

SOFTWARE REQUIREMENTS SPECIFICATION

PICASSO

REQUIREMENTS ASSISTANT

Prepared by:

Jeffrey Fox
Heather Huber
David Krum
Jay Moon
Dong Ouyang
Insuk Sickler
Julie Vo

Table of Contents

1. SCOPE.....	3
1.1 IDENTIFICATION	3
1.2 CSCI OVERVIEW.....	3
1.3 DOCUMENT OVERVIEW	6
2. APPLICABLE DOCUMENTS.....	7
2.1 GOVERNMENT DOCUMENTS.....	7
2.2 NON_GOVERNMENT DOCUMENTS.....	7
3. ENGINEERING REQUIREMENTS.....	8
3.1 EXTERNAL INTERFACE DESCRIPTION	8
3.1.1 <i>Picasso Inputs</i>	9
3.1.2 <i>Picasso Outputs</i>	10
3.2 PROCESS LIST	12
3.2.1 <i>Process Picasso</i>	12
3.3 INTERNAL INTERFACE DESCRIPTION	38
3.4 CSCI DATA ELEMENT REQUIREMENTS	40
3.4.1 <i>Data Dictionary List</i>	40
3.4.2 <i>Data Dictionary Hierarchy</i>	71
3.5 ADAPTATION REQUIREMENTS	80
3.5.1 <i>Site Adaptation</i>	80
3.5.2 <i>Portability</i>	80
3.5.3 <i>Maintainability</i>	80
3.6 SIZING AND TIMING REQUIREMENTS.....	81
3.6.1 <i>Sizing</i>	81
3.6.2 <i>Timing</i>	81
3.7 SAFETY REQUIREMENTS.....	82
3.8 SECURITY REQUIREMENTS.....	82
3.9 DESIGN CONSTRAINTS	82
3.10 SOFTWARE QUALITY FACTORS	83
3.11 HUMAN PERFORMANCE/HUMAN ENGINEERING REQUIREMENTS	83
3.12 RELIABILITY REQUIREMENTS.....	83
3.12.1 <i>Robustness</i>	83
3.12.2 <i>Data Integrity</i>	83
3.13 TRACEABILITY MATRIX.....	84
3.13.1 <i>Governing Requirements Traced to Engineering Requirements</i>	84
3.13.2 <i>Engineering Requirements Traced to Governing Requirements</i>	94
4. QUALIFICATION REQUIREMENTS.....	110
5. PREPARATION FOR DELIVERY	110
6. NOTES	111
6.1 GLOSSARY	111
6.2 JUSTIFICATION FOR USING CONCEPTUAL GRAPHS.....	125
6.3 JUSTIFICATION FOR USERS COMMUNICATING DIRECTLY WITH EXTERNAL TOOLS	126
6.4 CONCEPTUAL GRAPH TRANSLATION TABLES	127
6.5 JUSTIFICATION FOR SEPARATE EXTERNAL TOOLS IN THE CONTEXT LEVEL DATA FLOW DIAGRAM.....	132
6.6 AXIOM CASE TOOL.....	133

Table of Figures

FIGURE 1 SYSTEM DATA FLOW DIAGRAM	4
FIGURE 2 CONTEXT LEVEL DATA FLOW DIAGRAM	8
FIGURE 3 PICASSO LEVEL 1 DATA FLOW DIAGRAM.....	12
FIGURE 4 RECEIVE PICASSO_USER_REQUEST R-NET	15
FIGURE 5 EXTERNAL_TOOLS_CONFIGURATION STORE	17
FIGURE 6 REQUIREMENTS_VIEW_DATA STORE	18
FIGURE 7 TYPE_HIERARCHY_DATA STORE.....	19
FIGURE 8 METRICS_LOG STORE.....	20
FIGURE 9 MANAGE_EXTERNAL_TOOLS LEVEL 2 DATA FLOW DIAGRAM	21
FIGURE 10 MANAGE_REQTS_VIEWS LEVEL 2 DATA FLOW DIAGRAM	30

List of Tables

TABLE 3-1 OPERATION TIME PARAMETERS.....	82
TABLE 6-1 TRANSLATION TABLE OBJECT-ORIENTED	127
TABLE 6-2 TRANSLATION TABLE DATA-ORIENTED	128
TABLE 6-3 TRANSLATION TABLE FUNCTION-ORIENTED (DATAFLOW)	129
TABLE 6-4 TRANSLATION TABLE STATE-ORIENTED (STD).....	130
TABLE 6-5 TRANSLATION TABLE STATE-ORIENTED (R-NETS).....	131

1. Scope

1.1 Identification

The software to be developed will be identified as Picasso.

1.2 CSCI Overview

Picasso will function as a major component of the Requirements Assistant system. This system is being developed to provide an environment in which a group of developers can collaborate on the production of a set of software requirements. These developers may be working on different continents and may not be able to meet in person. The system will support a multiple-viewed strategy that allows developers to create Requirements_Views using the CASE tools with which they are familiar. Views that have been created will be stored in a common repository for future reference. In addition to providing a storage and retrieval facility, the system will provide facilities to analyze for inconsistency, incompleteness, and ambiguity between the different views. As a result, the translation of views from the respective CASE tool format into a common internal representation is required. The notation of conceptual graphs is a well-defined knowledge representation which has been chosen as a suitable internal representation for the system [SOW97]. A description of the reasons for choosing the conceptual graphs notation is presented in the notes section of this document. The system will provide facilities to resolve the conflicts detected between the views. Finally, the system will track activity in the system by keeping a log of user requests, problem reports, and corrective actions that will be used to calculate and report project metrics.

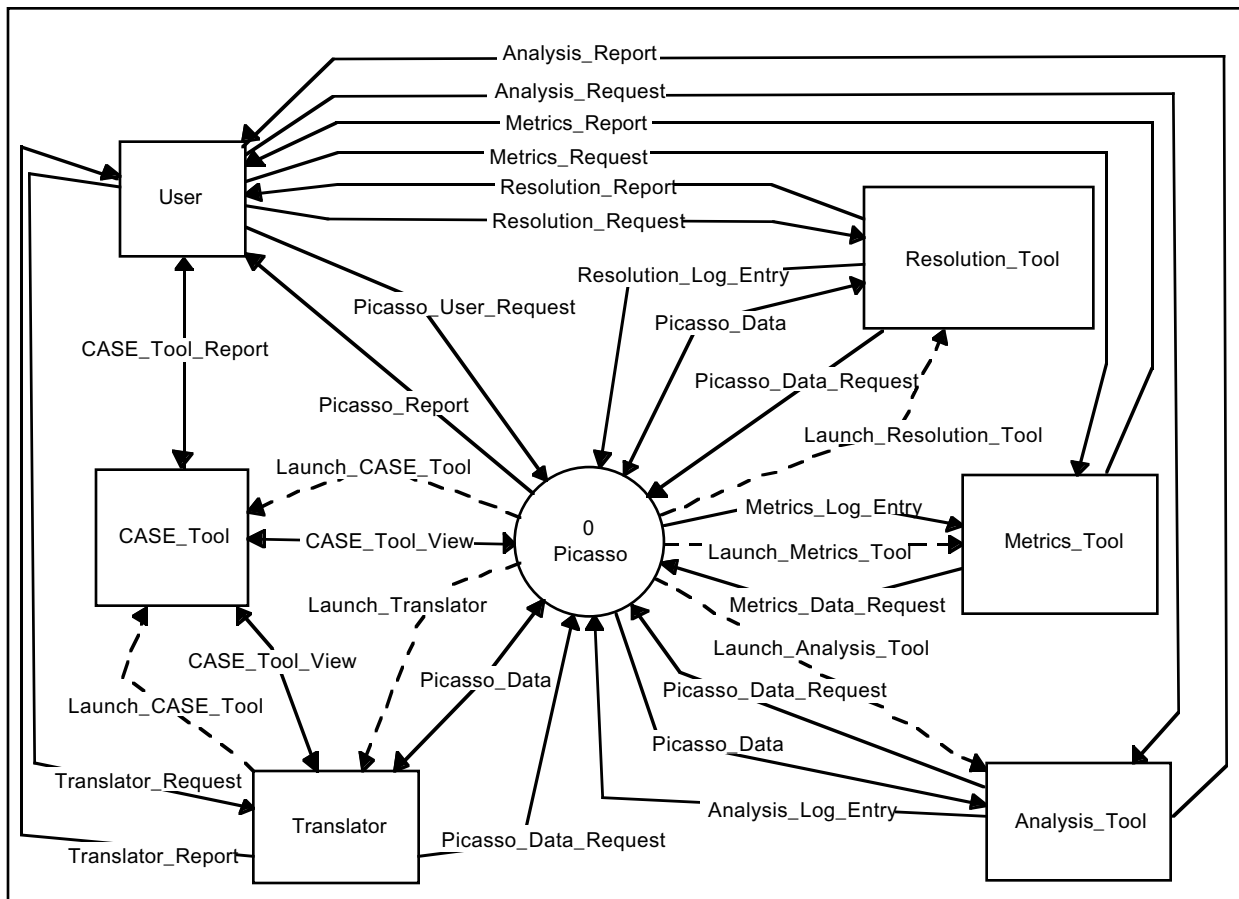


Figure 1 System Data Flow Diagram

The seven major components of the system will be the users, CASE tools, translators, analysis tools, resolution tools, metrics tools, and Picasso. The system's flexibility will allow the developers to experiment with different tools for analysis, problem resolution, and Metrics_Reporting. Each of these components are described with enough detail to provide the reader with sufficient background information to understand how the Picasso software fits into the system.

Users will include individuals (e.g. end users, designers, customers, developers, etc.) who have a stake in ensuring that their software will meet its goals. The users will have a basic knowledge of computers, including the Windows 95 operating system. They will also be familiar with at least one CASE tool to use to create the Requirements_Views. Users will be the primary source of stimuli to the system. The users will send requests to the other system components (Picasso and any external tools). These components will attempt to fulfill the users' requests and provide reports back to the user. The reasoning for having the user communicate directly with the external tools is described in the notes section of this document.

The term *CASE tool* normally refers to any software tool used for computer-aided software engineering. For the remainder of this document, the term CASE tool will refer to any software tool specifically used for computer-aided software requirements development. These CASE

tools may use data-oriented, object-oriented, function-oriented, or state-oriented methodologies. The user will employ CASE tools to create new Requirements_Views and view Requirements_Views previously stored in the system.

Translators have two components: input translators and output translators. Input translators are the software tools that retrieve a Requirements_View created by a CASE tool, generate a conceptual graph, and send this conceptual graph to Picasso for storage. Output translators are the software tools that allow users to retrieve the conceptual graph representation of a view from Picasso, generate a Requirements_View for the appropriate CASE tool, and launch that CASE tool to display the Requirements_View. A translator will be able to function as both an input translator and an output translator. The notes section of this document contains translation tables for common types of data-oriented, object-oriented, function-oriented, and state-oriented CASE tools.

Analysis tools are the software tools that retrieve the conceptual graph representation of multiple Requirements_Views and the necessary Type_Hierarchy_Data, check for consistency, completeness, and ambiguity between the views, and send problem reports to the users. For the remainder of this document, the term *inconsistency* will refer to conflicting requirements between views. This may include conflicts such as differences in the software's functionality, timing, non-functional aspects, and data item descriptions. For the remainder of this document, the term *incompleteness* will apply to information that is contained in one view, but is missing in another. For the remainder of this document, the term *ambiguity* will apply to differences between views that are not necessarily due to inconsistency or incompleteness. In addition to performing analysis functions, the analysis tools will log each problem found by sending an Analysis_Log_Entry to Picasso.

Resolution tools are the software tools that correct problems found during analysis. They retrieve the conceptual graph representation of a view and/or Type_Hierarchy_Data from Picasso, modify this information, and send it back to Picasso for storage. In addition to performing problem resolution functions, the resolution tools will log each corrective action by sending a Resolution_Log_Entry to Picasso.

Metrics tools are the software tools that retrieve metrics entry logs (log entries from analysis activities, problem resolution activities, and user requests to Picasso) from Picasso, calculate various project metrics, and generate Metrics_Reports.

The Picasso software will accept requests from the users, attempt to satisfy the requests, and send Confirmation_Reports and/or Error_Reports to the users. Picasso will log all user requests and the respective results. Upon a request to store a new Requirements_View, Picasso will retrieve the view from the CASE tool, launch the appropriate input translator, attempt to satisfy data requests from the translator, and store the Requirements_View. Upon a request to retrieve a CASE_Tool_View, Picasso will retrieve the view, launch the appropriate CASE tool, and send the Requirements_View to the CASE tool. Upon a request to retrieve a conceptual graph view, Picasso will launch the appropriate translator and attempt to satisfy data requests from the translator. Upon a request to perform analysis, Picasso will launch the appropriate tool, attempt to satisfy data requests from the analysis tool, and log problems reported by the analysis tool.

Upon a request to perform problem resolution, Picasso will launch the appropriate tool, attempt to satisfy data requests from the resolution tool, store the updated data, and log corrective actions reported by the resolution tool. Upon a request to report project metrics, Picasso will launch the appropriate tool and attempt to satisfy Metrics_Data_Requests from the metrics tool. Upon a request to add a new tool to the External_Tool_Configuration, Picasso will store the information for the new tool in the External_Tool_Configuration data store. Upon a request to delete a tool from the External_Tool_Configuration, Picasso will remove the tool from the External_Tool_Configuration data store.

1.3 Document Overview

The purpose of this document is to specify the requirements for development of the software known as Picasso. This document will function as an agreement between the customer and the developer that will establish both the basis for the rest of the software development process and the conditions for acceptance of the final product. This document will describe the external interfaces of the software, the processes in the software, and the internal interfaces between these processes. The external interfaces will be specified by a textual description of the inputs and outputs of the system, accompanied by a context level data flow diagram. The processes will be specified by a textual description combined with data flow diagrams and R-Nets. The internal interface will be specified by a textual description and the appropriate data flow diagrams. The document will specify the non-functional requirements of the system and will contain a data dictionary that includes a BNF and text description of every data item. Finally, a traceability matrix, a glossary, and an index will be included for readability purposes.

2. Applicable documents

2.1 Government documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of the specification shall be considered a superseding requirement.

STANDARDS:

Military MIL-STD-2167A, DI-MCCR-80025, Software Requirements Specification

2.2 Non_Government documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- [DAV93] Davis, Alan M., *Software Requirements: Objects, Functions, & States*, Prentice Hall PTR, Upper Saddle River, NJ, 1993.
- [DEL92] Delugach, Harry S., "Specifying Multiple-Viewed Software Requirements With Conceptual Graphs," *Jour. Systems and Software*, vol. 19, pp. 207-224, 1992.
- [FIN94] Finklestein, A., Gabbay D., Hunter, A., Kramer, J., and Nuseibeh, B., Inconsistency Handling in Multi-Perspective Specifications, *IEEE Transactions on Software Engineering*, 20(8), pp. 569-578, August 1994.
- [PRE97] Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, USA, 1997.
- [SOW84] Sowa, John F., *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA, 1984.
- [SOW97] Sowa, John F., *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, PWS Publishing Company, Boston, MA, 1997.

3. Engineering Requirements

3.1 External Interface Description

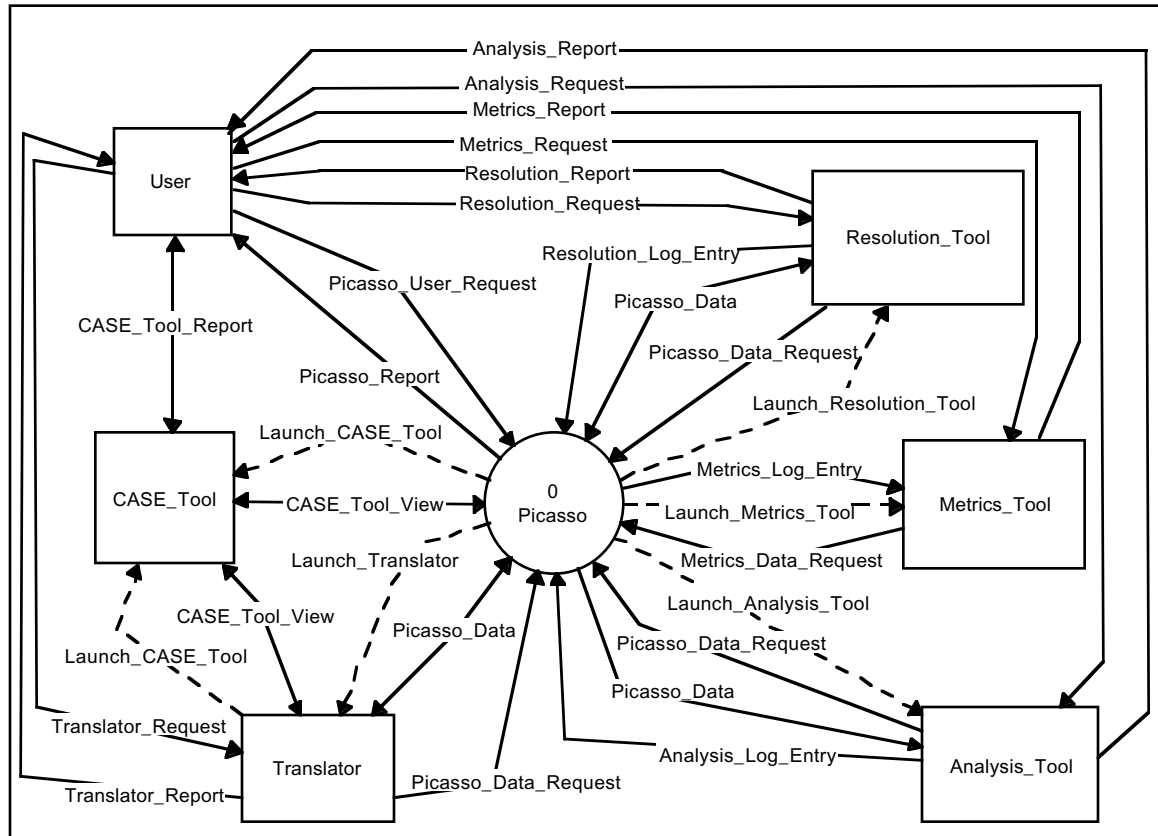


Figure 2 Context Level Data Flow Diagram

Picasso will accept a Picasso_User_Request as input from the user and generate a Picasso_Report back to the user indicating the success or failure of the request.

If the request is a Store_View_Request, Picasso will retrieve the CASE_Tool_View to be stored and issue a Launch_Translator Command. The Translator will issue a Picasso_Data_Request to Picasso for any necessary Picasso_Data from the system. After the translation is completed, the new Picasso_Data will be transmitted to Picasso to be stored.

If the request is a Retrieve View Request for an original CASE_Tool_View, Picasso will issue a Launch_CASE_Tool command and transmit the CASE_Tool_View to the appropriate CASE tool. If the request is a Retrieve View Request for CG_Objects, Picasso will issue a Launch_Translator command. Then, based on the user's parameters, the Translator will issue a Picasso_Data_Request for the desired Picasso_Data. After translation is completed, the translator will launch the appropriate CASE tool and send it to the CASE_Tool_View.

If the request is a `Launch_Analysis_Tool` Request, Picasso will issue a `Launch_Analysis_Tool` command. Once launched, the Analysis Tool will issue a `Picasso_Data_Request` for any desired `Picasso_Data`. Picasso will process the request and transmit the appropriate `Picasso_Data`. In addition, for every problem found, an `Analysis_Log_Entry` will be sent from the Analysis Tool to Picasso to be stored in the `Metrics_Log`.

If the request is a `Launch_Resolution_Tool` Request, Picasso will issue a `Launch_Resolution_Tool` command. Once launched, the Resolution Tool will issue a `Picasso_Data_Request` for any desired `Picasso_Data`. Picasso will process the request and transmit the appropriate `Picasso_Data`. Finally, any updated `Picasso_Data` will be transmitted back to Picasso. In addition, for every corrective `Action_Taken` a `Resolution_Log_Entry` will be sent from the Resolution Tool to Picasso to be stored in the `Metrics_Log`.

If the request is a `Launch_Metrics_Tool` Request, Picasso will issue a `Launch_Metrics_Tool` command. Once launched, the Metrics Tool will issue a `Metrics_Data_Request` for any desired `Metrics_Log_Entry` or group of entries. Picasso will process the request and transmit the appropriate `Metrics_Log_Entry` or group of entries.

If the request is a `Metrics_Data_Request` from the Metrics Tool, Picasso will perform a `Metrics_Data_Request` query, find the `Metrics_Log_Entry` that matches the request, and send the `Metric_Log_Entry` to `Metrics_Tool` for Metrics calculation.

3.1.1 Picasso Inputs

Picasso_User_Request

A `Picasso_User_Request` is data transmitted directly from the user to the Picasso software. It includes the user's location and one of the following: `Launch_Tool_Request`, `Store_View_Request`, `Retrieve CASE_Tool_View Request`, or `Update External_Tool_Configuration Request`.

CASE_Tool_View

The `CASE_Tool_View` is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the `Requirements_View` database. The original data can then be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary according to the CASE tool used to create it.

Picasso_Data

`Picasso_Data` includes any data the software provides in response to `Picasso_Data_Requests` from the translators, analysis tools, or resolution tools. This may include conceptual graph objects, `Type_Hierarchy_Data`, and `Requirements_View_Info`.

Picasso_Data_Request

A `Picasso_Data_Request` is a mechanism for the translators, analysis tools, and resolution tools to request `CG_Objects`, `Type_Hierarchy_Data`, and/or `Requirements_View_Info` from the Picasso software. The request may include any combination of these types of requests, which will result in a Join operation on the results.

Analysis_Log_Entry

An Analysis_Log_Entry contains the results of analysis that are to be stored in the Metrics_Log. This entry includes the User_Name, Analysis_Tool_Name, Problem_Type, and a textual Problem_Description.

Metrics_Data_Request

The Metrics_Data_Request is used by the external metrics tools to request data from the Metrics_Log. A Metrics_Data_Request may include the three following types of requests: Analysis_Data_Query, Resolution_Data_Query, or User_Command_Query. In addition, these requests may be combined together to allow the user to perform an Analysis_Data_Query and use its results in a Resolution_Data_Query.

Resolution_Log_Entry

Resolution_Log_Entries contain information about corrective actions taken by the problem Resolution Tools which will be stored in the Metrics_Log. It includes the problem resolution, Tool_Name, and Action_Taken.

