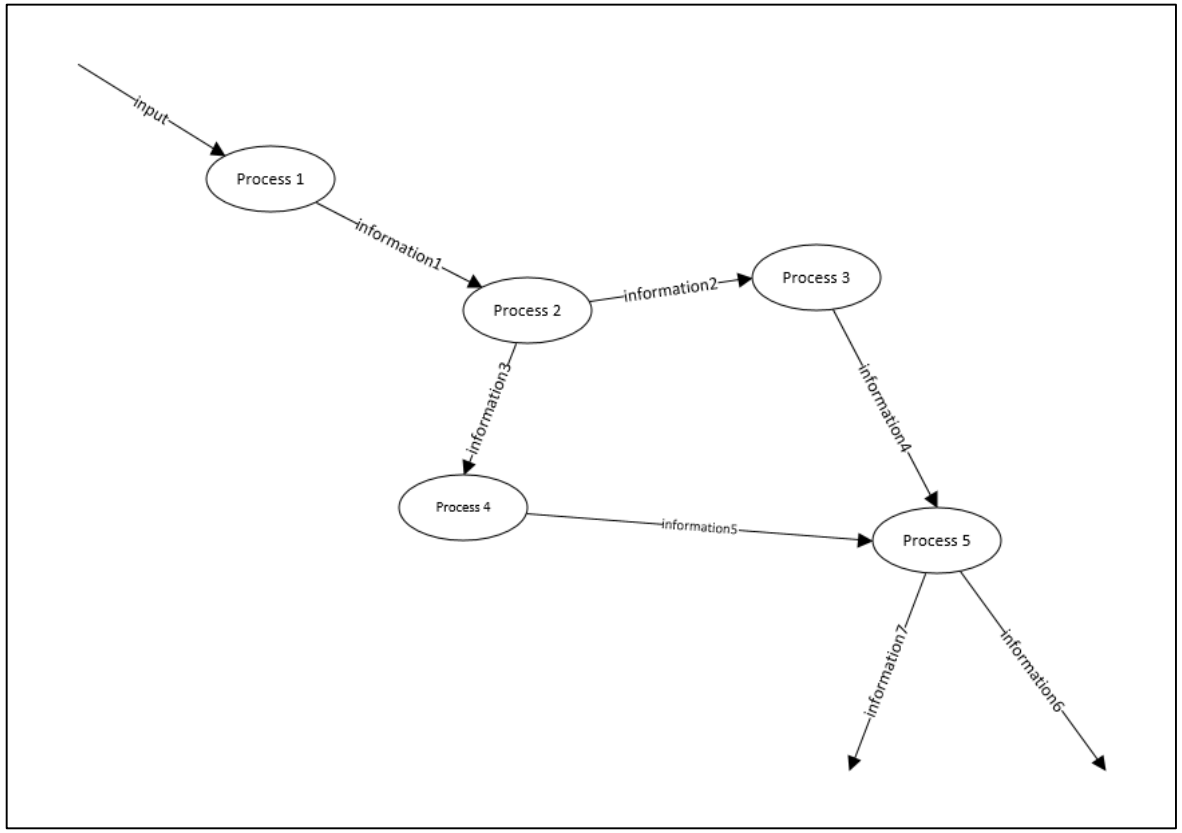
Context Diagram

- Shows relationships of program with its environment



Data Flow Diagram

- Shows information flow between processing blocks



The End

More discussion on a few of the diagrams here:

UML-Diagrams.htm