3.1.2 Picasso Outputs

Picasso_User_Request

A Picasso_User_Request is data transmitted directly from the user to the Picasso software. It includes the user's location and one of the following: Launch_Tool_Request, Store_View_Request, Retrieve CASE_Tool_View Request, or Update External_Tool_Configuration Request.

CASE_Tool_View

The CASE_Tool_View is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

Picasso_Data

Picasso_Data includes any data the software provides in response to Picasso_Data_Requests from the translators, analysis tools, or resolution tools. This may include conceptual graph objects, Type_Hierarchy_Data, and Requirements_View_Info.

Picasso_Data_Request

A Picasso_Data_Request is a mechanism for the translators, analysis tools, and resolution tools to request CG_Objects, Type_Hierarchy_Data, and/or Requirements_View_Info from the Picasso software. The request may include any combination of these types of requests, which will result in a Join operation on the results.

Analysis_Log_Entry

An Analysis_Log_Entry contains the results of analysis that are to be stored in the Metrics_Log. This entry includes the User_Name, Analysis Tool_Name, Problem_Type, and a textual Problem_Description.

Metrics_Data_Request

The Metrics_Data_Request is used by the external metrics tools to request data from the Metrics_Log. A Metrics_Data_Request may include the three following types of requests: Analysis_Data_Query, Resolution_Data_Query, or User_Command_Query. In addition, these requests may be combined together to allow the user to perform an Analysis_Data_Query and use its results in a Resolution_Data_Query.

Resolution_Log_Entry

Resolution_Log_Entries contain information about corrective actions taken by the problem Resolution Tools which will be stored in the Metrics_Log. It includes the problem resolution, Tool_Name, and Action_Taken.

3.2 Process List

3.2.1 Process Picasso

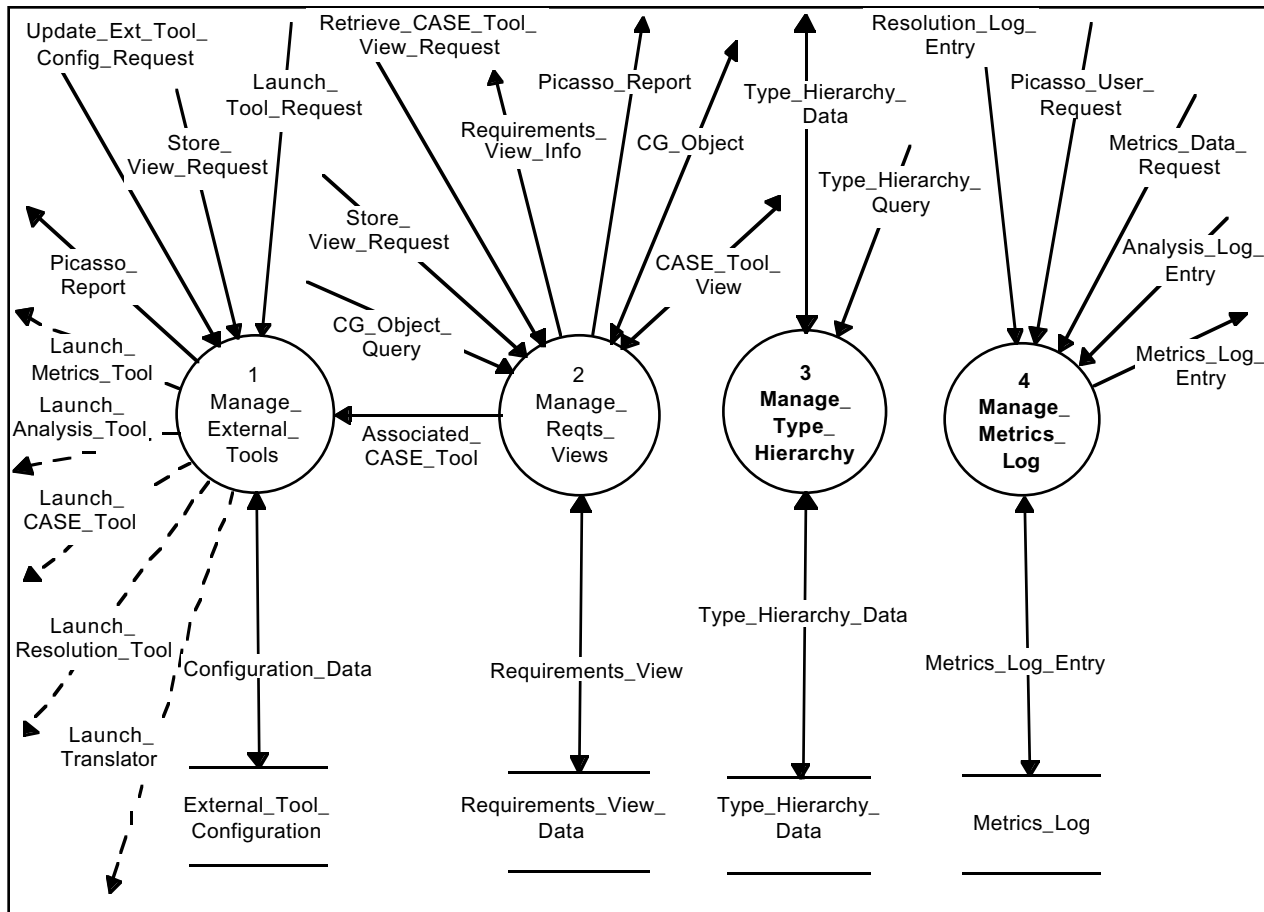


Figure 3 Picasso Level 1 Data Flow Diagram

Picasso will consist primarily of four data stores and the processes which control access to them. The External_Tool_Configuration data store will contain all of the information necessary to Launch_External_Tools. Access to the External_Tool_Configuration data store will be controlled by the Manage_External_Tools process. The Requirements_View_Data data store will contain all of the information related to previously stored Requirements_View(s). Access to the Requirements_View_Data data store will be controlled by the Manage_Reqs_Views process. The Type_Hierarchy_Data data store will contain a list of the conceptual graph Type entries and their parent/child relationships. Access to the Type_Hierarchy_Data data store will be controlled by the Manage_Type_Hierarchy process. The Metrics_Log data store will contain all of the data collected for Metrics_Reporting purposes. Access to the Metrics_Log data store will be controlled by the Manage_Metrics_Log process.

Upon receiving a Launch_Analysis_Tool Request, the Manage_Reqs_Views process will retrieve the Configuration_Data from the data store and issue the appropriate launch command:

Launch_Analysis_Tool for Analysis_Tool(s), Launch_Resolution_Tool for Resolution_Tool(s), Launch_Metrics_Tool for Metrics_Tool(s), and Launch_Translator for Output Translators. Upon receiving a Store_View_Request, the process will retrieve the Configuration_Data from the data store, and issue a Launch_Translator command for the appropriate Input_Translator. Upon receiving an Associated_CASE_Tool message from the Manage_Requirements_Views process, the Manage_External_Tools process will retrieve the Configuration_Data from the data store and issue a Launch_CASE_Tool command. Upon receiving a Update_External_Tool_Configuration_Request, the process will add or delete the Configuration_Data entry as requested.

Upon receiving a Store_View_Request, the Manage_Reqs_Views process will retrieve the CASE_Tool_View, wait for the CG_Object(s) from the translator, and store the Requirements_View in the data store. Upon receiving a CG_Object_Query from an external tool, the process will retrieve the Requirements_View(s) from the data store and send the requested CG_Object(s) to the tool. Upon receiving a Retrieve_CASE_Tool_View_Request, the process will retrieve the Requirements_View from the Requirements_Tool, send the Associated_CASE_Tool message to the Manage_External_Tools process, wait for the CASE tool to be launched, and then send the CASE_Tool_View to the CASE_Tool. Upon receiving CG_Object(s) from a Resolution_Tool, the process will store them as a new Requirements_View. In all cases, a Picasso_Report will be sent to the user to indicate the results.

Upon receiving a Type_Hierarchy_Query from an external tool, the Manage_Type_Hierarchy process will retrieve the Type_Hierarchy_Data from the data store and send it to the external tool. Upon receiving Type_Hierarchy_Data from a Resolution_Tool, the process will store the entries in the data store.

Upon receiving a Picasso_User_Request, the Maintain_Metrics_Log process will create a new Metrics_Log_Entry and store it in the data store. Upon receiving an Analysis_Log_Entry or a Resolution_Log_Entry, the process will store it as a Metrics_Log_Entry in the data store. Upon receiving a Metrics_Data_Request from a Metrics_Tool, the process will retrieve the Metrics_Log_Entry(s) from the data store and send the entry(s) to the Metrics_Tool.

Inputs:

Picasso_User_Request

A Picasso_User_Request is data transmitted directly from the user to the Picasso software. It includes the user's location and one of the following: Launch_Tool_Request, Store_View_Request, Retrieve_CASE_Tool_View_Request, or Update_External_Tool_Configuration_Request.

CASE_Tool_View

The CASE_Tool_View is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

Picasso_Data

Picasso_Data includes any data the software provides in response to Picasso_Data_Requests from the translators, analysis tools, or resolution tools. This may include conceptual graph objects, Type_Hierarchy_Data, and Requirements_View_Info.

Picasso_Data_Request

A Picasso_Data_Request is a mechanism for the translators, analysis tools, and resolution tools to request CG_Objects, Type_Hierarchy_Data, and/or Requirements_View_Info from the Picasso software. The request may include any combination of these types of requests, which will result in a Join operation on the results.

Analysis_Log_Entry

An Analysis_Log_Entry contains the results of analysis that are to be stored in the Metrics_Log. This entry includes the User_Name, Analysis_Tool_Name, Problem_Type, and a textual Problem_Description.

Metrics_Data_Request

The Metrics_Data_Request is used by the external metrics tools to request data from the Metrics_Log. A Metrics_Data_Request may include the three following types of requests: Analysis_Data_Query, Resolution_Data_Query, or User_Command_Query. In addition, these requests may be combined together to allow the user to perform an Analysis_Data_Query and use its results in a Resolution_Data_Query.

Resolution_Log_Entry

Resolution_Log_Entries contain information about corrective actions taken by the problem Resolution Tools which will be stored in the Metrics_Log. It includes the problem resolution, Tool_Name, and Action_Taken.

User Request Processing:

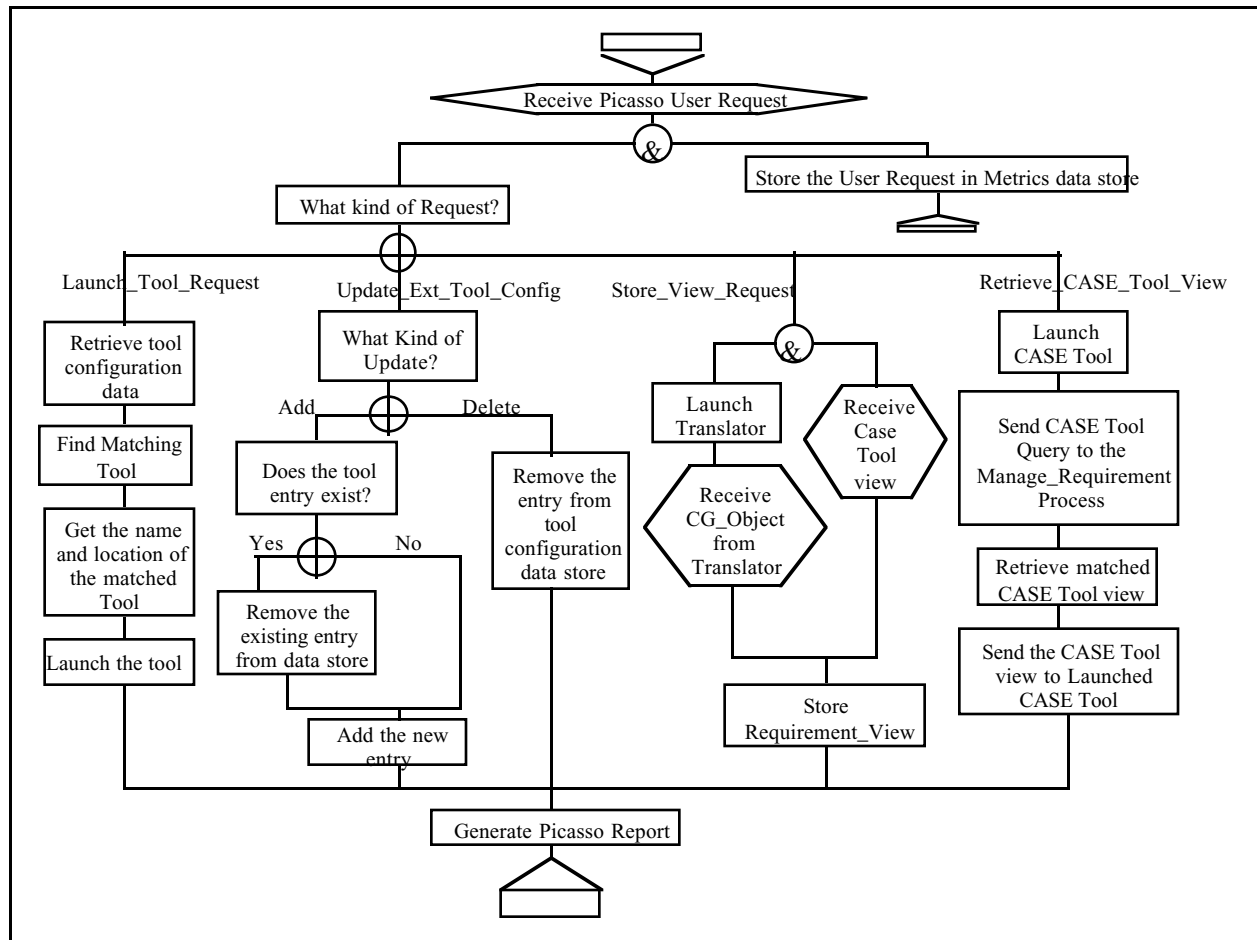


Figure 4 Receive Picasso_User_Request R-Net

Outputs:

Picasso_Report

This includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, warnings, beeps, etc.

CASE_Tool_View

The CASE_Tool_View is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

Picasso_Data

Picasso_Data includes any data the software provides in response to Picasso_Data_Requests from the translators, analysis tools, or resolution tools. This may include conceptual graph objects, Type_Hierarchy_Data, and Requirements_View_Info.

Metrics_Log_Entry

Metrics_Log_Entry is used as raw data for the metrics that the external metrics tool calculates. These metrics will be used to help describe how the system is being used, determine how well the current configuration of the system is performing, and measure the development process of a specific project. A Metrics_Log_Entry can be one of three different types: Analysis_Log_Entry, Resolution_Log_Entry, or User_Command_Entry. Every Metrics_Log_Entry has Date/Time_Info.

Launch_CASE_Tool

Launch_CASE_Tool is the control signal from Picasso used to launch the various CASE tools. The information on how to launch the CASE tools is stored in the External_Tool_Configuration data store. The CASE tool is used an external viewer for the CASE_Tool_Views.

Launch_Translator

Launch_Translator is the control signal from Picasso used to launch the various external translator tools. The information on how to launch the translator tools is stored in the External_Tool_Configuration data store.

Launch_Resolution_Tool

Launch_Resolution_Tool is the control signal from Picasso used to launch the various external problem resolution tools. The information on how to launch the resolution tools is stored in the External_Tool_Configuration data store.

Launch_Metrics_Tool

Launch_Metrics_Tool is the control signal from Picasso used to launch the various external metrics tools. The information on how to launch the metrics tools is stored in the External_Tool_Configuration data store.

Launch_Analysis_Tool

Launch_Analysis_Tool is the control signal from Picasso used to launch the various external analysis tools. The information on how to launch the analysis tools is stored in the External_Tool_Configuration data store.

Data Stores:

External_Tool_Configuration

The External_Tool_Configuration data store will contain all of the information necessary to Launch_External_Tools. This data store will consist of Configuration_Data entries that include a Tool_Name, Tool_Type, and Tool_Launch_Info. In addition, Translator entries will also include an Associated_CASE_Tool message. When launching an Analysis_Tool, Resolution_Tool, Metrics_Tool, or CASE tool, the Launch_External_Tools process will retrieve the Tool_Launch_Info by Tool_Name. When storing or retrieving Requirements_Views, the Launch_External_Tool process will retrieve the Tool_Launch_Info by Associated_CASE_Tool. The data store will be updated by add and Delete_Requests to the Update_Configuration_Data process.

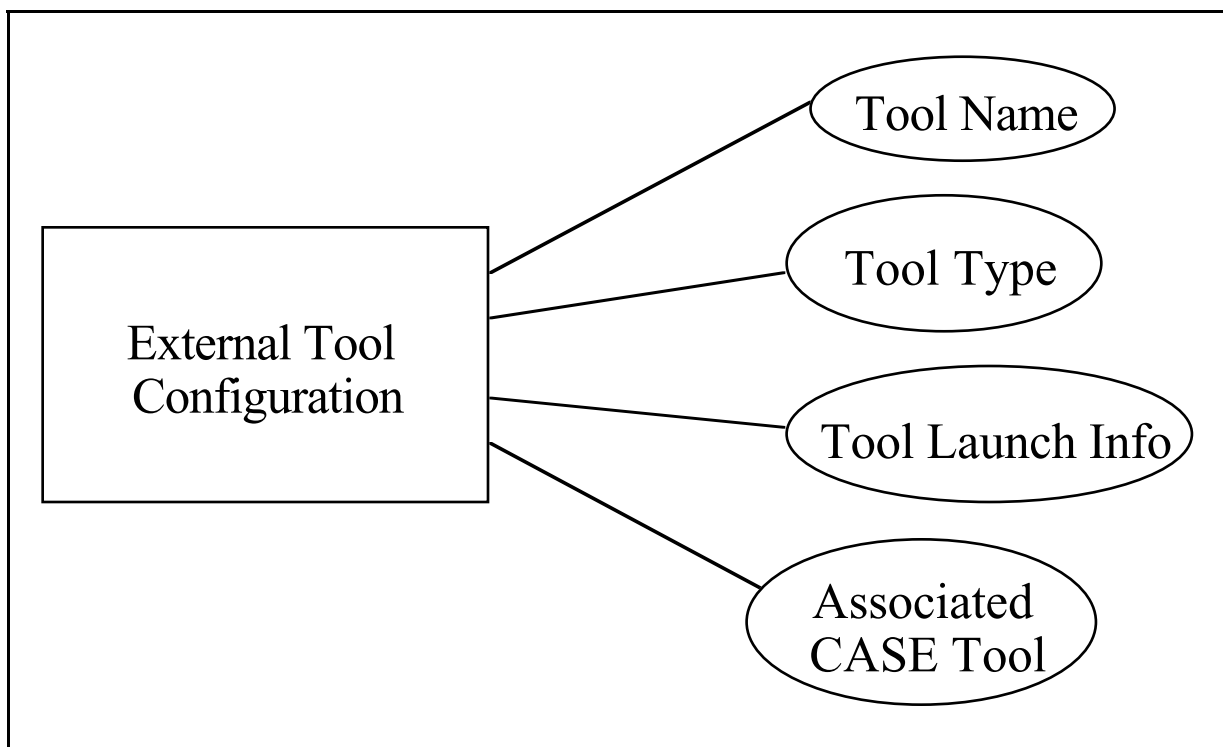


Figure 5 External_Tools_Configuration Store

Requirements_View_Data

The Requirements_View_Data data store will contain all of the information related to previously stored Requirements_View(s). This will include a CASE_Tool_View, Requirements_View_Info, and CG_Object(s). The CASE_Tool_View will be the original view as generated by the CASE tool. The Requirements_View_Info will include View_Name, Author, Date_Time_Info, Version, and Associated_CASE_Tool. The CG_Object(s) will be the conceptual graphs representation of the CASE_Tool_View as generated by the Input_Translator. The Store_View process will be used to add new Requirements_VIEWS to the data store. The Retrieve_Original_View process will be used to retrieve CASE_Tool_View(s) from the data store. The Retrieve_CG_Object_Data process will be used to Retrieve_CG_Objects from the data store.

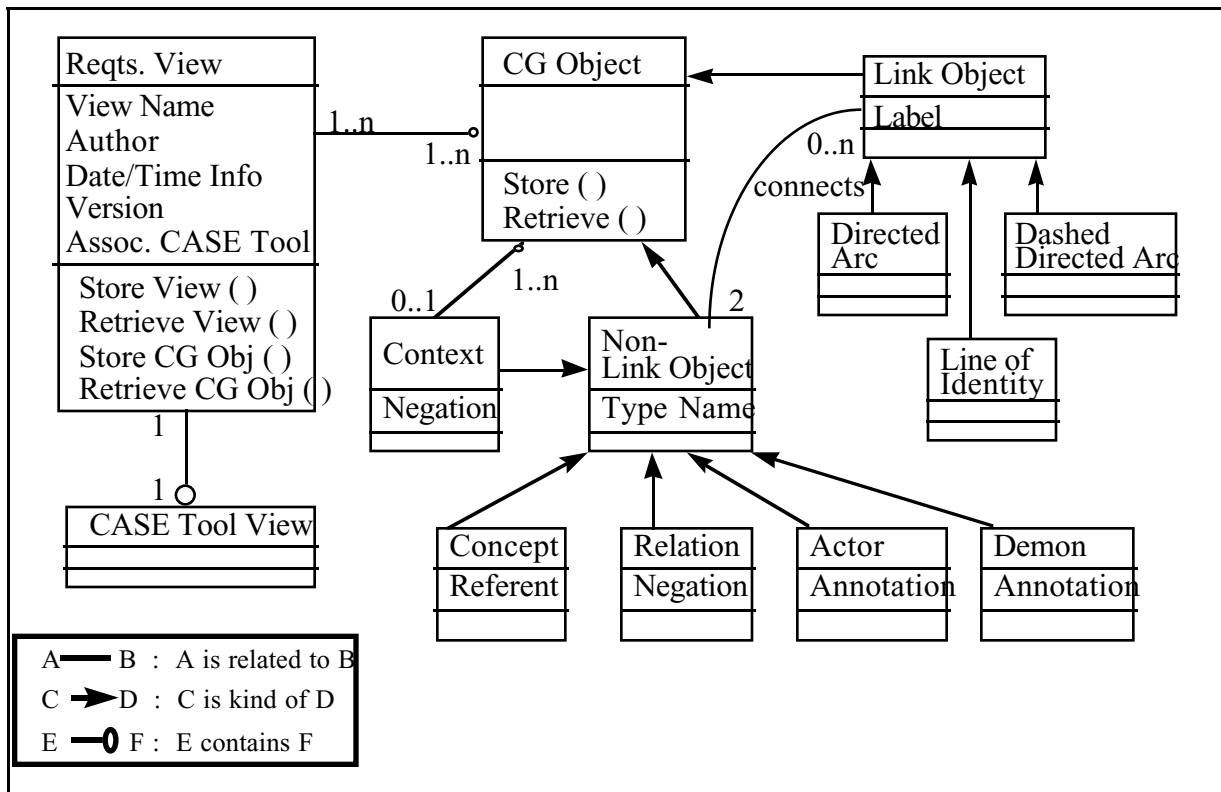


Figure 6 Requirements_View_Data Store

Type_Hierarchy_Data

The Type_Hierarchy_Data data store will contain a list of the conceptual graph Type entries and their parent/child relationships. Each Type entry will include a Type_Name, Date_Time_Info, and User_Name. The Manage_Type_Hierarchy process will be used to store new Type entries and retrieve Type_Hierarchy_Data from the data store.

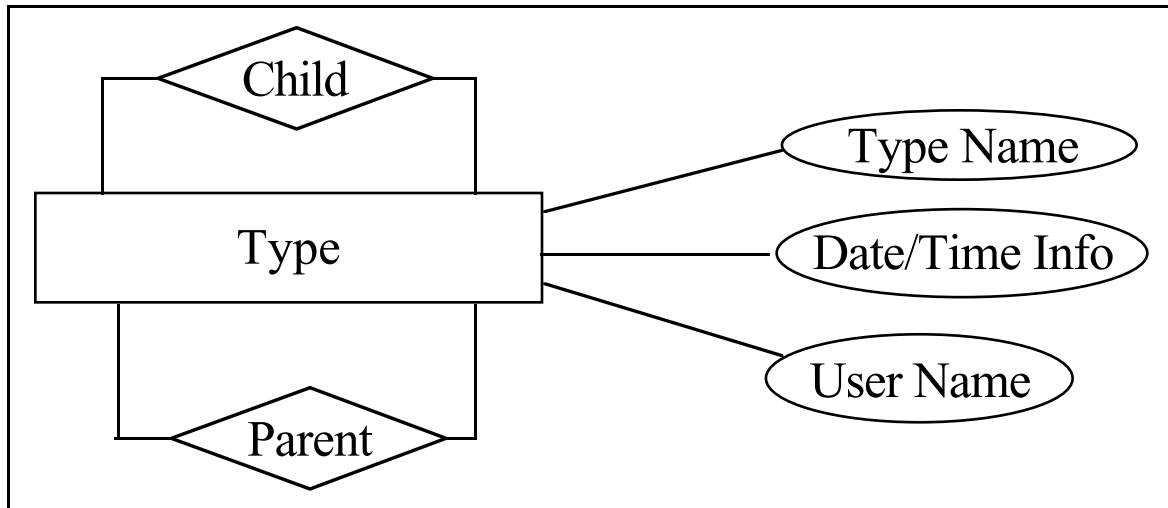


Figure 7 Type_Hierarchy_Data Store

Metrics_Log

The Metrics_Log data store will contain all of the data collected for Metrics_Reporting purposes. This data store will consist of three types of Metrics_Log_Entry(s): Analysis_Log_Entry(s), Resolution_Log_Entry(s), and User_Command_Entry(s). Every Metrics_Log_Entry will include Date_Time_Info and User_Name. Analysis_Log_Entry(s) will be used to store problem reports from the Analysis_Tool(s). Each entry will include a Tool_Name, Problem_Type, and Problem_Description. Resolution_Log_Entry(s) will be used to store corrective action reports from the Resolution_Tool(s). Each entry will include a Tool_Name and Action_Taken. User_Command_Entry(s) will be used to log all Picasso_User_Requests. Each entry will include a Command_Name and Command_Results. The Manage_Metrics_Log process will be used to store new entries and to retrieve entries from the data store.

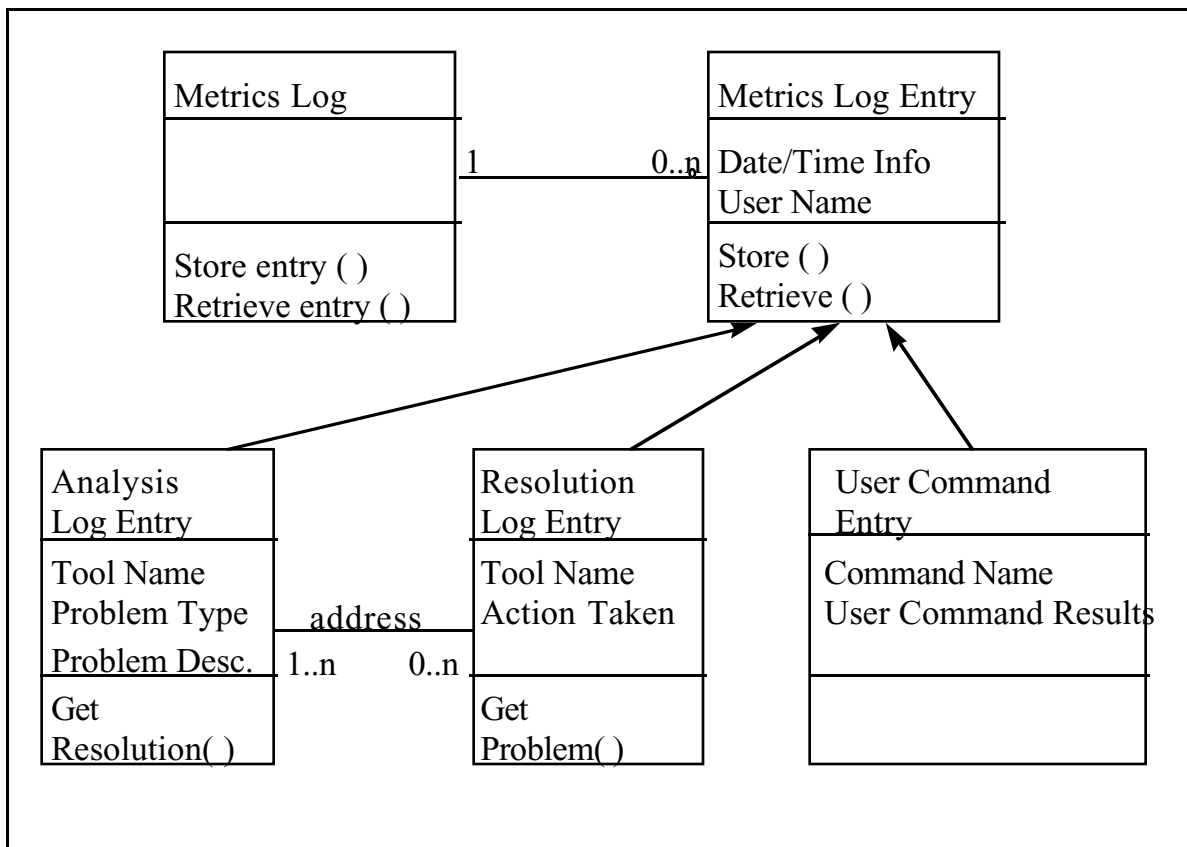


Figure 8 Metrics_Log Store

3.2.1.1 Process Manage_External_Tools

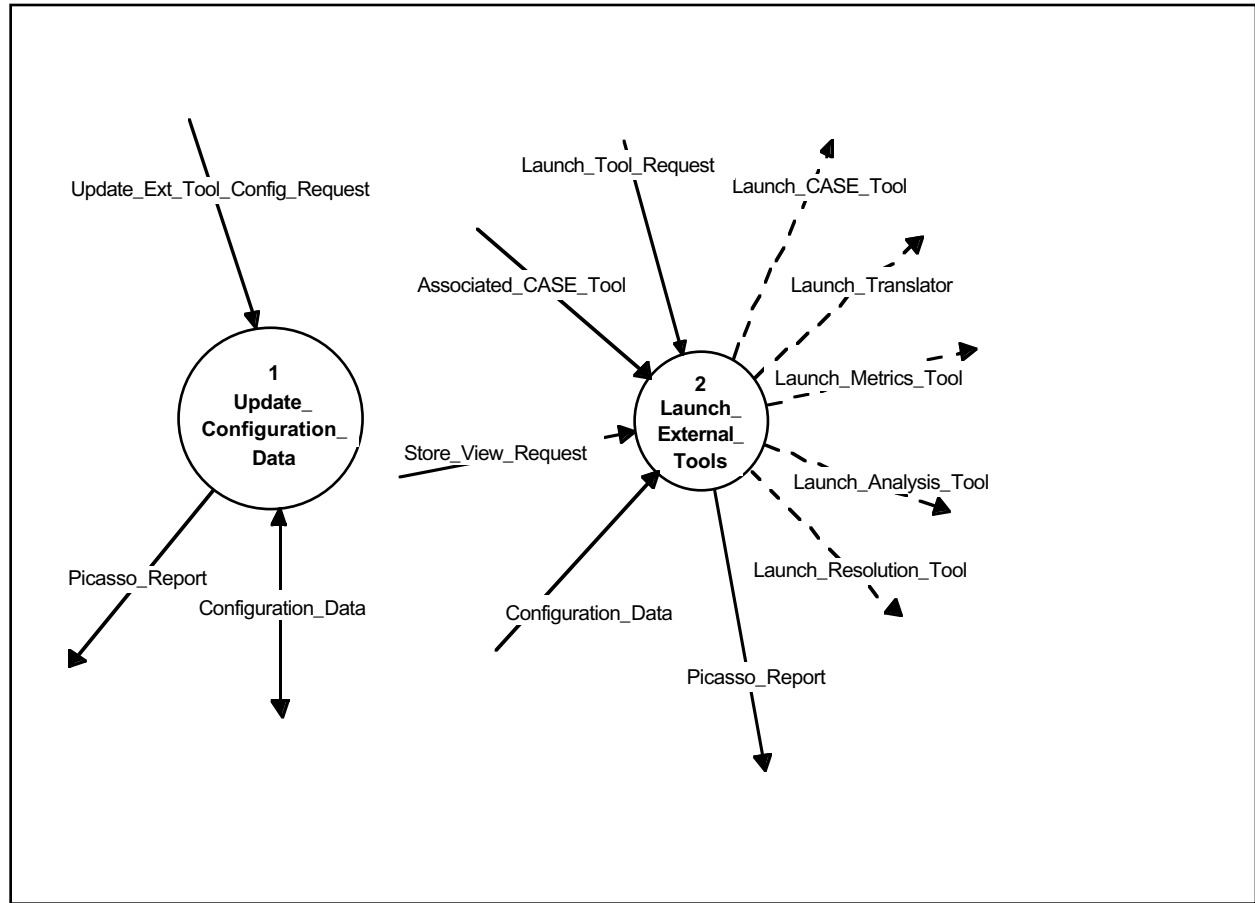


Figure 9 Manage_External_Tools Level 2 Data Flow Diagram

The Manage_External_Tools process will retrieve and store the appropriate Configuration_Data in the External_Tool_Configuration Store based on a user's request. A Picasso_Report will be returned to the user indicating the status or confirmation of the request.

For a Launch_Tool_Request, the process will use the Tool_Type to determine which external tool to launch. If the Tool_Type is a translator, then the process will obtain Configuration_Data from the External_Tool_Configuration data store to launch the input or output translator. If the Tool_Type is metrics, then the process will obtain Configuration_Data from the External_Tool_Configuration data store. This data will be used to determine which metrics tool to launch. If the Tool_Type is analysis, then the process will obtain Configuration_Data from the External_Tool_Configuration data store to determine which analysis tool to initiate. If the Tool_Type is resolution, then the process will obtain Configuration_Data from the External Configuration data store to determine which resolution tool to initiate. Finally, if the tool type is CASE tool, the process will obtain the configuration data from the External Configuration Store to determine which CASE tool to initiate.

For a Store_View_Request from the user, the process will retrieve Configuration_Data from the External_Tool_Configuration data store. The appropriate input translator will be launched based on the Configuration_Data.

For an Update External Configuration Request, the process will add or delete a tool information External_Tool_Configuration data store. When a Delete tool is requested, the process will remove the External_Tool_Configuration data from the External_Tool_Configuration data store. If an Add_Request is issued, then the process will store the new External_Tool Configuration data to the External_Tool_Configuration data store.

For an Associated_CASE_Tool request, the process will obtain the appropriate Configuration_Data from the External_Tool_Configuration data store. Once the data is obtained, the process will launch the appropriate CASE tool. A Picasso_Report will be generated for the user indicating the status of the request.

Inputs:

Update_Ext_Tool_Config_Request

This is a type of Picasso_User_Request by which the User may update the Data in the External_Tool_Configuration data store. The Update_External_Tool_Configuration_Request can be of two forms. It can be an Add_Request which allows the user to add information about a new tool to the configuration, or it can be a Delete_Request which removes information about a tool in the current configuration.

Store_View_Request

Store_View_Request is a type of Picasso_User_Request which prepares the system for the receipt of a CASE_Tool_View and corresponding CG_Object View. It contains the Name of the view and the Author of the view as well as a View_Location field that describes to Picasso where to find the view. The Associated_CASE_Tool is also sent to Picasso which it uses to determine from which translator to fetch the CG_Object View .

Launch_Tool_Request

Launch_Tool_Request is a type of Picasso_User_Request that specifies that the user wants to launch either an analysis tool, a metrics tool, a problem resolution tool, or an output translator. It includes all the information necessary to perform a lookup in the External_Tool_Configuration data store and launch the tool. This information is either the name of the tool or the name of the Associated_CASE_Tool.

Configuration_Data

Configuration_Data contains the name of the External Tool used, the type of the tool, the external Tool_Launch_Info, and the Associated_CASE_Tool information.

Associated_CASE_Tool

The Associated_CASE_Tool provides the Picasso software with the necessary information to determine which Translator to launch when storing or retrieving CASE_Tool_VIEWS and/or CG_Objects.

Processing:

FOR Launch_Tool_Request

CASE Tool_Type

WHEN Translator

Obtain Configuration_Data From External_Tool_Configuration Store

Initiate Launch_Translator From Manage_External_Tools Process

WHEN Metrics

Obtain Configuration_Data From External_Tool_Configuration Store

Initiate Launch_Metrics_Tool From Manage_External_Tools Process

WHEN Resolution

Obtain Configuration_Data From External_Tool_Configuration Store

Initiate Launch_Resolution_Tool From Manage_External_Tools Process

WHEN Analysis

Obtain Configuration_Data From External_Tool_Configuration Store

Initiate Launch_Analysis_Tool From Manage_External_Tools Process

Generate Picasso_Report To User

FOR Store_View_Request

Obtain Configuration_Data From External_Tool_Configuration Store

Initiate Launch_Translator From External_Tool_Configuration Store

Generate Picasso_Report To User

FOR Update_Ext_Tool_Config_Request

CASE Request

WHEN Add_Request

Send Configuration_Data To External_Tool_Configuration Store

WHEN Delete_Request

Remove Configuration_Data From External_Tool_Configuration Store

Generate Picasso_Report To User

FOR Associated_CASE_Tool

Obtain Configuration_Data From External_Tool_Configuration Store

Initiate Launch_CASE_Tool From Manage_External_Tools Process

Generate Picasso_Report To User

Outputs:

Picasso_Report

A Picasso_Report includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, warnings, beeps, etc.

Configuration_Data

Configuration_Data contains the name of the External Tool used, the type of the tool, the external Tool_Launch_Info, and the Associated_CASE_Tool information.

Launch_Translator

Launch_Translator is the control signal from Picasso used to launch the various external translator tools. The information on how to launch the translator tools is stored in the External_Tool_Configuration data store.

Launch_Resolution_Tool

Launch_Resolution_Tool is the control signal from Picasso used to launch the various external problem resolution tools. The information on how to launch the resolution tools is stored in the External_Tool_Configuration data store.

Launch_CASE_Tool

Launch_CASE_Tool is the control signal from Picasso used to launch the various CASE tools. The information on how to launch the CASE tools is stored in the External_Tool_Configuration data store. The CASE tool is used an external viewer for the CASE_Tool_Views.

Launch_Analysis_Tool

Launch_Analysis_Tool is the control signal from Picasso used to launch the various external analysis tools. The information on how to launch the analysis tools is stored in the External_Tool_Configuration data store.

Launch_Metrics_Tool

Launch_Metrics_Tool is the control signal from Picasso used to launch the various external metrics tools. The information on how to launch the metrics tools is stored in the External_Tool_Configuration data store.

3.2.1.1.1 Process Update_Configuration_Data

For an Update External_Tool_Configuration Request, the process will add or delete an external tool from the Configuration Data Store. Upon an Add_Request for a new tool, the process will store the new tool information to the External_Tool_Configuration data store. If the Tool_Name and location already exists, then a Picasso_Report will be generated to warn the user. In addition, Picasso will provide the user an option to overwrite or keep the existing tool configuration. In the case of an incomplete external tool configuration entry, Picasso will warn the user and provide an option the complete the entry or cancel.

If the Tool_Name exists for a Delete_Request, then the process will remove the External_Tool_Configuration data from the External_Tool_Configuration data store and warn the user that the original CASE tool view that corresponds to the tool can not longer be viewed. If the tool does not exist, the user will be warned that no such tool exists to be deleted.

Inputs:

Update_Ext_Tool_Config_Request

Update_Ext_Tool_Config_Request is a type of Picasso_User_Request by which the User may update the Data in the External_Tool_Configuration data store. The Update_External_Tool_Configuration_Request can be of two forms. It can be an Add_Request which allows the user to add information about new tool to the configuration, or it can be a Delete_Request which removes information about a tool in the current configuration.

Configuration_Data

Configuration_Data contains the name of the External Tool used, the type of the tool, the external Tool_Launch_Info, and the Associated_CASE_Tool information.

Processing:

Retrieve Update_Ext_Tool_Config_Request

Retrieve Configuration_Data From External_Tool_Cofiguration Store

IF Update_Ext_Tool_Config_Request is Add_Request

IF Tool_Name Exists

Remove Existing entry From External_Tool_Configuration Store

Add New entry user requested
ELSE IF Update_Ext_Tool_Config_Request is Delete_Request
 IF Tool_Name Exists
 Remove Matched entry
Generate Picasso_Report to User

This process is primitive and has no decomposition.

Outputs:

Configuration_Data

Configuration_Data contains the name of the External Tool used, the type of the tool, the external Tool_Launch_Info, and the Associated_CASE_Tool information.

Picasso_Report

A Picasso_Report includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, error messages, warnings, beeps, etc.

3.2.1.1.2 Process Launch_External_Tools

Upon an Launch_Tool_Request, the Tool_Type is used to determine which external tool is launched. When the Tool_Type is translator, the Configuration_Data is retrieved from the External_Tool_Configuration data store. The process will search for an Associated_CASE_Tool that matches the query data. When a match is found, the appropriate translator is launched. If no matches are found, the process will generate a Picasso_Report to the user describing the error status of the request. The user will then be prompted to correct the request, possibly adding the requested tool to the external tool configuration, or cancelling the request.

Upon an metrics Tool_Type, the process will retrieve Configuration_Data from the External_Tool_Configuration data store. The process will search for a Tool_Name that matches the query data. Once a match is found, the appropriate metrics tool is launched. If no matches are found, a Picasso_Report will be generated for the user describing the error status of the request. The user will then be prompted to correct the request, possibly adding the requested tool to the external tool configuration, or cancelling the request.

Upon an resolution Tool_Type, the process will retrieve Configuration_Data from the External_Tool_Configuration data store. The process will search for a Tool_Name that matches the query data. Once a match is found, the appropriate resolution tool is launched. If no matches are found, a Picasso_Report will be generated for the user describing the error status of the request. The user will then be prompted to correct the request, possibly adding the requested tool to the external tool configuration, or cancelling the request.

Upon an analysis Tool_Type, the process will retrieve Configuration_Data from the External_Tool_Configuration data store. The process will search for the Tool_Name that

matches the query data. Once a match is found, the appropriate analysis tool is launched. If no matches are found, a Picasso_Report will be generated for the user describing the error status of the request. The user will then be prompted to correct the request, possibly adding the requested tool to the external tool configuration, or cancelling the request.

Upon an Store_View_Request from the user, the process retrieves Configuration_Data from the External_Tool_Configuration data store. The process will search for the Associated_CASE_Tool that matches the query data. Once a match is found, the appropriate input translator is launched. If no matches are found, a Picasso_Report will be generated for the user describing the error status of the request. The user will then be prompted to correct the request, possibly adding the requested tool to the external tool configuration, or cancelling the request.

Upon an Associated_Case_Tool request from the Manage Requirements_View Data process, the Launch_External_Tools process retrieves Configuration_Data from the External_Tool_Configuration data store. The process will search for the Associated_CASE_Tool that matches the query data. Once a match is found, the appropriate CASE tool is launched. If no matches are found, a Picasso_Report will be generated for the user describing the error status of the request. The user will then be prompted to correct the request, possibly adding the requested tool to the external tool configuration, or cancelling the request.

Inputs:

Configuration_Data

Configuration_Data contains the name of the External Tool used, the type of the tool, the external Tool_Launch_Info, and the Associated_CASE_Tool information.

Associated_CASE_Tool

Associated_CASE_Tool provides the Picasso software with the necessary information to determine which Translator to launch when storing or retrieving CASE_Tool_VIEWS and/or CG_Objects.

Launch_Tool_Request

Launch_Tool_Request is a type of Picasso_User_Request that specifies that the user wants to launch either an analysis tool, a metrics tool, problem resolution tool, or an output translator. It includes all the information necessary to perform a lookup in the External_Tool_Configuration data store and launch the tool. This information is either the name of the tool or the name of the Associated_CASE_Tool.

Store_View_Request

Store_View_Request is a type of Picasso_User_Request which prepares the system for the receipt of a CASE_Tool_View and corresponding CG_Object View. It contains the Name of the view and the Author **Harry S. Delugach** of the view as well as a View_Location field that describes to Picasso where to find the view. The Associated_CASE_Tool is also sent to Picasso which it uses to determine from which translator to fetch the CG_Object View .

Processing:

IF Launch_Tool_Request is received

FOR Tool_Type

CASE Translator

Retrieve Configuration_Data From External_Tool_Configuration Store

Find Associated_CASE_Tool that match Query Data

Initiate Launch_Translator From Launch_External_Tools Process

CASE Metrics

Retrieve Configuration_Data From External_Tool_Configuration Store

Find Tool_Name that match Query Data

Initiate Launch_Metrics_Tool From Launch_External_Tools Process

CASE Resolution

Retrieve Configuration_Data From External_Tool_Configuration Store

Find Tool_Name that match Query Data

Initiate Launch_Resolution_Tool From Launch_External_Tools Process

CASE Analysis

Retrieve Configuration_Data From External_Tool_Configuration Store

Find Tool_Name that match Query Data

Initiate Launch_Analysis_Tool From Launch_External_Tools Process

Generate Picasso_Report To User

ELSE IF Store_View_Request is received

Retrieve Configuration_Data From External_Tool_Configuration Store

Find Associated_Case_Tool that match Query Data

Initiate Launch_Translator From Launch_External_Tools Process

Generate Picasso_Report To User

ELSE IF Associated_CASE_Tool is received from Manage_Requirments_Views

Retrieve Configuration_Data From External_Tool_Configuration Store

Find Associated_CASE_Tool that match Query Data

Initiate Launch_CASE_Tool From Launch_External_Tools Process

Generate Picasso_Report To User

This process is primitive and has no decomposition.

Outputs:

Picasso_Report

This includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, warnings, beeps, etc.

Launch_CASE_Tool

Launch_CASE_Tool is the control signal from Picasso used to launch the various CASE tools. The information on how to launch the CASE tools is stored in the External_Tool_Configuration data store. The CASE tool is used an external viewer for the CASE_Tool_Views.

Launch_Translator

Launch_Translator is the control signal from Picasso used to launch the various external translator tools. The information on how to launch the translator tools is stored in the External_Tool_Configuration data store.

Launch_Metrics_Tool

Launch_Metrics_Tool is the control signal from Picasso used to launch the various external metrics tools. The information on how to launch the metrics tools is stored in the External_Tool_Configuration data store.

Launch_Analysis_Tool

Launch_Analysis_Tool is the control signal from Picasso used to launch the various external analysis tools. The information on how to launch the analysis tools is stored in the External_Tool_Configuration data store.

Launch_Resolution_Tool

Launch_Resolution_Tool is the control signal from Picasso used to launch the various external problem resolution tools. The information on how to launch the resolution tools is stored in the External_Tool_Configuration data store.

3.2.1.2 Process Manage_Reqts_Views

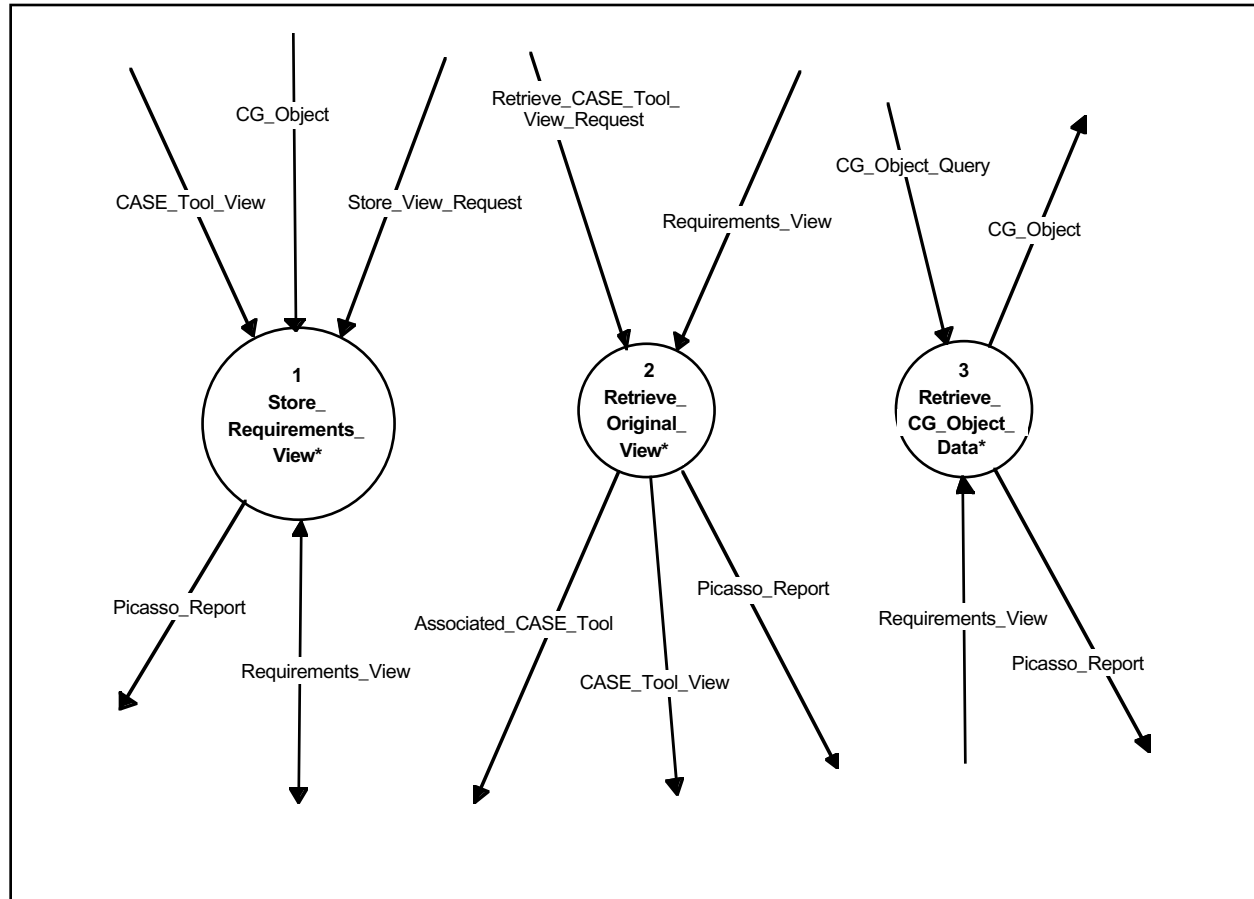


Figure 10 Manage_Reqts_Views Level 2 Data Flow Diagram

If a Store_View_Request is received, the Store_Requirements_View process will retrieve CASE_Tool_View from the CASE Tool and receive CG_Objects from a translator. The process then will store CASE_Tool_View and CG_Objects as a Requirements_View to the Requirements_View_Data store. Lastly, the process will generate a Picasso_Report back to the user indicating the success or failure of the request.

If a Retrieve_CASE_Tool_View_Request is received, the Retrieve_Original_View process will check the Requirements_View_Data store to match the request. If the view is found, then the process will retrieve Requirements_View from the data store. The process then will retrieve CASE_Tool_View and Associated_CASE_Tool from the retrieved requirement view. The process will send Associated_CASE_Tool to the process Manage_External_Tools to Launch_CASE_Tool. Once the CASE Tool is launched, the process will then send the CASE_Tool_View to the launched CASE Tool. Lastly the process will generate a Picasso_Report back to the user indicating the success or failure of the request.

If a CG_Object_Query is received, the Retrieve_CG_Object_Data process will check the Requirements_View_Data Store to match the request. If the view is found, then the process will

Retrieve_CG_Object and Requirement_View_Info from the retrieved requirement view. The process will then send CG_Objects and Requirement_View_Info to the requesting external tool.

Inputs:

Retrieve_CASE_Tool_View_Request

A Retrieve_CASE_Tool_View_Request is a Picasso_User_Request which allows the user to request a pre-existing CASE_Tool_View with a CASE_Tool_View_Query.

Store_View_Request

Store_View_Request is a type of Picasso_User_Request which prepares the system for the receipt of a CASE_Tool_View and corresponding CG_Object View. It contains the Name of the view and the Author of the view as well as a View_Location field that describes to Picasso where to find the view. The Associated_CASE_Tool is also sent to Picasso which uses it to determine which translator to fetch the CG_Object View.

CASE_Tool_View

The CASE_Tool_View is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

CG_Object

Conceptual Graph Objects (aka CG_Objects) are a knowledge representation of the various processes, data items, events, entities, and states in a Requirements_View. The external translators can either take a CASE_Tool_View and convert it into a collection of CG_Objects, or take a collection of CG_Objects and convert it into a CASE_Tool_View. CG_Objects fall into two categories: Link_Objects and Non_Link_Objects.

CG_Object_Query

The CG_Object_Request can Retrieve_CG_Objects based on the Requirements_View that they are a part of. They can be requested by specifying the Name, Author, Date/Time, or Version. Other conditions based on the characteristics of the individual CG_Objects can also be specified. These include the Type (ex. Concept, Context, Relation, etc.), Referent (used for Concepts), Label (used for Links), or Annotation (used for Demons and Actors).

Requirements_View

Requirements_View contains a CASE_Tool_View and one or more CG_Object (s).

Outputs:

CASE_Tool_View

The CASE_Tool_View is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then

be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

CG_Object

Conceptual Graph Objects(aka CG_Objects) are a knowledge representation of the various processes, data items, events, entities, and states in a Requirements_View. The external translators can either take a CASE_Tool_View and convert it into a collection of CG_Objects, or take a collection of CG_Objects and convert it into a CASE_Tool_View. CG_Objects fall into two categories: Link_Objects and Non_Link_Objects.

Requirements_View_Info

Requirements_View_Info contains general information about a Requirements_View. This includes the View_Name, User_Name of the author, Date_Time_Info, Version, and the Associated_CASE_Tool.

Requirements_View

Requirements_View contains a CASE_Tool_View and one or more CG_Object (s).

Associated_CASE_Tool

Associated_CASE_Tool provides the Picasso software with the necessary information to determine which Translator to launch when storing or retrieving CASE_Tool_VIEWS and/or CG_Objects.

Picasso_Report

A Picasso_Report includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, warnings, beeps, etc.

3.2.1.2.1 Process Store_Requirements_View

If a Store_View_Request is received, the process will retrieve Version information from the Requirements_View_Data data store, Date_Time_Info from the operating system, and the CASE_Tool_View from the CASE Tool. The process will also receive CG_Object(s) from a translator. Once all the input information are available, the process will store these inputs as a Requirements_View in the Requirements_View_Data store. Lastly the process will generate a Picasso_Report back to the user indicating the success or failure of the request.

Inputs:

CASE_Tool_View

The CASE_Tool_View is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

CG_Object

Conceptual Graph Objects (aka CG_Objects) are a knowledge representation of the various processes, data items, events, entities, and states in a Requirements_View. The external translators can either take a CASE_Tool_View and convert it into a collection of CG_Objects, or take a collection of CG_Objects and convert it into a CASE_Tool_View. CG_Objects fall into two categories: Link_Objects and Non_Link_Objects.

Store_View_Request

Store_View_Request is a type of Picasso_User_Request which prepares the system for the receipt of a CASE_Tool_View and corresponding CG_Object View. It contains the Name of the view and the Author of the view as well as a View_Location field that describes to Picasso where to find the view. The Associated_CASE_Tool is also sent to Picasso which uses it to determine which translator to fetch the CG_Object View.

Requirements_View

Requirements_View contains a CASE_Tool_View and one or more CG_Object (s).

This process is primitive and has no decomposition.

Outputs:

Requirements_View

Requirements_View contains a CASE_Tool_View and one or more CG_Object (s).

Picasso_Report

A Picasso_Report includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, warnings, beeps, etc.

Processing:

Retrieve Version information From Requirements_View_Data

Retrieve Date_Time_Info

Retrieve CASE_Tool_View From CASE_Tool

Retrieve CG_Object From Translator

IF CASE_Tool_View and CG_Object are received

Generate Requirements_View by Adding Time and Version information to the Original CASE_Tool_View and CG_Object

Store Requirements_View To Requirements_View_Data Store

Generate Picasso_Report To User

3.2.1.2.2 Process Retrieve_Original_View

If a Retrieve_CASE_Tool_View_Request is received, the process will check the Requirements_View_Data store to find the Requirements_View that matches the request. If the view is found, then the process will retrieve the Requirements_View from the data store. The process will then send Associated_CASE_Tool to the process Manage_External_Tools to Launch_CASE_Tool. Once the CASE Tool is launched, the process will send CASE_Tool_View to the launched CASE Tool. Lastly the process will generate a Picasso_Report back to the user indicating the success or failure of the request.

Inputs:

Retrieve_CASE_Tool_View_Request

A Retrieve_CASE_Tool_View_Request is a Picasso_User_Request which allows the user to request a pre-existing CASE_Tool_View with a CASE_Tool_View_Query.

Requirements_View

Requirements_View contains a CASE_Tool_View and one or more CG_Object (s).

This process is primitive and has no decomposition.

Outputs:

CASE_Tool_View

The CASE_Tool_View is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

Associated_CASE_Tool

Associated_CASE_Tool provides the Picasso software with the necessary information to determine which Translator to launch when storing or retrieving CASE_Tool_VIEWS and/or CG_Objects.

Picasso_Report

A Picasso_Report includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, warnings, beeps, etc.

Processing:

IF Requirements_View Found

Retrieve Requirements_View From Requirements_View_Data Store

Find CASE_Tool_View in Requirements_View that match the request

Send	Associated_CASE_Tool	To	Manage_External_Tool Process
Send	matched CASE_Tool_View	To	CASE_Tool
Generate	Picasso_Report	To	User

3.2.1.2.3 Process Retrieve_CG_Object_Data

If a CG_Object_Query is received, the process will check the Requirements_View_Data data store to find the Requirements_View(s) that matches the request. If a matching view is found, then the process will retrieve the Requirements_View(s) from the data store. The process will then send the CG_Object(s) and Requirement_View_Info to the requesting external tool. Lastly the process will generate a Picasso_Report back to the user indicating the success or failure of the request.

Inputs:

CG_Object_Query

The CG_Object_Request can Retrieve_CG_Objects based on the Requirements_View of which they are a part. They can be requested by specifying the Name, Author, Date/Time, or Version. Other conditions based on the characteristics of the individual CG_Objects can also be specified. These include the Type (ex. Concept, Context, Relation, etc.), Referent (used for Concepts), Label (used for Links), or Annotation (used for Demons and Actors).

Requirements_View

Requirements_View contains a CASE_Tool_View and one or more CG_Object (s).

This process is primitive and has no decomposition.

Outputs:

CG_Object

Conceptual Graph Objects (aka CG_Objects) are a knowledge representation of the various processes, data items, events, entities, and states in a Requirements_View. The external translators can either take a CASE_Tool_View and convert it into a collection of CG_Objects, or take a collection of CG_Objects and convert it into a CASE_Tool_View. CG_Objects fall into two categories: Link_Objects and Non_Link_Objects.

Picasso_Report

A Picasso_Report includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, warnings, beeps, etc.

Processing:

IF Requirements_View Found

Retrieve	Requirements_View	From	Requirements_View_Data Store
Send	CG_Objects in Requirements_View that match Query Data		
Send	Matched CG_Object	To	External Tool
Send	Requirements_View_Info	To	External Tool
Generate	Picasso_Report	To	User

3.2.1.3 Process Manage_Type_Hierarchy

If a Type_Hierarchy_Query is received, the process will search the Type_Hierarchy_Data store for the requested Type_Hierarchy_Data. If data is found then the Process will retrieve Type_Hierarchy_Data from the Type_Hierarchy_Data Store and send Type_Hierarchy_Data to the requesting external tool; translator, analysis tool, and resolution tool. In the case of the submission of an already existing data type, Picasso will generate an error message to the submitting resolution tool.

Inputs:

Type_Hierarchy_Data

Type_Hierarchy_Data includes a collection of Types and the Parent/Child Relationships between them.

Type_Hierarchy_Query

A Type_Hierarchy_Query is a mechanism which will be used by translators and analysis tools to retrieve type hierarchy information to process conceptual graphs. The basic query specifies a search based on conditions for Name, Date/Time, and the person who added the entry. Like other search conditions, these may be keyword, pattern matching, equality, inequality, or greater than or less than. A further extension to this query allows the request to fetch ancestors or descendants of all the entries retrieved by the basic query. The ancestor condition can be specified to fetch all ancestors, ancestors back to the nth generation, ancestors beyond the nth generation, ancestors at the nth generation. Likewise, for a descendent condition, all descendent, descendent forward to the nth generation, descendent beyond the nth generation, and descendent at the nth generation

Processing:

IF	Type_Hierarchy_Data Found		
Retrieve	Type_Hierarchy_Data	From	Type_Hierarchy_Data Store
Send	Type_Hierarchy_Data	To	External Tool

This process is primitive and has no decomposition.

Outputs:

Type_Hierarchy_Data

Type_Hierarchy_Data includes a collection of Types and the Parent/Child Relationships between them.

3.2.1.4 Process Manage_Metrics_Log

Upon a Picasso_User_Request, the process will create a Command Log Entry to be stored in the Metrics_Log Store. The Command Log Entry will include such information as the User Command and the corresponding command result(s).

The process will also store Metrics_Log_Entry for the analysis tool, resolution tool, and user. Upon a Metrics_Data_Request, the process will perform a query on the Metrics_Log store and return the Metrics_Log_Entry found to the appropriate tool or user. If an analysis query is requested from the analysis tool, then such information as Problem_Description and Problem_Type will be returned. If a Resolution Query is requested, such information as the action take will be returned to the Resolution Tool. Upon a User_Command_Query Metrics_Data_Request, such information as user command and results will be returned to the user.

Upon an Analysis_Log_Entry, the process will store the analysis entry as a Metrics_Log_Entry to the Metrics_Log store. Likewise, upon a Resolution_Log_Entry, the process will store the resolution entry as a Metrics_Log_Entry to the Metrics_Log store.

In the case of the submission of an ill-formed log entries, Picasso will generate an error message to the submitter, whether it is the Picasso user or an external tool.

Inputs:

Picasso_User_Request

A Picasso_User_Request is data transmitted directly from the user to the Picasso software. It includes the user's location and one of the following: Launch_Tool_Request, Store_View_Request, Retrieve CASE_Tool_View Request, or Update External_Tool_Configuration data Request.

Resolution_Log_Entry

Resolution_Log_Entries contain information about corrective actions taken by the problem Resolution Tools which will be stored in the Metrics_Log. It includes the problem resolution Tool_Name, and Action_Taken.

Analysis_Log_Entry

An Analysis_Log_Entry contains the results of analysis that are to be stored in the Metrics_Log. This entry includes the User_Name, Analysis Tool_Name, Problem_Type, and a textual Problem_Description.

Metrics_Data_Request

The Metrics_Data_Request is used by the external metrics tools to request data from the Metrics_Log. A Metrics_Data_Request may include the three following types of requests: Analysis_Data_Query, Resolution_Data_Query, or User_Command_Query. In addition, these requests may be combined together to allow the user to perform an Analysis_Data_Query and use its results in a Resolution_Data_Query.

Metrics_Log_Entry

Metrics_Log_Entry is used as raw data for the metrics that the external metrics tool calculates. These metrics will be used to help describe how the system is being used, determine how well the current configuration of the system is performing, and measure the development process of a specific project. A Metrics_Log_Entry can be one of three different types: Analysis_Log_Entry, Resolution_Log_Entry, or User_Command_Entry. Every Metrics_Log_Entry has Date/Time_Info.

Processing:

```
IF      Picasso_User_Request is received
    Send  Metrics_Log_Entry  To    Metrics_Log Store
ELSE IF Metrics_Data_Request is received
    Retrieve Metrics_Log_Entry From Metrics_Log Store
    Send    Metrics_Log_Entry
ELSE IF Analysis_Log_Entry is received
    Stores Metrics_Log_Entry To Metrics_Log Store
ELSE IF Resolution_Log_Entry
    Stores Metrics_Log_Entry To Metrics_Log Store
```

This process is primitive and has no decomposition.

Outputs:

Metrics_Log_Entry

Metrics_Log_Entry is used as raw data for the metrics that the external metrics tool calculates. These metrics will be used to help describe how the system is being used, determine how well the current configuration of the system is performing, and measure the development process of a specific project. A Metrics_Log_Entry can be one of three different types: Analysis_Log_Entry, Resolution_Log_Entry, or User_Command_Entry. Every Metrics_Log_Entry has Date/Time_Info.

3.3 Internal Interface Description

Picasso will be a unique piece of software in that there will be virtually no communication between the different functional areas. Each process will have a very specific function and there

will be very little overlap between them. Only two of the four processes will communicate at all, and that will be a one way communication. When retrieving an original CASE_Tool_View, the appropriate CASE tool must be launched. The Retrieve_Original_View process (contained in the Manage_Reqs_Views process) must send the data item Associated_CASE_Tool to the Launch_External_Tools (contained in the Manage_External_Tools process.)

Data Items:

Associated_CASE_Tool

Description:

Associated_CASE_Tool provides the Picasso software with the necessary information to determine which Translator to launch when storing or retrieving CASE_Tool_Views and/or CG_Objects.

Structure:

Data item is primitive

Data item is Sent by:

- Process Manage_Reqs_Views/Retrieve_Original_View

Data item is Received by:

- Process Manage_External_Tools/Launch_External_Tools

3.4 CSCI Data Element Requirements

3.4.1 Data Dictionary List

3.4.1.1 Data Item Action_Taken

Description:

This is a textual description of the corrective action taken by the user.

Data item is referenced in:

- Data Item Resolution_Data_Query
- Data Item Resolution_Log_Entry

Structure:

Data item is primitive

3.4.1.2 Data Item Actor

Description:

An Actor is a type of Non_Link_Object which is used to represent a processing function. Objects that are connected in the direction toward the Actor are inputs. Objects that are connected in the direction away from the Actor are outputs. Actors often include a description of the algorithm used to transform the inputs into the outputs. This description is called an Annotation.

Data item is referenced in:

- Data Item Non_Link_Object

Structure:

CG_Object_Name +Type_Name+ (Annotation)

3.4.1.3 Data Item Add_Request

Description:

An Add_Request is a type of Update External_Tool_Configuration data Request which allows the User to add a new tool into the External_Tool_Configuration data. All Add_Requests will include the Tool_Type and the Tool_Name. If the tool is a Translator, the Add_Request will include the name of the Associated_CASE_Tool for which the translator is written.

Data item is referenced in:

- Data Item Update_Ext_Tool_Config_Request

Structure:

User_Location + User_Name + Tool_Type + Tool_Name + Tool_Launch_Info +
(Associated_CASE_Tool)

3.4.1.4 Data Item Analysis_Data_Query

Description:

The Analysis_Data_Query is used to request metrics entry logs which were created by the analysis tools. The metrics tools may request entries by Entry_Name, Date_Time_Info, User_Name, Tool_Name, Problem_Type and Problem_Description. In addition, any combination of these keys is possible.

Data item is referenced in:

- Data Item Metrics_Data_Request

Structure:

(Entry_Name) + (Date_Time_Info) + (User_Name) + (Tool_Name) + (Problem_Type) + (Problem_Description)

3.4.1.5 Data Item Analysis_Log_Entry

Description:

An Analysis_Log_Entry contains the results of analysis to be stored in the Metrics_Log. It includes the User_Name, Analysis Tool_Name, Problem_Type, and a textual Problem_Description.

Data item is referenced in:

- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Manage_Metrics_Log
- P-spec Picasso
- Data Item Metrics_Log_Entry

Structure:

User_Name + Tool_Name + Problem_Type + Problem_Description

3.4.1.6 Data Item Analysis_Report

Description:

This includes all data transmitted from the analysis tool directly to the user. The content of this data item will vary depending on the which analysis tool is being used. This may include screen displays, error messages, notifications, warnings, beeps, etc.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.7 Data Item Analysis_Request**Description:**

This includes all data transmitted from the user directly to the analysis tool. The content of this data item will vary depending on which analysis tool is being used.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.8 Data Item Ancestor_Descendent_Query**Description:**

This is a mechanism by which the Translators, Analysis Tools, and Resolution Tools may request Type_Hierarchy_Data based on Parent/Child Relationships. The query may include the Type_Name, Date_Time_Info, User_Name of the person who added it, Direction, and the number of Levels.

Data item is referenced in:

- Data Item Type_Hierarchy_Query

Structure:

(Type_Name) + (Date_Time_Info) + (User_Name) + (Direction) + (Levels)

3.4.1.9 Data Item Annotation**Description:**

A textual description of the algorithm used to transform the inputs of an Actor into the outputs of the Actor.

Data item is referenced in:

- Data Item Actor
- Data Item CG_Object_Query
- Data Item Demon

Structure:

Data item is primitive

3.4.1.10 Data Item Associated_CASE_Tool

Description:

This provides the Picasso software with the necessary information to determine which Translator to launch when storing or retrieving CASE_Tool_Views and/or CG_Objects.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Manage_Reqs_Views
- Flow Diagram Picasso
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Manage_Reqs_Views
- Data Item Add_Request
- Data Item CASE_Tool_View_Query
- Data Item Configuration_Data
- Data Item Launch_Tool_Request
- Data Item Requirements_View_Info
- Data Item Requirements_View_Info_Query
- Data Item Store_View_Request

Structure:

Data item is primitive

3.4.1.11 Data Item Author

Description:

The User_Name of the person who originally stored the CASE_Tool_View.

Data item is referenced in:

- Data Item CASE_Tool_View_Query
- Data Item Requirements_View_Info
- Data Item Requirements_View_Info_Query

Structure:

ALIAS User_Name

3.4.1.12 Data Item CASE_Tool_Report

Description:

This includes any data transmitted directly from the CASE tool to the user. The content will vary depending on which CASE tool is being used. This may include screen displays, error messages, notifications, warnings, beeps, etc.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.13 Data Item CASE_Tool_Request

Description:

This includes any data transmitted from the user directly to the CASE tool. The content will vary depending on which CASE tool is being used.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.14 Data Item CASE_Tool_View

Description:

The CASE_Tool_View is the original data from the CASE tool. This data is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then be retrieved by any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

Data item is referenced in:

- Flow Diagram Manage_Reqs_Views
- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Manage_Reqs_Views
- P-spec Picasso
- Data Item Requirements_View

Structure:

Data item is primitive

3.4.1.15 Data Item CASE_Tool_View_Query

Description:

The CASE_Tool_View_Request recovers all CASE_Tool_Views from Picasso by specifying conditions for the fields, Name, Author, Date/Time, and Version. All CASE_Tool_Views that are returned to the requester will satisfy the conditions specified in the request.

Data item is referenced in:

- Data Item Retrieve_CASE_Tool_View_Request

Structure:

(View_Name) + (Date_Time_Info) + (Author) + (Version) + (Associated_CASE_Tool)

3.4.1.16 Data Item CG_Object

Description:

Conceptual Graph Objects (aka CG_Objects) are a knowledge representation of the various processes, data items, events, entities, and states in a Requirements_View. The external translators can either take a CASE_Tool_View and convert it into a collection of CG_Objects, or take a collection of CG_Objects and convert it into a CASE_Tool_View. CG_Objects fall into two categories: Link_Objects and Non_Link_Objects.

Data item is referenced in:

- Flow Diagram Manage_Reqs_Views
- Flow Diagram Picasso
- P-spec Manage_Reqs_Views
- P-spec Picasso
- Data Item Picasso_Data
- Data Item Requirements_View

Structure:

[Link_Object | Non_Link_Object]

3.4.1.17 Data Item CG_Object_Name

Description:

A CG_Object Name is a unique identifier associated with each CG_Object.

Data item is referenced in:

- Data Item Actor
- Data Item CG_Object_Query
- Data Item Concept

- Data Item Context
- Data Item Dashed_Directed_Arc
- Data Item Demon
- Data Item Directed_Arc
- Data Item Line_Of_Identity
- Data Item Relation

Structure:

Data item is primitive

3.4.1.18 Data Item CG_Object_Query

Description:

The CG_Object_Request can Retrieve_CG_Objects based on the Requirements_View that they are a part of. They can be requested by specifying the Name, Author, Date/Time, or Version. Other conditions based on the characteristics of the individual CG_Objects can also be specified. These include the Type (ex. Concept, Context, Relation, etc.), Referent (used for Concepts), Label (used for Links), or Annotation (used for Demons and Actors).

Data item is referenced in:

- Flow Diagram Manage_Reqs_Views
- Flow Diagram Picasso
- P-spec Manage_Reqs_Views
- Data Item Picasso_Data_Request

Structure:

(CG_Object_Name) + (Type_Name) + (Referent) + (Label) + (Annotation)

3.4.1.19 Data Item Command_Name

Description:

This is a unique identifier that identifies the type of command that was issued by the user.

Data item is referenced in:

- Data Item User_Command_Entry
- Data Item User_Command_Query

Structure:

Data item is primitive

3.4.1.20 Data Item Concept

Description:

A Concept is a Non_Link_Object used to represent an idea or thought. A Concept may include a unique identifier, called a referent, to distinguish it from other concepts of the same type.

Data item is referenced in:

- Data Item Non_Link_Object

Structure:

CG_Object_Name + Type_Name + (Referent)

3.4.1.21 Data Item Configuration_Data

Description:

Configuration_Data contains the name of the External Tool used, the type of the tool, the external Tool_Launch_Info, and the Associated_CASE_Tool information.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Picasso
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Update_Configuration_Data
- Data Item External_Tool_Configuration

Structure:

Tool_Type + Tool_Name + Tool_Launch_Info + Associated_CASE_Tool

3.4.1.22 Data Item Confirmation_Report

Description:

The Confirmation_Message notifies the user with confirmed message.

Data item is referenced in:

- Data Item Picasso_Report

Structure:

Data item is primitive

3.4.1.23 Data Item Context

Description:

A Context is a type of Non_Link_Object used to group objects that belong together. This may be in respect to time, subsystem, or textual context. It may contain other CG_Objects, including other contexts.

Data item is referenced in:

- Data Item Non_Link_Object

Structure:

CG_Object_Name + Type_Name

3.4.1.24 Data Item Dashed_Directed_Arc

Description:

A Dashed_Directed_Arc is used to connect either a Concept or Context with either an Actor or Demon that is in the same context. For an Actor, the direction of the arrow shows the which concepts/contexts are inputs to the actor and which are outputs of the actor. For a demon, the direction of the relation show the time sequence in which concepts/contexts come into existence and which cease to exist.

Data item is referenced in:

- Data Item Link_Object

Structure:

CG_Object_Name + (Label)

3.4.1.25 Data Item Date_Time_Info

Description:

This contains date and time information. It is typically used as a time stamp for logging various requests to the software.

Data item is referenced in:

- Data Item Analysis_Data_Query
- Data Item Ancestor_Descendent_Query
- Data Item CASE_Tool_View_Query
- Data Item Metrics_Log_Entry
- Data Item Requirements_View_Info
- Data Item Requirements_View_Info_Query
- Data Item Resolution_Data_Query

- Data Item Type
- Data Item Type_Query
- Data Item User_Command_Query

Structure:

Data item is primitive

3.4.1.26 Data Item Delete_Request

Description:

A Delete_Request is a type of Update_External_Tool_Configuration data Request which allows the User to delete tools from the External_Tool_Configuration data Store. The Delete_Request takes the Tool_Name and causes the entry in the External_Tool_Configuration data store with that Tool_Name to be deleted.

Data item is referenced in:

- Data Item Update_Ext_Tool_Config_Request

Structure:

User_Location + User_Name + Tool_Name

3.4.1.27 Data Item Demon

Description:

A Demon is a type of Non_Link_Object that is used to represent time sequences. Objects that are linked towards the Demon are said to go out of existence. Objects that are linked away from the Demon are said to come into existence.

Data item is referenced in:

- Data Item Non_Link_Object

Structure:

CG_Object_Name + Type_Name + (Annotation)

3.4.1.28 Data Item Directed_Arc

Description:

A Directed_Arc is used to connect a Concept with a Relation that is in the same context. The direction of the arrow is used to indicate the direction of the relation.

Data item is referenced in:

- Data Item Link_Object

Structure:

CG_Object_Name + (Label)

3.4.1.29 Data Item Direction**Description:**

This data indicates whether tool is requesting the ancestors, descendants, or both.

Data item is referenced in:

- Data Item Ancestor_Descendent_Query

Structure:

Data item is primitive

3.4.1.30 Data Item Entry_Name**Description:**

The Entry_Name is a unique identifier associated with each entry in the Metrics_Log

Data item is referenced in:

- Data Item Analysis_Data_Query
- Data Item Resolution_Data_Query

Structure:

Data item is primitive

3.4.1.31 Data Item Error_Report**Description:**

The Error_Report notifies the user with error type.

Data item is referenced in:

- Data Item Picasso_Report

Structure:

Data item is primitive

3.4.1.32 Data Item External_Tool_Configuration**Description:**

The External_Tool_Configuration Store contains Configuration_Data.

Data item is referenced in:

- Flow Diagram Picasso

Structure:

{Configuration_Data}

3.4.1.33 Data Item Label**Description:**

A Label is an optional word that provides additional information about the Link_Object.

Data item is referenced in:

- Data Item CG_Object_Query
- Data Item Dashed_Directed_Arc
- Data Item Directed_Arc
- Data Item Line_Of_Identity

Structure:

Data item is primitive

3.4.1.34 Data Item Launch_Analysis_Tool**Description:**

This is the control signal from Picasso used to launch the various external analysis tools. The information on how to launch the analysis tools is stored in the External_Tool_Configuration data store.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Picasso

Structure:

Data item is primitive

3.4.1.35 Data Item Launch_CASE_Tool**Description:**

This is the control signal from Picasso used to launch the various CASE tools. The information on how to launch the CASE tools is stored in the External_Tool_Configuration data store. The CASE tool is used an external viewer for the CASE_Tool_Views.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Picasso

Structure:

Data item is primitive

3.4.1.36 Data Item Launch_Input_Translator**Description:**

This is the control signal from Picasso used to launch the input translator tools.

Data item is referenced in:

- Data Item Launch_Translator

Structure:

Data item is primitive

3.4.1.37 Data Item Launch_Metrics_Tool**Description:**

This is the control signal from Picasso used to launch the various external metrics tools. The information on how to launch the metrics tools is stored in the External_Tool_Configuration data store.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Picasso

Structure:

Data item is primitive

3.4.1.38 Data Item Launch_Output_Translator

Description:

This is the control signal from Picasso used to launch the output translator tools.

Data item is referenced in:

- Data Item Launch_Translator

Structure:

Data item is primitive

3.4.1.39 Data Item Launch_Resolution_Tool

Description:

This is the control signal from Picasso used to launch the various external problem resolution tools. The information on how to launch the resolution tools is stored in the External_Tool_Configuration data store.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Picasso

Structure:

Data item is primitive

3.4.1.40 Data Item Launch_Tool_Request

Description:

This is a type of Picasso_User_Request that specifies that the user wants to launch either an analysis tool, a metrics tool, problem resolution tool, or an output translator. It includes all the information necessary to perform a lookup in the External_Tool_Configuration data store and launch the tool. This information is either the name of the tool or the name of the Associated_CASE_Tool.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Picasso

- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Picasso
- Data Item Picasso_User_Request

Structure:

User_Location + User_Name + Tool_Type + [Tool_Name | Associated_CASE_Tool]

3.4.1.41 Data Item Launch_Translator

Description:

This is the control signal from Picasso used to launch the various external translator tools. The information on how to launch the translator tools is stored in the External_Tool_Configuration data store.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Picasso

Structure:

[Launch_Input_Translator | Launch_Output_Translator]

3.4.1.42 Data Item Levels

Description:

This specifies how many levels up or down the query should extend from the beginning type.

Data item is referenced in:

- Data Item Ancestor_Descendent_Query

Structure:

Data item is primitive

3.4.1.43 Data Item Line_Of_Identity

Description:

A Line_of_Identity is a type of Link_Object that is used to connect two Concepts. It shows that the two objects are really the same object.

Data item is referenced in:

- Data Item Link_Object

Structure:

CG_Object_Name + (Label)

3.4.1.44 Data Item Link_Object

Description:

A Link_Object is a type of Conceptual Graph Object used to connect two Non_Link_Objects Together. There are three types of Link_Objects: Directed_Arcs, Dashed_Directed_Arcs, and Lines of Identity. A Link_Object may have an optional Label.

Data item is referenced in:

- Data Item CG_Object

Structure:

[Directed_Arc|

Dashed_Directed_Arc|

Line_Of_Identity]

3.4.1.45 Data Item Metrics_Data_Request

Description:

The Metrics_Data_Request is used by the external metrics tools to request data from the Metrics_Log. A Metrics_Data_Request may include the three following types of requests: Analysis_Data_Query, Resolution_Data_Query, or User_Command_Query. In addition, these requests may be combined together to allow the user to perform an Analysis_Data_Query and use its results in a Resolution_Data_Query.

Data item is referenced in:

- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Manage_Metrics_Log
- P-spec Picasso

Structure:

(Analysis_Data_Query) + (Resolution_Data_Query) + (User_Command_Query)

3.4.1.46 Data Item Metrics_Log

Description:

Metrics_Log Store contains Metrics_Log_Entry (s)

Data item is referenced in:

- Flow Diagram Picasso

Structure:

{Metrics_Log_Entry}

3.4.1.47 Data Item Metrics_Log_Entry

Description:

Metrics_Log_Entry is used as raw data for the metrics that the external metrics tool calculates. These metrics will be used to help describe how the system is being used, determine how well the current configuration of the system is performing, and measure the development process of a specific project. A Metrics_Log_Entry can be one of three different types: Analysis_Log_Entry, Resolution_Log_Entry, or User_Command_Entry. Every Metrics_Log_Entry has Date/Time_Info.

Data item is referenced in:

- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Manage_Metrics_Log
- P-spec Picasso
- Data Item Metrics_Log

Structure:

Date_Time_Info + [Analysis_Log_Entry | Resolution_Log_Entry | User_Command_Entry]

3.4.1.48 Data Item Metrics_Report

Description:

This includes all data transmitted from the various metrics tools directly to the user. The content will vary depending on which metrics tool is being used. This may include screen displays, error messages, notifications, warnings, beeps, etc.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.49 Data Item Metrics_Request

Description:

This includes all data transmitted from the user directly to the metrics tools. The content will vary depending on which metrics tool being used.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.50 Data Item Negation

Description:

This is a Boolean value that indicates whether or not the Context or Relation should be negated.

Data item is referenced in:

- Data Item Relation

Structure:

Data item is primitive

3.4.1.51 Data Item Non_Link_Object

Description:

A Non_Link_Object can be any one of the following types: Context, Concept, Relation, Actor, or Demon. They are used to represent contexts, ideas, relationships, processing functions, or time sequences. Every Non_Link_Object has a Type. They are connected to each other by Link_Objects.

Data item is referenced in:

- Data Item CG_Object

Structure:

[Context | Concept | Relation | Actor | Demon]

3.4.1.52 Data Item Parent_Child_Relation

Description:

This contains information about the relationship between two types in the Type_Hierarchy_Data Store. The Parent/Child Relationship indicates a Generalization/Specialization kind of connection between the Types.

Data item is referenced in:

- Data Item Type_Hierarchy_Data

Structure:

Data item is primitive

3.4.1.53 Data Item Picasso_Data Description:

Picasso_Data includes any data the software provides in response to Picasso_Data_Requests from the translators, analysis tools, or resolution tools. This may include conceptual graph objects, Type_Hierarchy_Data, and Requirements_View_Info.

Data item is referenced in:

- Flow Diagram Requirements_Assistant
- P-spec Picasso

Structure:

(CG_Object) + (Type_Hierarchy_Data) + (Requirements_View_Info)

3.4.1.54 Data Item Picasso_Data_Request

Description:

A Picasso_Data_Request is a mechanism for the translators, analysis tools, and resolution tools to request CG_Objects, Type_Hierarchy_Data, and/or Requirements_View_Info from the Picasso software. The request may include any combination of these types of requests, which will result in a Join operation on the results.

Data item is referenced in:

- Flow Diagram Requirements_Assistant
- P-spec Picasso

Structure:

(CG_Object_Query) + (Type_Hierarchy_Query) + (Requirements_View_Info_Query)

3.4.1.55 Data Item Picasso_Report

Description:

This includes all the data which the Picasso software transmits directly to the user. This may include screen displays, Error_Reports, Confirmation_Reports, notifications, warnings, beeps, etc.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Manage_Reqs_Views

- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Manage_Reqs_Views
- P-spec Picasso
- P-spec Update_Configuration_Data

Structure:

[Confirmation_Report|Error_Report]

3.4.1.56 Data Item Picasso_User_Request

Description:

A Picasso_User_Request is data transmitted directly from the user to the Picasso software. It includes the user's location and one of the following: Launch_Tool_Request, Store_View_Request, Retrieve CASE_Tool_View Request, or Update External_Tool_Configuration data Request.

Data item is referenced in:

- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Manage_Metrics_Log
- P-spec Picasso

Structure:

[Launch_Tool_Request | Store_View_Request | Retrieve_CASE_Tool_View_Request | Update_Ext_Tool_Config_Request]

3.4.1.57 Data Item Problem_Description

Description:

This is a textual description of the problem which was found by the analysis tool.

Data item is referenced in:

- Data Item Analysis_Data_Query
- Data Item Analysis_Log_Entry

Structure:

Data item is primitive

3.4.1.58 Data Item Problem_Type

Description:

This is an identifier which is used to determine the category of the problem. This may be as simple as conflict, incompleteness, or ambiguity. It could include more information such as importance and difficulty.

Data item is referenced in:

- Data Item Analysis_Data_Query
- Data Item Analysis_Log_Entry

Structure:

Data item is primitive

3.4.1.59 Data Item Referent

Description:

A referent is a unique identifier which is used to distinguish one concept from other concepts of the same type.

Data item is referenced in:

- Data Item CG_Object_Query
- Data Item Concept

Structure:

Data item is primitive

3.4.1.60 Data Item Relation

Description:

A Relation is a type of Non_Link_Object that is used to describe the relationship between two concepts. The direction of the arrows on the connection indicate the direction of the relationship. A Relation may also include an optional negation.

Data item is referenced in:

- Data Item Non_Link_Object

Structure:

CG_Object_Name + Type_Name + (Negation)

3.4.1.61 Data Item Requirements_View

Description:

Requirements_View contains a CASE_Tool_View and one or more CG_Object (s).

Data item is referenced in:

- Flow Diagram Manage_Reqs_Views
- Flow Diagram Picasso
- P-spec Manage_Reqs_Views
- P-spec Picasso
- Data Item Requirements_View_Data

Structure:

CASE_Tool_View+

Requirements_View_Info+

{CG_Object}

3.4.1.62 Data Item Requirements_View_Data**Description:**

Requirements_View_Data Store contains Requirements_View(s).

Data item is referenced in:

- Flow Diagram Picasso

Structure:

{Requirements_View}

3.4.1.63 Data Item Requirements_View_Info**Description:**

Requirements_View_Info contains general information about a Requirements_View. This includes the View_Name, User_Name of the author, Date_Time_Info, Version, and the Associated_CASE_Tool.

Data item is referenced in:

- Flow Diagram Picasso
- P-spec Manage_Reqs_Views
- Data Item Picasso_Data
- Data Item Requirements_View

Structure:

View_Name+

Author+

Date_Time_Info+

Version+

Associated_CASE_Tool

3.4.1.64 Data Item Requirements_View_Info_Query

Description:

The Requirements_View_Info_Query is a mechanism for translators, analysis tools, and resolution tools to request Requirements_View_Info from the Requirements_View Data Store. The tools may request objects by View_Name, Author, Date_Time_Info, Version, and Associated_CASE_Tool. In addition, any combination of the above will be accepted.

Data item is referenced in:

- Data Item Picasso_Data_Request

Structure:

(View_Name) + (Author) + (Date_Time_Info) + (Version) + (Associated_CASE_Tool)

3.4.1.65 Data Item Resolution_Data_Query

Description:

The Resolution_Data_Query is used to request metrics entry logs which were created by the problem resolution tools. The metrics tools may request entries by Entry_Name, Date_Time_Info, User_Name, Tool_Name, and Action_Taken. In addition, any combination of these keys is possible.

Data item is referenced in:

- Data Item Metrics_Data_Request

Structure:

(Entry_Name) + (Date_Time_Info) + (User_Name) + (Tool_Name) + (Action_Taken)

3.4.1.66 Data Item Resolution_Log_Entry

Description:

Resolution_Log_Entries contain information about corrective actions taken by the problem Resolution Tools which will be stored in the Metrics_Log. It includes the problem resolution Tool_Name, and Action_Taken.

Data item is referenced in:

- Flow Diagram Picasso
- Flow Diagram Requirements_Assistant
- P-spec Manage_Metrics_Log
- P-spec Picasso

- Data Item Metrics_Log_Entry

Structure:

User_Name + Tool_Name + Action_Taken

3.4.1.67 Data Item Resolution_Report

Description:

This includes all data transmitted from the Resolution Tools directly to the User. This may include screen displays, error messages, notifications, warnings, beeps, etc. The content of this data item will vary depending on which Resolution Tool is being used.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.68 Data Item Resolution_Request

Description:

This includes all the data transmitted directly from the User to the Resolution Tools. The content of this data item will vary depending on which Resolution Tool is being used.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.69 Data Item Retrieve_CASE_Tool_View_Request

Description:

A Retrieve_CASE_Tool_View_Request is a Picasso_User_Request which allows the user to request a pre-existing CASE_Tool_View with a CASE_Tool_View_Query.

Data item is referenced in:

- Flow Diagram Manage_Reqs_Views
- Flow Diagram Picasso
- P-spec Manage_Reqs_Views
- P-spec Picasso
- Data Item Picasso_User_Request

Structure:

User_Location + User_Name + CASE_Tool_View_Query

3.4.1.70 Data Item Store_View_Request**Description:**

Store_View_Request is a type of Picasso_User_Request which prepares the system for the receipt of a CASE_Tool_View and corresponding CG_Object View. It contains the Name of the view and the Author of the view as well as a View_Location field that describes to Picasso where to find the view. The Associated_CASE_Tool is also sent to Picasso which it uses to determine from which translator to fetch the CG_Object View .

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Manage_Reqs_Views
- Flow Diagram Picasso
- P-spec Launch_External_Tools
- P-spec Manage_External_Tools
- P-spec Manage_Reqs_Views
- P-spec Picasso
- Data Item Picasso_User_Request

Structure:

User_Location + User_Name + View_Name + View_Location + Associated_CASE_Tool

3.4.1.71 Data Item Tool_Launch_Info**Description:**

This includes all the information necessary to launch the external tool. This would most likely be a command line which is executed to launch the tool.

Data item is referenced in:

- Data Item Add_Request
- Data Item Configuration_Data

Structure:

Data item is primitive

3.4.1.72 Data Item Tool_Name

Description:

This is the name of an external tool which may be a CASE Tool or some other tool used for Translation, Metrics_Reporting, Problem Analysis, or Problem Resolution

Data item is referenced in:

- Data Item Add_Request
- Data Item Analysis_Data_Query
- Data Item Analysis_Log_Entry
- Data Item Configuration_Data
- Data Item Delete_Request
- Data Item Launch_Tool_Request
- Data Item Resolution_Data_Query
- Data Item Resolution_Log_Entry

Structure:

Data item is primitive

3.4.1.73 Data Item Tool_Type

Description:

Specifies whether the tool is an analysis tool, resolution tool, metrics tool, or translator.

Data item is referenced in:

- P-spec Picasso
- Data Item Add_Request
- Data Item Configuration_Data
- Data Item Launch_Tool_Request

Structure:

Data item is primitive

3.4.1.74 Data Item Translator_Report

Description:

This includes all information transmitted from the Translator directly to the User. The content of this data item will vary depending on which Translator is being used. This may include screen displays, error messages, notifications, warnings, beeps, etc.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.75 Data Item Translator_Request

Description:

This includes all information transmitted directly from the User to the Translators. The content of this data item will vary depending on which Translator is being used.

Data item is referenced in:

- Flow Diagram Requirements_Assistant

Structure:

Data item is primitive

3.4.1.76 Data Item Type

Description:

A type is pre-defined category which is used to classify Non_Link_Objects. Types have a hierarchical organization in which each type is a sub-category of some other more general type. A type includes a Type_Name, Date_Time_Info, and User_Name of the person who added it.

Data item is referenced in:

- Data Item Type_Hierarchy_Data

Structure:

Type_Name + Date_Time_Info + User_Name

3.4.1.77 Data Item Type_Hierarchy_Data

Description:

Type_Hierarchy_Data includes a collection of Types and the Parent/Child Relationships between them.

Data item is referenced in:

- Flow Diagram Picasso
- P-spec Manage_Type_Hierarchy
- Data Item Picasso_Data

Structure:

{Type} + {Parent_Child_Relation}

3.4.1.78 Data Item Type_Hierarchy_Query

Description:

A Type_Hierarchy_Query is a mechanism which will be used by translators and analysis tools to retrieve type hierarchy information to process conceptual graphs. The basic query specifies a search based on conditions for Name, Date/Time, and who added the entry. Like other search conditions, these may be keyword, pattern matching, equality, inequality, or greater than or less than. A further extension to this query allows the request to fetch ancestors or descendants of all the entries retrieved by the basic query. The ancestor condition can be specified to fetch all ancestors, ancestors back to the nth generation, ancestors beyond the nth generation, ancestors at the nth generation. Likewise, for a descendent condition, all descendent, descendent forward to the nth generation, descendent beyond the nth generation, and descendent at the nth generation

Data item is referenced in:

- Flow Diagram Picasso
- P-spec Manage_Type_Hierarchy
- Data Item Picasso_Data_Request

Structure:

(Type_Query) + (Ancestor_Descendent_Query)

3.4.1.79 Data Item Type_Name

Description:

This is an identifier which is used to classify the various Non_Link_Objects into specific, predefined categories.

Data item is referenced in:

- Data Item Actor
- Data Item Ancestor_Descendent_Query
- Data Item CG_Object_Query
- Data Item Concept
- Data Item Context
- Data Item Demon
- Data Item Relation
- Data Item Type
- Data Item Type_Query

Structure:

Data item is primitive

3.4.1.80 Data Item Type_Query

Description:

A Type_Query is a mechanism by which the tools may request Type_Hierarchy_Data by Type_Name, Date_Time_Info, or User_Name of the person who added it.

Data item is referenced in:

- Data Item Type_Hierarchy_Query

Structure:

(Type_Name) + (Date_Time_Info) + (User_Name)

3.4.1.81 Data Item Update_Ext_Tool_Config_Request

Description:

This is a type of Picasso_User_Request by which the User may update the Data in the External_Tool_Configuration data Store. The Update_External_Tool_Configuration_Request can be of two forms. It can be an Add_Request which allows the user to add information about new tool to the configuration, or it can be a Delete_Request which removes information about a tool in the current configuration.

Data item is referenced in:

- Flow Diagram Manage_External_Tools
- Flow Diagram Picasso
- P-spec Manage_External_Tools
- P-spec Picasso
- P-spec Update_Configuration_Data
- Data Item Picasso_User_Request

Structure:

[Add_Request | Delete_Request]

3.4.1.82 Data Item User_Command_Entry

Description:

This is a type of Metrics_Log_Entry which records each Picasso_User_Request received from the User. This includes User_Name, User Command, and User_Command_Results.

Data item is referenced in:

- Data Item Metrics_Log_Entry

Structure:

User_Name + Command_Name + User_Command_Results

3.4.1.83 Data Item User_Command_Query

Description:

A User_Command_Query is a mechanism which allows Metrics Tools to request Command_Log_Entries from the Metrics_Log. The available fields in User_Command_Query are Date/Time, User_Name, and User_Command_Results. These fields correspond to the fields in the User_Command log entries. The fields in the request are conditions that must be satisfied by the User_Command log entries that are returned when this request is sent to Picasso.

Data item is referenced in:

- Data Item Metrics_Data_Request

Structure:

(Date_Time_Info) + (User_Name) + (Command_Name) + (User_Command_Results)

3.4.1.84 Data Item User_Command_Results

Description:

This data item provides a text description of the results of Picasso_User_Requests indicating whether or not the request was success or unsuccessful. In addition, if request was unsuccessful, an explanation of why it failed (Error Code) should be included.

Data item is referenced in:

- Data Item User_Command_Entry
- Data Item User_Command_Query

Structure:

Data item is primitive

3.4.1.85 Data Item User_Location

Description:

This specifies the location of the user. Since the users are globally distributed, there must be some way to specify the user's location for reporting purposes.

Data item is referenced in:

- Data Item Add_Request
- Data Item Delete_Request
- Data Item Launch_Tool_Request
- Data Item Retrieve_CASE_Tool_View_Request
- Data Item Store_View_Request

Structure:

Data item is primitive

3.4.1.86 Data Item User_Name**Description:**

This a unique identifier used to specify a particular user of the system.

Data item is referenced in:

- Data Item Add_Request
- Data Item Analysis_Data_Query
- Data Item Analysis_Log_Entry
- Data Item Ancestor_Descendent_Query
- Data Item Author
- Data Item Delete_Request
- Data Item Launch_Tool_Request
- Data Item Resolution_Data_Query
- Data Item Resolution_Log_Entry
- Data Item Retrieve_CASE_Tool_View_Request
- Data Item Store_View_Request
- Data Item Type
- Data Item Type_Query
- Data Item User_Command_Entry
- Data Item User_Command_Query

Structure:

Data item is primitive

3.4.1.87 Data Item Version**Description:**

This is information about the version of the Requirements_View. As a view is modified during the project, new versions are created and stored in the Requirements_View Data Store.

Data item is referenced in:

- Data Item CASE_Tool_View_Query
- Data Item Requirements_View_Info

- Data Item Requirements_View_Info_Query

Structure:

Data item is primitive

3.4.1.88 Data Item View_Location

Description:

This provides the location of the CASE_Tool_View to Picasso. This is necessary for Picasso to be able to read and store the CASE_Tool_View in the Requirements_View Data Store.

Data item is referenced in:

- Data Item Store_View_Request

Structure:

Data item is primitive

3.4.1.89 Data Item View_Name

Description:

This is a unique identifier associated with each Requirements_View.

Data item is referenced in:

- Data Item CASE_Tool_View_Query
- Data Item Requirements_View_Info
- Data Item Requirements_View_Info_Query
- Data Item Store_View_Request

Structure:

Data item is primitive

3.4.2 Data Dictionary Hierarchy

3.4.2.1 Data Item Analysis_Report

Data item is primitive.

3.4.2.2 Data Item Analysis_Request

Data item is primitive.

3.4.2.3 Data Item CASE_Tool_Report

Data item is primitive.

3.4.2.4 Data Item CASE_Tool_Request

Data item is primitive.

3.4.2.5 Data Item External_Tool_Configuration

External_Tool_Configuration

```
\__ {Configuration_Data}  
    |__ Tool_Type  
    |__ +Tool_Name  
    |__ +Tool_Launch_Info  
    \__ +Associated_CASE_Tool
```

3.4.2.6 Data Item Launch_Analysis_Tool

Data item is primitive.

3.4.2.7 Data Item Launch_CASE_Tool

Data item is primitive.

3.4.2.8 Data Item Launch_Metrics_Tool

Data item is primitive.

3.4.2.9 Data Item Launch_Resolution_Tool

Data item is primitive.

3.4.2.10 Data Item Launch_Translator

Launch_Translator

```
|__ [Launch_Input_Translator  
\__ |Launch_Output_Translator]
```

3.4.2.11 Data Item Metrics_Data_Request

Metrics_Data_Request

```
|__ (Analysis_Data_Query)  
|  |__ (Entry_Name)  
|  |__ (Date_Time_Info)  
|  |__ (User_Name)  
|  |__ (Tool_Name)
```

- | |__ (Problem_Type)
- | |__ (Problem_Description)
- |__ (Resolution_Data_Query)
- | |__ (Entry_Name)
- | |__ (Date_Time_Info)
- | |__ (User_Name)
- | |__ (Tool_Name)
- | |__ (Action_Taken)
- |__ (User_Command_Query)
 - |__ (Date_Time_Info)
 - |__ (User_Name)
 - |__ (Command_Name)
 - |__ (User_Command_Results)

3.4.2.12 Data Item Metrics_Log

Metrics_Log

- |__ {Metrics_Log_Entry}
 - |__ Date_Time_Info
 - |__ [Analysis_Log_Entry
 - | |__ User_Name
 - | |__ +Tool_Name
 - | |__ +Problem_Type
 - | |__ +Problem_Description
 - |__ |Resolution_Log_Entry
 - | |__ User_Name
 - | |__ +Tool_Name
 - | |__ +Action_Taken
 - |__ |User_Command_Entry]
 - |__ User_Name
 - |__ +Command_Name
 - |__ +User_Command_Results

3.4.2.13 Data Item Metrics_Report

Data item is primitive.

3.4.2.14 Data Item Metrics_Request

Data item is primitive.

3.4.2.15 Data Item Picasso_Data

Picasso_Data

```
|__ (CG_Object)
|  |__ [Link_Object
|    |__ [Directed_Arc
|      |__ CG_Object_Name
|      |__ \__ (Label)
|      |__ |Dashed_Directed_Arc
|      |__ CG_Object_Name
|      |__ \__ (Label)
|      |__ \__ |Line_Of_Identity]
|      |__ CG_Object_Name
|      |__ \__ (Label)
|      |__ \__ [Non_Link_Object]
|      |__ [Context
|        |__ CG_Object_Name
|        |__ \__ +Type_Name
|        |__ |Concept
|        |__ CG_Object_Name
|        |__ +Type_Name
|        |__ \__ (Referent)
|        |__ |Relation
|        |__ CG_Object_Name
|        |__ +Type_Name
|        |__ \__ (Negation)
|        |__ |Actor
|        |__ CG_Object_Name
|        |__ +Type_Name
|        |__ \__ (Annotation)
|        |__ \__ |Demon]
```

```

|      |__ CG_Object_Name
|      |__ +Type_Name
|      |__ (Annotation)
|__ (Type_Hierarchy_Data)
|  |__ {Type}
|  |  |__ Type_Name
|  |  |__ +Date_Time_Info
|  |  |__ +User_Name
|  |__ {Parent_Child_Relation}
|__ (Requirements_View_Info)
|    |__ View_Name
|    |__ +Author
|    |__ ALIAS User_Name
|    |__ +Date_Time_Info
|    |__ +Version
|    |__ +Associated_CASE_Tool

```

3.4.2.16 Data Item Picasso_Data_Request

Picasso_Data_Request

```

|__ (CG_Object_Query)
|  |__ (CG_Object_Name)
|  |__ (Type_Name)
|  |__ (Referent)
|  |__ (Label)
|  |__ (Annotation)
|__ (Type_Hierarchy_Query)
|  |__ (Type_Query)
|  |  |__ (Type_Name)
|  |  |__ (Date_Time_Info)
|  |  |__ (User_Name)
|  |__ (Ancestor_Descendent_Query)
|    |__ (Type_Name)
|    |__ (Date_Time_Info)
|    |__ (User_Name)

```

- | |__ (Direction)
- | __ (Levels)
- __ (Requirements_View_Info_Query)
 - |__ (View_Name)
 - |__ (Author)
 - | __ ALIAS User_Name
 - |__ (Date_Time_Info)
 - |__ (Version)
 - __ (Associated_CASE_Tool)

3.4.2.17 Data Item Picasso_Report

Picasso_Report

- |__ [Confirmation_Report]
- __ [Error_Report]

3.4.2.18 Data Item Picasso_User_Request

Picasso_User_Request

- |__ [Launch_Tool_Request]
 - | |__ User_Location
 - | |__ +User_Name
 - | |__ +Tool_Type
 - | |__ [Tool_Name]
 - | __ [Associated_CASE_Tool]
- |__ [Store_View_Request]
 - | |__ User_Location
 - | |__ +User_Name
 - | |__ +View_Name
 - | |__ +View_Location
 - | __ +Associated_CASE_Tool
- |__ [Retrieve_CASE_Tool_View_Request]
 - | |__ User_Location
 - | |__ +User_Name

```

|  \__ +CASE_Tool_View_Query
|      |__ (View_Name)
|      |__ (Date_Time_Info)
|      |__ (Author)
|      |  \__ ALIAS User_Name
|      |__ (Version)
|      \__ (Associated_CASE_Tool)
\__ [Update_Ext_Tool_Config_Request]
    |__ [Add_Request
    |  |__ User_Location
    |  |__ +User_Name
    |  |__ +Tool_Type
    |  |__ +Tool_Name
    |  |__ +Tool_Launch_Info
    |  \__ (Associated_CASE_Tool)
    \__ [Delete_Request]
        |__ User_Location
        |__ +User_Name
        \__ +Tool_Name

```

3.4.2.19 Data Item Requirements_View_Data

Requirements_View_Data

```

\__ {Requirements_View}
    |__ CASE_Tool_View
    |__ +Requirements_View_Info
    |  |__ View_Name
    |  |__ +Author
    |  |  \__ ALIAS User_Name
    |  |__ +Date_Time_Info
    |  |__ +Version
    |  \__ +Associated_CASE_Tool

```

```

\__ {CG_Object}
  |__ [Link_Object
    | |__ [Directed_Arc
      | | |__ CG_Object_Name
      | | \__ (Label)
      | |__ [Dashed_Directed_Arc
        | | |__ CG_Object_Name
        | | \__ (Label)
        | \__ [Line_Of_Identity]
        | |__ CG_Object_Name
        | | \__ (Label)
  \__ [Non_Link_Object]
    |__ [Context
      | |__ CG_Object_Name
      | \__ +Type_Name
    |__ [Concept
      | |__ CG_Object_Name
      | |__ +Type_Name
      | \__ (Referent)
    |__ [Relation
      | |__ CG_Object_Name
      | |__ +Type_Name
      | \__ (Negation)
    |__ [Actor
      | |__ CG_Object_Name
      | |__ +Type_Name
      | \__ (Annotation)
  \__ [Demon]
    |__ CG_Object_Name
    |__ +Type_Name
    \__ (Annotation)

```

3.4.2.20 Data Item Resolution_Report

Data item is primitive.

3.4.2.21 Data Item Resolution_Request

Data item is primitive.

3.4.2.22 Data Item Translator_Report

Data item is primitive.

3.4.2.23 Data Item Translator_Request

Data item is primitive.

3.5 Adaptation Requirements

3.5.1 Site Adaptation

Picasso will not have any specific requirements for adapting the CSCI to site-unique conditions.

3.5.2 Portability

Portability is an important requirement in the system because it must support existing and available CASE tools. These CASE tools may be available only on certain operating systems. Furthermore, it must run on the machines that are available to the software requirements researcher.

To help insure portability, the software must be implemented in a high level language. This includes such languages as C, C++, Java, Pascal, and FORTRAN. It does not include assembly language. To help further insure portability, experts in software portability should be consulted to help architect the system to minimize and modularize non-portable and machine dependent functions in the software..

Microsoft Windows is the most popular operating system for home and office computer systems. Therefore, many of the CASE tools currently on the market are targeted for the Microsoft Windows family of operating systems. Therefore, Picasso must run on the Microsoft Windows 95. This will also show, to a reasonable degree of certainty, that the software is portable to many different operating systems, such as Windows NT and the upcoming Memphis. It will also demonstrate a high degree of certainty that the software will be portable to different hardware architectures such as PowerPC, MIPS, and Alpha which can run Windows NT, and Macintosh which can be made to emulate or run Windows 95.

UNIX is also a very popular operating system for software development environments. It is very popular among academic institutions. Consequently, many CASE tools are targeted for the UNIX platform. Since this system is targeted for academic research, and a graphical user interface is also desired, Picasso will be portable to the X-Windows operating system. The ability to run under UNIX and X-Windows will also demonstrate that the software will be portable to many workstations manufactured by Sun, Silicon Graphics, IBM, and Hewlett-Packard.

3.5.3 Maintainability

It is important that the system be maintainable since it is an experimental system and will be heavily modified during the course of its life. It is important to minimize cost of experiments and minimize the learning curve required by a programmer to master the system architecture and be able to make modifications.

To help insure maintainability of the software, it is required that a documented coding practice be followed in developing the software. This coding practice should include a function and variable naming notation (such as Microsoft's Hungarian Notations), a formatting convention, and a code commenting convention. Since different organizations usually have a documented coding practice, and research indicates that following a code practice is helpful to maintainability, but is

inconclusive about whether a particular coding practice is better than others, the selection of a particular code practice will be left to the implementation team.

3.6 Sizing and Timing Requirements

3.6.1 Sizing

It is difficult to estimate the memory and storage requirements of the software, particularly without specifying the system and software architecture. Furthermore, the size of the data stores depend greatly on the number of users and exactly how each user chooses to use the system. It is not possible to predict such usage since this is one of the factors that this system is meant to explore.

However, this section will attempt to identify some of the factors and the realationships between them that will determine the size of the largest data store, the Requirements View data store.

$$C = N_v [S_{CT} (N_{CGO} S_{CGO}) (1-k)]$$

C: Capacity

N_v : number of requirement views

S_{CT} : size of an average CASE tool view

N_{CGO} : average number of conceptual graphs generated from an average CASE tool view

S_{CGO} : size of an average conceptual graph object

k: amount of overlap between requirements views ($0 \leq k \leq 1$)

The size of the Requirements View data store will be determined largely by the number and size of CASE tool view documents stored plus the number and size of the conceptual graph objects that are created from that CASE tool view document. The number of CASE tool view documents that are checked in will be determined by the number of users and how often they check in information will greatly affect . It is expected that up to 50 users will use the system at a time. The size of each conceptual graph object will be determined by how the system implementors build the data types.

However, it is desirable (but not a requirement) that the software will run on a typical personal computer. At the time of this document, a typical new personal computer is at least a 120 MHz Pentium machine with 32 Mb RAM and a 1 gigabyte hard drive.

3.6.2 Timing

The time for an operation will be equal to $T_O + T_N$. T_O is the time for the operation and T_N is the time for the request and data to travel across the network and back. Since the design phase will determine the type of the network or even if a network will be used, it is not possible to determine the magnitude of the network delay.

Below are some example operations and the variables to which T_O will be proportional. Most of the operations involve queries. A query can be based on a number of properties, so a linear order search is most likely.

Retrieving a conceptual graph object	N or N^2 , when N is the number of conceptual graph objects in the system. It depends if the request is for a conceptual graph object, or for all conceptual graph objects linked to the conceptual graph objects specified.
Retrieving a CASE tool view	N , where N is the number of CASE tool views in the system.
Storing a requirements view	Dependent on the size of the requirements view which comprised of the size of the CASE tool view plus the number of conceptual graph objects generated from the CASE tool view.
Retrieving a Type Hierarchy entry	$\log N$, where N is the number of entries in the Type Hierarchy. The worst case is N . The Type Hierarchy is a tree, so a search should have a logarithmic time characteristic.
Updating an External Tool Configuration entry	N , where N is the number of External Tools in the External Tool data store.
Retrieving a Metric Log Entry	N , where N is the number of Metric Log Entries.

Table 3-1 Operation Time Parameters

3.7 Safety Requirements

Picasso will not have any potential hazards to personnel property and the physical environment; therefore, it will not have any safety requirements.

3.8 Security Requirements

Picasso will be used as for academic research purposes; therefore, the system will not have any security requirements with respect to potential compromise of sensitive data.

3.9 Design Constraints

The most common operating systems used on the machines for which Picasso is targeted are Windows 95 and Windows NT; therefore, the software must be able to operate on both of these operating systems.

3.10 Software Quality Factors

Software quality issues should be addressed in a Quality Assurance Plan, therefore, they will not be described in this document.

3.11 Human Performance/Human Engineering Requirements

Picasso should be easy for new users to learn, therefore it will use the standard Microsoft Windows interface with which most potential users will be familiar. In addition, it should provide convenient mechanisms (e.g. pop-up menus, toolbars, and shortcuts) for frequently used operations. The user should be able to access functions quickly without going through a multitude of menu levels; therefore, no menus will be more than three levels deep. Furthermore, program text should be easy to read; therefore, all text will use a font size of at least 10 points.

The user may want to use one tool while another tool is busy working; therefore, the user must be able to pull up the Picasso windows and enter another request while an external tool is running. This is commonly referred to as being non-modal.

The software should be responsive; therefore, most operations should complete within a few seconds and the more complex operations should complete within minutes. Picasso will also provide confirmation messages or error messages for all user operations. Error messages will consist of an error message that can be understood by a typical Picasso user and an error identification number that will be used to aid in technical support. As stated in the scope of this document, the typical Picasso user will have a basic knowledge of computers, including the Windows 95 operating system. They will also be familiar with at least one CASE tool used for requirements engineering.

3.12 Reliability Requirements

3.12.1 Robustness

Picasso will respond to all requests by either a confirmation or error messages to the user or to the external tools. In the event of invalid inputs from the user, Picasso will provide an option for the user to correct the problem. For example, in case of an invalid request from a user, Picasso will prompt the user of an error and provide an option to reenter the request or void the request entirely. In the case of an invalid request or invalid data received from the external tools, Picasso will generate an error message to the appropriate external tool.

Backing up the program data and recovering previous versions of the data is left to the computer system administrator and falls outside the scope of Picasso s functionality.

3.12.2 Data Integrity

Due to the concurrent nature of the Picasso application, data integrity becomes a critical issue. To preserve the integrity of the information in the data stores, all data stores will support multiple reads, but not multiple writes. At any one time, there can be multiple reads or just one write operation. A read and a write are not allowed since it cannot be known if the read operation will acquire the old information, the updated information, or some combination.

3.13 Traceability Matrix

3.13.1 Governing Requirements Traced to Engineering Requirements

3.13.1.1 Trace Item Cross-references

3.13.1.1.1 R1

Requirement:

For every user request, Picasso will generate either a Confirmation_Report or an Error_Report to the user indicating whether or not the request was satisfied.

Is satisfied by data processes:

- Launch_External_Tools
- Manage_External_Tools
- Manage_Reqs_Views
- Picasso
- Update_Configuration_Data

is satisfied by data items:

- Confirmation_Report
- Error_Report
- Picasso_Report
- Picasso_User_Request
- User_Location

3.13.1.1.2 R2

Requirement:

Picasso will automatically log all user requests. Log entries will include the date/time stamp, User_Name, user request, and results. Picasso will provide a mechanism that will allow metrics tools to retrieve log entries by any combination of these attributes.

Is satisfied by data processes:

- Manage_Metrics_Log
- Picasso

is satisfied by data items:

- Command_Name
- Date_Time_Info

- Entry_Name
- Metrics_Data_Request
- Metrics_Log
- Metrics_Log_Entry
- Picasso_User_Request
- User_Command_Entry
- User_Command_Query
- User_Command_Results
- User_Name

3.13.1.1.3 R3

Requirement:

Picasso will maintain an External_Tool_Configuration data whose entries will include the Tool_Name, Tool_Type (e.g. analysis, resolution, metrics, translator) and how to launch the tool. Entries for translator tools will also include the name of the Associated_CASE_Tool. Picasso will provide a mechanism that will allow users to add new tools and/or delete tools from the current configuration.

Is satisfied by data processes:

- Manage_External_Tools
- Picasso
- Update_Configuration_Data

is satisfied by data items:

- Add_Request
- Associated_CASE_Tool
- Configuration_Data
- Delete_Request
- External_Tool_Configuration
- Picasso_Report
- Picasso_User_Request
- Tool_Launch_Info
- Tool_Name
- Tool_Type
- Update_Ext_Tool_Config_Request

3.13.1.1.4 R4

Requirement:

Picasso will provide a mechanism for users to Launch_External_Tools (e.g. analysis tools, problem resolution tools, and metrics tools). Picasso will launch the specified tool using the information contained in the External_Tool_Configuration data.

Is satisfied by data processes:

- Launch_External_Tools
- Manage_External_Tools
- Picasso

is satisfied by data items:

- Configuration_Data
- External_Tool_Configuration
- Launch_Analysis_Tool
- Launch_Metrics_Tool
- Launch_Resolution_Tool
- Launch_Tool_Request
- Picasso_Report
- Picasso_User_Request
- Tool_Launch_Info
- Tool_Name
- Tool_Type

3.13.1.1.5 R5

Requirement:

Picasso will log all problems reported by the analysis tools. Each log entry will include the Tool_Name, User_Name, time/date stamp, Problem_Type, and Problem_Description. Picasso will provide a mechanism for metrics tools to retrieve entries by any combination of these attributes.

Is satisfied by data processes:

- Manage_Metrics_Log
- Picasso

is satisfied by data items:

- Analysis_Data_Query
- Analysis_Log_Entry
- Date_Time_Info
- Entry_Name
- Metrics_Data_Request
- Metrics_Log
- Metrics_Log_Entry
- Problem_Description
- Problem_Type
- User_Name

3.13.1.1.6 R6

Requirement:

Picasso will log all corrective actions reported by the resolution tools. Each log entry will include the Tool_Name, User_Name, time/date stamp, Action_Taken, and associated problems. Picasso will provide a mechanism for metrics tools to retrieve entries by any combination of these attributes.

Is satisfied by data processes:

- Manage_Metrics_Log
- Picasso

is satisfied by data items:

- Action_Taken
- Date_Time_Info
- Entry_Name
- Metrics_Data_Request
- Metrics_Log
- Metrics_Log_Entry
- Resolution_Data_Query
- Resolution_Log_Entry
- User_Name

3.13.1.1.7 R7

Requirement:

Picasso will provide a mechanism that will allow users to store a Requirements_View generated by any CASE tool, provided that an input translator is available that will generate a conceptual graph representation of that view. Picasso will retrieve the specified Requirements_View from the CASE tool and launch the input translator associated with that CASE tool in the External_Tool_Configuration data. Picasso will store the original view as generated by the CASE tool and the conceptual graph representation as generated by the input translator. Picasso will also store the View_Name, time/date stamp, User_Name, and the CASE tool associated with the view.

Is satisfied by data processes:

- Launch_External_Tools
- Manage_External_Tools
- Manage_Reqs_Views
- Picasso

is satisfied by data items:

- Actor
- Annotation
- Associated_CASE_Tool
- Author
- CASE_Tool_View
- CG_Object
- CG_Object_Name
- Concept
- Configuration_Data
- Context
- Dashed_Directed_Arc
- Date_Time_Info
- Demon
- Directed_Arc
- External_Tool_Configuration
- Label
- Launch_Input_Translator

- Launch_Translator
- Line_Of_Identity
- Link_Object
- Negation
- Non_Link_Object
- Picasso_Report
- Picasso_User_Request
- Referent
- Relation
- Requirements_View
- Requirements_View_Data
- Requirements_View_Info
- Store_View_Request
- Tool_Launch_Info
- Type_Name
- User_Name
- Version
- View_Location
- View_Name

3.13.1.1.8 R8

Requirement:

Picasso will provide a mechanism for retrieving and displaying Requirements_Views with the original CASE tool used to generate those views. Picasso will provide a mechanism for users to retrieve a Requirements_View by View_Name, date/time stamp, User_Name, Associated_CASE_Tool, or any combination of the above. Picasso will launch the CASE tool associated with the view and send the original Requirements_View to the CASE tool.

Is satisfied by data processes:

- Launch_External_Tools
- Manage_External_Tools
- Manage_Reqs_Views
- Picasso

is satisfied by data items:

- Associated_CASE_Tool
- Author
- CASE_Tool_View
- CASE_Tool_View_Query
- Configuration_Data
- Date_Time_Info
- External_Tool_Configuration
- Launch_CASE_Tool
- Picasso_Report
- Picasso_User_Request
- Requirements_View
- Requirements_View_Data
- Requirements_View_Info
- Retrieve_CASE_Tool_View_Request
- Tool_Launch_Info
- User_Location
- User_Name
- Version
- View_Name

3.13.1.1.9 R9

Requirement:

Picasso will provide a mechanism for retrieving and displaying Requirements_Views with any CASE tool provided that an appropriate output translator is available. This translator must be able to convert the internal conceptual graph representation of the views into the specified CASE tool format. Picasso will launch the output translator associated with the CASE tool in the External_Tool_Configuration data.

Is satisfied by data processes:

- Launch_External_Tools
- Manage_External_Tools
- Manage_Reqs_Views
- Picasso

is satisfied by data items:

- Actor
- Annotation
- Author
- CG_Object
- CG_Object_Name
- CG_Object_Query
- Concept
- Configuration_Data
- Context
- Dashed_Directed_Arc
- Date_Time_Info
- Demon
- Directed_Arc
- External_Tool_Configuration
- Label
- Launch_Output_Translator
- Launch_Tool_Request
- Launch_Translator
- Line_Of_Identity
- Link_Object
- Negation
- Non_Link_Object
- Picasso_Report
- Picasso_User_Request
- Referent
- Relation
- Requirements_View
- Requirements_View_Data
- Requirements_View_Info
- Requirements_View_Info_Query

- Tool_Launch_Info
- Type_Name
- User_Location
- User_Name
- Version
- View_Name

3.13.1.1.10 R10

Requirement:

Picasso will provide a mechanism for external tools to retrieve the conceptual graph representation of Requirements_Views by View_Name, date/time stamp, User_Name, Associated_CASE_Tool, conceptual graph information, or any combination of the above.

Is satisfied by data processes:

- Manage_Reqs_Views
- Manage_External_Tools
- Picasso

is satisfied by data items:

- Actor
- Annotation
- CG_Object
- CG_Object_Name
- CG_Object_Query
- Concept
- Context
- Dashed_Directed_Arc
- Date_Time_Info
- Demon
- Directed_Arc
- Label
- Line_Of_Identity
- Link_Object
- Negation

- Non_Link_Object
- Picasso_Data
- Picasso_Data_Request
- Referent
- Relation
- Requirements_View
- Requirements_View_Data
- Requirements_View_Info
- Requirements_View_Info_Query
- Type_Name
- User_Name
- Version
- View_Name

3.13.1.1.11 R11

Requirement:

Picasso will maintain a hierarchy of types which will be used by external tools to manipulate conceptual graph information. Type entries will include the date/time stamp, User_Name, Type_Name, and parent Type_Name. Picasso will provide a mechanism that will allow external tools to retrieve entries by any combination of these attributes. In addition, Picasso will provide a mechanism that will allow resolution tools to add new types to the hierarchy.

Is satisfied by data processes:

- Manage_Type_Hierarchy
- Picasso

is satisfied by data items:

- Ancestor_Descendent_Query
- Date_Time_Info
- Direction
- Levels
- Parent_Child_Relation
- Picasso_Data
- Picasso_Data_Request
- Type

- Type_Hierarchy_Data
- Type_Hierarchy_Query
- Type_Name
- Type_Query
- User_Name

3.13.2 Engineering Requirements Traced to Governing Requirements

3.13.2.1 Process Cross-References

3.13.2.1.1 Launch_External_Tools

Satisfies requirements:

- R1
- R4
- R7
- R8
- R9

3.13.2.1.2 Manage_External_Tools

Satisfies requirements:

- R1
- R3
- R4
- R7
- R8
- R9
- R10

3.13.2.1.3 Manage_Metrics_Log

Satisfies requirements:

- R2
- R5
- R6

3.13.2.1.4 Manage_Reqts_Views

Satisfies requirements:

- R1
- R7
- R8
- R9
- R10

3.13.2.1.5 Manage_Type_Hierarchy

Satisfies requirements:

- R11

3.13.2.1.6 Picasso

Satisfies requirements:

- R1
- R2
- R3
- R4
- R5
- R6
- R7
- R8
- R9
- R10
- R11

3.13.2.1.7 Requirements_Assistant

Satisfies requirements:

None

3.13.2.1.8 Retrieve_CG_Object_Data

Satisfies requirements:

None

3.13.2.1.9 Retrieve_Original_View

Satisfies requirements:

None

3.13.2.1.10 Store_Requirements_View

Satisfies requirements:

None

3.13.2.1.11 Update_Configuration_Data

Satisfies requirements:

- R1
- R3

3.13.2.2 Data Item Cross References

3.13.2.2.1 Action_Taken

Satisfies Requirements:

- R6

3.13.2.2.2 Actor

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.3 Add_Request

Satisfies Requirements:

- R3

3.13.2.2.4 Analysis_Data_Query

Satisfies Requirements:

- R5

3.13.2.2.5 Analysis_Log_Entry

Satisfies Requirements:

- R5

3.13.2.2.6 Analysis_Report

Satisfies Requirements:

3.13.2.2.7 Analysis_Request

Satisfies Requirements:

3.13.2.2.8 Ancestor_Descendent_Query

Satisfies Requirements:

- R11

3.13.2.2.9 Annotation

Satisfies Requirements:

- R7

- R9

- R10

3.13.2.2.10 Associated_CASE_Tool

Satisfies Requirements:

- R3

- R7

- R8

3.13.2.2.11 Author

Satisfies Requirements:

- R7

- R8

- R9

3.13.2.2.12 CASE_Tool_Report

Satisfies Requirements:

3.13.2.2.13 CASE_Tool_Request

Satisfies Requirements:

3.13.2.2.14 CASE_Tool_View

Satisfies Requirements:

- R7

- R8

3.13.2.2.15 CASE_Tool_View_Query

Satisfies Requirements:

- R8

3.13.2.2.16 CG_Object

Satisfies Requirements:

- R7

- R9

- R10

3.13.2.2.17 CG_Object_Name

Satisfies Requirements:

- R7

- R9

- R10

3.13.2.2.18 CG_Object_Query

Satisfies Requirements:

- R9

- R10

3.13.2.2.19 Command_Name

Satisfies Requirements:

- R2

3.13.2.2.20 Concept

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.21 Configuration_Data

Satisfies Requirements:

- R3
- R4
- R7
- R8
- R9

3.13.2.2.22 Confirmation_Report

Satisfies Requirements:

- R1

3.13.2.2.23 Context

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.24 Dashed_Directed_Arc

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.25 Date_Time_Info

Satisfies Requirements:

- R2
- R5

- R6
- R7
- R8
- R9
- R10
- R11

3.13.2.2.26 Delete_Request

Satisfies Requirements:

- R3

3.13.2.2.27 Demon

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.28 Directed_Arc

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.29 Direction

Satisfies Requirements:

- R11

3.13.2.2.30 Entry_Name

Satisfies Requirements:

- R2
- R5
- R6

3.13.2.2.31 Error_Report

Satisfies Requirements:

- R1

3.13.2.2.32 External_Tool_Configuration

Satisfies Requirements:

- R3
- R4
- R7
- R8
- R9

3.13.2.2.33 Label

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.34 Launch_Analysis_Tool

Satisfies Requirements:

- R4

3.13.2.2.35 Launch_CASE_Tool

Satisfies Requirements:

- R8

3.13.2.2.36 Launch_Input_Translator

Satisfies Requirements:

- R7

3.13.2.2.37 Launch_Metrics_Tool

Satisfies Requirements:

- R4

3.13.2.2.38 Launch_Output_Translator

Satisfies Requirements:

- R9

3.13.2.2.39 Launch_Resolution_Tool

Satisfies Requirements:

- R4

3.13.2.2.40 Launch_Tool_Request

Satisfies Requirements:

- R4

- R9

3.13.2.2.41 Launch_Translator

Satisfies Requirements:

- R7

- R9

3.13.2.2.42 Levels

Satisfies Requirements:

- R11

3.13.2.2.43 Line_Of_Identity

Satisfies Requirements:

- R7

- R9

- R10

3.13.2.2.44 Link_Object

Satisfies Requirements:

- R7

- R9

- R10

3.13.2.2.45 Metrics_Data_Request

Satisfies Requirements:

- R2
- R5
- R6

3.13.2.2.46 Metrics_Log

Satisfies Requirements:

- R2
- R5
- R6

3.13.2.2.47 Metrics_Log_Entry

Satisfies Requirements:

- R2
- R5
- R6

3.13.2.2.48 Metrics_Report

Satisfies Requirements:

3.13.2.2.49 Metrics_Request

Satisfies Requirements:

3.13.2.2.50 Negation

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.51 Non_Link_Object

Satisfies Requirements:

- R7
- R9
- R10

3.13.2.2.52 Parent_Child_Relation

Satisfies Requirements:

- R11

3.13.2.2.53 Picasso_Data

Satisfies Requirements:

- R10
- R11

3.13.2.2.54 Picasso_Data_Request

Satisfies Requirements:

- R10
- R11

3.13.2.2.55 Picasso_Report

Satisfies Requirements:

- R1
- R3
- R4
- R7
- R8
- R9

3.13.2.2.56 Picasso_User_Request

Satisfies Requirements:

- R1
- R2
- R3
- R4
- R7
- R8
- R9

3.13.2.2.57 Problem_Description

Satisfies Requirements:

- R5

3.13.2.2.58 Problem_Type

Satisfies Requirements:

- R5

3.13.2.2.59 Referent

Satisfies Requirements:

- R7

- R9

- R10

3.13.2.2.60 Relation

Satisfies Requirements:

- R7

- R9

- R10

3.13.2.2.61 Requirements_View

Satisfies Requirements:

- R7

- R8

- R9

- R10

3.13.2.2.62 Requirements_View_Data

Satisfies Requirements:

- R7

- R8

- R9

- R10

3.13.2.2.63 Requirements_View_Info

Satisfies Requirements:

- R7
- R8
- R9
- R10

3.13.2.2.64 Requirements_View_Info_Query

Satisfies Requirements:

- R9
- R10

3.13.2.2.65 Resolution_Data_Query

Satisfies Requirements:

- R6

3.13.2.2.66 Resolution_Log_Entry

Satisfies Requirements:

- R6

3.13.2.2.67 Resolution_Report

Satisfies Requirements:

3.13.2.2.68 Resolution_Request

Satisfies Requirements:

3.13.2.2.69 Retrieve_CASE_Tool_View_Request

Satisfies Requirements:

- R8

3.13.2.2.70 Store_View_Request

Satisfies Requirements:

- R7

3.13.2.2.71 Tool_Launch_Info

Satisfies Requirements:

- R3
- R4
- R7
- R8
- R9

3.13.2.2.72 Tool_Name

Satisfies Requirements:

- R3
- R4

3.13.2.2.73 Tool_Type

Satisfies Requirements:

- R3
- R4

3.13.2.2.74 Translator_Report

Satisfies Requirements:

3.13.2.2.75 Translator_Request

Satisfies Requirements:

3.13.2.2.76 Type

Satisfies Requirements:

- R11

3.13.2.2.77 Type_Hierarchy_Data

Satisfies Requirements:

- R11

3.13.2.2.78 Type_Hierarchy_Query

Satisfies Requirements:

- R11

3.13.2.2.79 Type_Name

Satisfies Requirements:

- R7
- R9
- R10
- R11

3.13.2.2.80 Type_Query

Satisfies Requirements:

- R11

3.13.2.2.81 Update_Ext_Tool_Config_Request

Satisfies Requirements:

- R3

3.13.2.2.82 User_Command_Entry

Satisfies Requirements:

- R2

3.13.2.2.83 User_Command_Query

Satisfies Requirements:

- R2

3.13.2.2.84 User_Command_Results

Satisfies Requirements:

- R2

3.13.2.2.85 User_Location

Satisfies Requirements:

- R1
- R8
- R9

3.13.2.2.86 User_Name

Satisfies Requirements:

- R2
- R5
- R6
- R7
- R8
- R9
- R10
- R11

3.13.2.2.87 Version

Satisfies Requirements:

- R7
- R8
- R9
- R10

3.13.2.2.88 View_Location

Satisfies Requirements:

- R7

3.13.2.2.89 View_Name

Satisfies Requirements:

- R7
- R8
- R9
- R10

4. Qualification Requirements

Software testing requirements should be addressed in a Software Test Plan; therefore, no qualification requirements will be described in this document.

5. Preparation for delivery

Picasso will not have any special requirements for delivery.

6. Notes

6.1 Glossary

Action_Taken	This is a textual description of the corrective action taken by the user.
Actor	An Actor is a type of Non_Link_Object which is used to represent a processing function. Objects that are connected in the direction toward the Actor are inputs. Objects that are connected in the direction away from the Actor are outputs. Actors often include a description of the algorithm used to transform the inputs into the outputs. This description is called an Annotation.
Add_Request	An Add_Request is a type of Update External_Tool_Configuration data Request which allows the User to add a new tool into the External_Tool_Configuration data. All Add_Requests will include the Tool_Type and the Tool_Name. If the tool is a Translator, the Add_Request will include the name of the Associated_CASE_Tool for which the translator is written.
Ambiguity	Refers to differences between views that are not necessarily due to inconsistency or incompleteness.
Analysis_Data_Query	The Analysis_Data_Query is used to request metrics entry logs which were created by the analysis tools. The metrics tools may request entries by Entry_Name, Date_Time_Info, User_Name, Tool_Name, Problem_Type and Problem_Description. In addition, any combination of these keys is possible.
Analysis_Log_Entry	An Analysis_Log_Entry contains the results of analysis to be stored in the Metrics_Log. It includes the User_Name, Analysis Tool_Name, Problem_Type, and a textual Problem_Description.
Analysis_Report	This includes all data transmitted from the analysis tool directly to the user. The content of this data item will vary depending on the which analysis tool is being used. This may include screen displays, error messages, notifications, warnings, beeps, etc.

Analysis_Request	This includes all data transmitted from the user directly to the analysis tool. The content of this data item will vary depending on which analysis tool is being used.
Analysis Tools	The software tools that are used to check for consistency, completeness, and ambiguity between multiple Requirements_Views, and to generate Analysis_Reports and analysis log entries.
Ancestor_Descendant_Query	This is a mechanism by which the Translators, Analysis Tools, and Resolution Tools may request Type_Hierarchy_Data based on Parent/Child Relationships. The query may include the Type_Name, Date_Time_Info, User_Name of the person who added it, Direction, and the number of Levels.
Annotation	A textual description of the algorithm used to transform the inputs of an Actor into the outputs of the Actor.
Associated_CASE_Tool	This provides the Picasso software with the necessary information to determine which Translator to launch when storing or retrieving CASE_Tool_Views and/or CG_Objects.
Author	The User_Name of the person who originally stored the CASE_Tool_View.
CASE_Tool_Report	This includes any data transmitted directly from the CASE tool to the user. The content will vary depending on which CASE tool is being used. This may include screen displays, error messages, notifications, warnings, beeps, etc.
CASE_Tool_Request	This includes any data transmitted from the user directly to the CASE tool. The content will vary depending on which CASE tool is being used.
CASE Tools	Computer Aided Software Engineering Tools. Any software tool specifically used for Computer Aided Software Requirements development.
CASE_Tool_View	The CASE_Tool_View is the original data from the CASE tool. This is saved along with a Conceptual Graph representation in the Requirements_View database. The original data can then be retrieved by

any user, if the user has the appropriate CASE tool. Its content will vary depending on the CASE tool used to create it.

CASE_Tool_View_Query

The CASE_Tool_View_Request recovers all CASE tool views from Picasso by specifying conditions for the fields, Name, Author, Date/Time, and Version. All CASE tool views that are returned to the requester will satisfy the conditions specified in the request.

CG_Object

Conceptual Graph Objects (AKA CG_Objects) are a knowledge representation of the various processes, data items, events, entities, and states in a Requirements_View. The external translators can either take a CASE Tool View and convert it into a collection of CG_Objects, or take a collection of CG_Objects and convert it into a CASE Tool View. CG_Objects fall into two categories: Link_Objects and Non_Link_Objects.

CG_Object_Name

A CG_Object_Name is a unique identifier associated with each CG_Object.

CG_Object_Query

The CG_Object_Request can Retrieve_CG_Objects based on the Requirements_View of which they are a part. They can be requested by specifying the Name, Author, Date/Time, or Version. Other conditions based on the characteristics of the individual CG_Objects can also be specified. These include the Type (ex. Concept, Context, Relation, etc.), Referent (used for Concepts), Label (used for Links), or Annotation (used for Demons and Actors).

Command_Name

This is a unique identifier that identifies the type of command that was issued by the user.

Concept

A Concept is a Non_Link_Object used to represent an idea or thought. A Concept may include a unique identifier, called a referent, to distinguish it from other concepts of the same type.

Configuration_Data Configuration_Data contains the name of the External Tool used, the type of the tool, the external Tool_Launch_Info, and the Associated_CASE_Tool information.

Confirmation_Report

The Confirmation_Report notifies the user with confirmed message.

Context	A Context is a type of Non_Link_Object used to group objects that belong together. This may be with respect to time, subsystem, or textual context. It may contain other CG_Objects, including other contexts.
Dashed_Directed_Arc	A Dashed_Directed_Arc is used to connect either a Concept or Context with either an Actor or Demon that is in the same context. For an Actor, the direction of the arrow shows which concepts/contextes are inputs to the actor and which are outputs of the actor. For a demon, the direction of the relation show the time sequence in which concepts/contextes come into existence and which cease to exist.
Date_Time_Info	This contains date and time information. It is typically used as a time stamp for logging various requests to the software.
Delete_Request	A Delete_Request is a type of Update External_Tool_Configuration data Request which allows the User to delete tools from the External_Tool_Configuration data store. The Delete_Request takes the Tool_Name and causes the entry in the External_Tool_Configuration data store with that Tool_Name to be deleted.
Demon	A Demon is a type of Non_Link_Object that is used to represent time sequences. Objects that are linked towards the Demon are said to go out of existence. Objects that are linked away from the Demon are said to come into existence.
Directed_Arc	A Directed_Arc is used to connect a Concept with a Relation that is in the same context. The direction of the arrow is used to indicate the direction of the relation.
Direction	This data indicates whether tool is requesting the ancestors, descendants, or both.
Entry_Name	The Entry_Name is a unique identifier associated with each entry in the Metrics_Log.
Error_Report	The Error_Report notifies the user with error type.
External_Tool_Configuration	External_Tool_Configuration Store contains Configuration_Data.

Incompleteness	Refers to information that is contained in one view, but is missing in another.
Inconsistency	Refers to conflicting requirements between views. This may include conflicts such as differences in the software's functionality, timing, non-functional aspects, and data item descriptions.
Label	A Label is an optional word that provides additional information about the Link_Object.
Launch_Analysis_Tool	This is the control signal from Picasso used to launch the various external analysis tools. The information on how to launch the analysis tools is stored in the External_Tool_Configuration data store.
Launch_CASE_Tool	This is the control signal from Picasso used to launch the various CASE tools. The information on how to launch the CASE tools is stored in the External_Tool_Configuration data store. The CASE tool is used as an external viewer for the CASE tool views.
Launch_External_Tools	A process which launches the appropriate external tool based on Tool_Name and Associated_CASE_Tool. The process will search for the appropriate Configuration_Data from the External_Tool_Configuration Store based on the Query Data.
Launch_Input_Translator	This is the control signal from Picasso used to launch the input translator tools.
Launch_Metrics_Tool	This is the control signal from Picasso used to launch the various external metrics tools. The information on how to launch the metrics tools is stored in the External_Tool_Configuration data store.
Launch_Output_Translator	This is the control signal from Picasso used to launch the output translator tools.
Launch_Resolution_Tool	This is the control signal from Picasso used to launch the various external problem resolution tools. The information on how to launch the resolution tools is stored in the External_Tool_Configuration data store.
Launch_Tool_Request	This is a type of Picasso_User_Request that specifies that the user wants to launch either an analysis tool, a metrics tool, problem resolution tool, or an output

translator. It includes all the information necessary to perform a lookup in the External_Tool_Configuration data store and launch the tool. This information is either the name of the tool or the name of the Associated_CASE_Tool.

Launch_Translator

This is the control signal from Picasso used to launch the various external translator tools. The information on how to launch the translator tools is stored in the External_Tool_Configuration data store.

Levels

This specifies how many levels up or down the query should extend from the beginning type.

Line_Of_Identity

A Line_of_Identity is a type of Link Object that is used to connect two Concepts. It shows that the two objects are really the same object.

Link_Object

A Link Object is a type of Conceptual Graph Object used to connect two Non-Link Objects Together. There are three types of Link Objects: Directed_Arcs, Dashed_Directed_Arcs, and Lines of Identity. A Link Object may have an optional Label.

Manage_External_Tools

A process which launches the appropriate external tool requests by obtaining Configuration_Data from the External_Tool_Configuration data store. The process will also add or remove tool information from the External_Tool_Configuration data store.

Manage_Metrics_Log

A process which stores Picasso_User_Requests, Analysis_Log_Entry, and Resolution_Log_Entry to the Metrics_Log Store. The process will also retrieve Metrics_Log_Entry from the Metrics_Log Store and send results from Metrics_Data_Request.

Manage_Reqts_Views

A process which manages the Requirements_View_Data store to store and/or to retrieve a Requirements_View from the Requirements_View_Data store. This process is consisted of the following processes: Store_Requirements_View, Retrieve_Original_View and Retrieve_CG_Object_Data.

Manage_Type_Hierarchy

A process which manages the Type_Hierarchy_Data store to update and/or to retrieve Type_Hierarchy_Data to/from the Type_Hierarchy_Data store. The process will search

for the appropriate Type_Hierarchy_Data from the Type_Hierarchy_Data store based on the Type_Hierarchy_Query.

Metrics_Data_Request

The Metrics_Data_Request is used by the external metrics tools to request data from the Metrics_Log. A Metrics_Data_Request may include the three following types of requests: Analysis_Data_Query, Resolution_Data_Query, or User_Command_Query. In addition, these requests may be combined together to allow the user to perform an Analysis_Data_Query and use its results in a Resolution_Data_Query.

Metrics_Log

Metrics_Log Store contains Metrics_Log_Entry (s)

Metrics_Log_Entry

Metrics_Log_Entry is used as raw data for the metrics that the external metrics tool calculates. These metrics will be used to help describe how the system is being used, determine how well the current configuration of the system is performing, and measure the development process of a specific project. A Metrics_Log_Entry can be one of three different types: Analysis_Log_Entry, Resolution_Log_Entry, or User_Command_Entry. Every Metrics_Log_Entry has Date_Time_Info.

Metrics_Report

This includes all data transmitted from the various metrics tools directly to the user. The content will vary depending on which metrics tool is being used. This may include screen displays, error messages, notifications, warnings, beeps, etc.

Metrics_Request

This includes all data transmitted from the user directly to the metrics tools. The content will vary depending on which metrics tool being used.

Metrics_Tools

The software tools that retrieve metrics entry logs (log entries from analysis activities, problem resolution activities, and user requests to Picasso) from Picasso, calculate various project metrics, and generate Metrics_Reports.

Negation

This is a Boolean value that indicates whether or not the Context or Relation should be negated.

Non_Link_Object

A Non-Link Object can be any one of the following types: Context, Concept, Relation, Actor, or Demon. They are used to represent contexts, ideas,

relationships, processing functions, or time sequences. Every Non-Link Object has a Type. They are connected to each other by Link Objects.

Parent_Child_Relation

This contains information about the relationship between two types in the Type_Hierarchy_Data Store. The Parent/Child Relationship indicates a Generalization/Specialization kind of connection between the Types.

Picasso

The name of the software to be developed. Picasso functions as a major component of the Requirement Assistant. Picasso provides a flexible environment in which globally distributed users can work together effectively on project requirements development efforts. The Picasso will accept requests from the users, attempt to satisfy the requests, and send Confirmation_Reports and/or Error_Reports to the users.

Picasso_Data

Picasso_Data includes any data the software provides in response to Picasso_Data_Requests from the translators, analysis tools, or resolution tools. It may include conceptual graph objects, Type_Hierarchy_Data, and Requirements_View_Info.

Picasso_Data_Request

A Picasso_Data_Request is a mechanism for the translators, analysis tools, and resolution tools to request CG_Objects, Type_Hierarchy_Data, and/or Requirements_View_Info from the Picasso software. The request may include any combination of these types of requests, which will result in a Join operation on the results.

Picasso_Report

This includes all the data which the Picasso software transmits directly to the user. This may include screen displays, error messages, notifications, warnings, beeps, etc.

Picasso_User_Request

A Picasso Request is data transmitted directly from the user to the Picasso software. It includes the user's location and one of the following: Launch_Tool_Request, Store_View_Request, Retrieve_CASE_Tool_View_Request, or Update_External_Tool_Configuration Request.

Problem_Description	This is a textual description of the problem which was found by the analysis tool.		
Problem_Type	This is an identifier which is used to determine the category of the problem. This may be as simple as conflict, incompleteness, or ambiguity. It could include more information such as importance and difficulty.		
Referent	A referent is a unique identifier which is used to distinguish one concept from other concepts of the same type.		
Relation	A Relation is a type of Non-Link Object that is used to describe the relationship between two concepts. The direction of the arrows on the connection indicate the direction of the relationship. A Relation may also include an optional negation.		
Requirements_View	Requirements_View contains a CASE_Tool_View and one or more CG_Object (s).		
Requirements_View_Data	Requirements_View_Data	Store	contains Requirements_View(s).
Requirements_View_Info	Requirements_View_Info	contains	general information about a Requirements_View. This includes the View_Name, User_Name of the author, Date_Time_Info, Version, and the Associated_CASE_Tool.
Requirements_View_Info_Query	The Requirements_View_Info_Query is a mechanism for translators, analysis tools, and resolution tools to request Requirements_View_Info from the Requirements_View Data Store. The tools may request objects by View_Name, Author, Date_Time_Info, Version, and Associated_CASE_Tool. In addition, any combination of the above will be accepted.		
Resolution_Data_Query	The Resolution_Data_Query is used to request metrics entry logs which were created by the problem resolution tools. The metrics tools may request entries by Entry_Name, Date_Time_Info, User_Name, Tool_Name, and Action_Taken. In addition, any combination of these keys is possible.		

Resolution_Log_Entry	Resolution_Log_Entries contain information about corrective actions taken by the problem Resolution Tools which will be stored in the Metrics_Log. It includes the problem resolution Tool_Name, and Action_Taken.
Resolution_Report	This includes all data transmitted from the Resolution Tools directly to the User. This may include screen displays, error messages, notifications, warnings, beeps, etc. The content of this data item will vary depending on which Resolution Tool is being used.
Resolution_Request	This includes all the data transmitted directly from the User to the Resolution Tools. The content of this data item will vary depending on which Resolution Tool is being used.
Resolution Tools	The software tools that are used by users to correct problems found during analysis, and to generate Resolution_Reports and Resolution_Log_Entries.
Retrieve_CASE_Tool_View_Request	A Retrieve_CASE_Tool_View_Request is a Picasso_User_Request which allows the user to request a pre-existing CASE_Tool_View with a CASE_Tool_View_Query.
Retrieve_CG_Object	A process which manages retrieving CG_Objects from the Requirements_View_Data store. The process retrieves a Requirements_View from the data store to Retrieve_CG_Objects and Requirement_View_Info. The process then send CG_Objects and Requirement_View_Info to the requesting external tool. The process also generates a Picasso_Report back to the user indicating the success or failure of the request.
Retrieve_Original_View	A process which manages retrieving a CASE_Tool_View from the Requirements_View_Data store. The process retrieves a Requirements_View from the data store to retrieve an original CASE_Tool_View and an Associated_CASE_Tool_Name. The process then sends Associated_CASE_Tool_Name to the process Manage_External_Tools to Launch_CASE_Tool. Once CASE Tool is launched, the process then sends a CASE_Tool_View to the launched CASE Tool. The

	process also generates a Picasso_Report back to the user indicating the success or failure of the request.
Store_Requirements_View	A process which retrieves a CASE_Tool_View from a CASE tool, and receives CG_Objects from the translator and stores a Requirements_View to the Requirements_View_Data store. The process also generates a Picasso_Report back to the user indicating the success or failure of the request.
Store_View_Request	Store_View_Request is a type of Picasso_User_Request which prepares the system for the receipt of a CASE_Tool_View and corresponding CG_Object View. It contains the Name of the view and the Author of the view as well as a View_Location field that describes to Picasso where to find the view. The Associated_CASE_Tool is also sent to Picasso which uses it to determine from which translator to fetch the CG_Object View.
Tool_Launch_Info	This includes all the information necessary to launch the external tool. This would most likely be a command line which is executed to launch the tool.
Tool_Name	This is the name of an external tool which may be a CASE Tool or some other tool used for Translation, Metrics_Reporting, Problem Analysis, or Problem Resolution.
Tool_Type	Specifies whether the tool is an analysis tool, resolution tool, metrics tool, or translator.
Translator_Report	This includes all information transmitted from the Translator directly to the User. The content of this data item will vary depending on which Translator is being used. This may include screen displays, error messages, notifications, warnings, beeps, etc.
Translator_Request	This includes all information transmitted directly from the User to the Translators. The content of this data item will vary depending on which Translator is being used.
Translators	The software tools that translate the Requirements_View created by a CASE tools into the conceptual graph representation of the view.

Type	A type is predefined category which is used to classify Non-Link Objects. Types have a hierarchical organization in which each type is a sub-category of some other more general type. A type includes a Type_Name, Date_Time_Info, and User_Name of the person who added it.
Type_Hierarchy_Data	Type_Hierarchy_Data includes a collection of Types and the Parent/Child Relationships between them.
Type_Hierarchy_Query	A Type_Hierarchy_Query is a mechanism which will be used by translators and analysis tools to retrieve type hierarchy information to process conceptual graphs. The basic query specifies a search based on conditions for Name, Date/Time, and who added the entry. Like other search conditions, these may be keyword, pattern matching, equality, inequality, or greater than or less than. A further extension to this query allows the request to fetch ancestors or descendants of all the entries retrieved by the basic query. The ancestor condition can be specified to fetch all ancestors, ancestors back to the nth generation, ancestors beyond the nth generation, ancestors at the nth generation. Likewise, for a descendant condition, all descendant , descendant forward to the nth generation, descendant beyond the nth generation, and descendant at the nth generation
Type_Name	This is an identifier which is used to classify the various Non-Link Objects into specific, predefined categories.
Type_Query	A Type_Query is a mechanism by which the tools may request Type_Hierarchy_Data by Type_Name, Date_Time_Info, or User_Name of the person who added it.
Update_Configuration_Data	A process which adds or removes an external tool from the External_Tool_Configuration data store. The process will also generate a Picasso_Report regarding the status of the request.
Update_Ext_Tool_Config_Request	This is a type of Picasso_User_Request by which the User may update the Data in the External_Tool_Configuration data store. The Update_External_Tool_Configuration_Request can be

of two forms. It can be an Add_Request which allows the user to add information about new tool to the configuration, or it can be a Delete_Request which removes information about a tool in the current configuration.

User_Command_Entry

This is a type of Metrics_Log_Entry which records each Picasso_User_Request received from the User. This includes User_Name, User Command, and User_Command_Results.

User_Command_Query

A User_Command_Query is a mechanism which allows Metrics Tools to request Command_Log_Entries from the Metrics_Log. The available fields in User_Command_Query are Date/Time, User_Name, and User_Command_Results. These fields correspond to the fields in the User_Command log entries. The fields in the request are conditions that must be satisfied by the User_Command log entries that are returned when this request is sent to Picasso.

User_Command_Results

This data item provides a text description of the results of Picasso_User_Requests indicating whether or not the request was success or unsuccessful. In addition, if request was unsuccessful, an explanation of why it failed (Error Code) should be included.

User_Location

This specifies the location of the user. Since the users are globally distributed, there must be some way to specify the user's location for reporting purposes.

User_Name

This is a unique identifier used to specify a particular user of the system.

Users

Individuals (e.g. end users, designers, customers, developers, etc.) who have a stake in ensuring that the software will meet its goals. They are the primary source of stimuli to the system.

Version

This is information about the version of the Requirements_View. As a view is modified during the project, new versions are created and stored in the Requirements_View Data Store.

View_Location

This provides the location of the CASE_Tool_View to Picasso. This is necessary for Picasso to be able to

read and store the CASE_Tool_View in the Requirements_View Data Store.

View_Name

This is a unique identifier associated with each Requirements_View.

6.2 Justification for Using Conceptual Graphs

This section has been included to answer any questions as to why conceptual graphs were chosen as the internal representation format for Picasso.

One of the primary concerns during requirements analysis is customer communication. It is a good practice to describe the system using things in the domain of the customer. When choosing a notation for specifying the software, it is advisable to use a notation with which the customer is familiar. This will reduce the overhead time associated with explaining the notation to the customer and the possibility of miscommunication between the developer and the customer. In the case of this project, the both the developer and customer are very familiar with the notation of conceptual graphs. Even though conceptual graphs are not being used as a requirements notation, it seemed logical to use the same principles when choosing an internal representation. Therefore, it was decided that conceptual graphs should be investigated for their use in this system.

While researching conceptual graphs, a paper [DEL92] written by our customer was discovered in which he had already described a methodology for using conceptual graphs for such an application. Unfortunately, the translation schemes were not completely specified and required further study. After a considerable amount of effort, it was determined that the principles in the paper were sound, but some of the translation schemes would not work for this project. Therefore, based on the principles found in the original paper, a new set of translation schemes was developed for this project. These translation tables are contained in the notes section of this document.

6.3 Justification for Users Communicating Directly with External Tools

This section has been included to answer any questions as to why the decision was made to have the users communicate directly with the external tools.

When first considering this issue, it seemed that having the user communicate with the external tools through Picasso was the best choice. This was based on the goal of presenting the user with a consistent user interface. A secondary benefit was to reduce duplication of code (GUI, routing) among the various external tools.

However, as the project progressed and the interface for the external tools became more solid, it became clear that there would have to be a trade-off between flexibility and the integration. In order to maintain a consistent user interface, every tool in each category would have to offer exactly the same options to the user. This would not be very flexible which was a primary goal for this system. Based on previous conversations with the customer, it was decided that the flexibility of the system more important than a consistent user interface or the reduction of duplicated code. As a result, the decision was made to have the users launch the external tools through Picasso, but communicate directly with the tool thereafter.

6.4 Conceptual Graph Translation Tables

Object-Oriented (Class Diagram)	Conceptual Graph Notation
Classes	Concept (Type = ENTITY; Referent = Identifier) Example: [Entity: Patient]
Attributes	Concept (Type = DATA; Referent = Identifier) Linked to Class by Relation (Type = ATTR) Example: [DATA: Patient] => (ATTR) => [CLASS:Vital Signs]
Procedures/Functions	Concept (Type = PROCESS; Referent = Identifier) Inputs Linked by Relation (Type = INPUT) Outputs Linked by Relation (Type = OUTPUT) Examples: [ENTITY: Patient] => (INPUT) => [PROCESS: Get Reading] [PROCESS: Get Reading] => (OUTPUT) => [DATA: Reading]
Inheritance	Relation (Type = INHERITS) Example: [ENTITY: OutPatient] => (INHERITS) => [ENTITY: Patient]
Aggregation	Relation (Type = CONTAINS) Example: [ENTITY: Car] => (CONTAINS) => [ENTITY: Engine]
Association	Relation (Type = Identifier) Example: [ENTITY: Owner] => (OWNS) => [ENTITY: Store]
Cardinality	Relation (Type = Identifier) Examples: [ENTITY: Car] => (1-TO-1) => [ENTITIY: Engine] [ENTITY: Engine] => (1-TO-N) => [ENTITIY: Cylinders]

Table 6-1 Translation Table Object-Oriented

Data-Oriented (Entity-Relation Diagram)	Conceptual Graph Notation
Entities	Concept (Type = ENTITY; Referent = Identifier) Example: [ENTITY: Patient]
Attributes	Concept (Type = DATA; Referent = Identifier) Linked to Entity by Relation (Type = ATTR) Example: [ENTITY: Patient] => (ATTR) => [DATA:Vital Signs]

Relationships	<p>Relation (Type = Identifier)</p> <p>Example: [ENTITY: Owner] => (OWNS) => [ENTITY: Store]</p>
Cardinality	<p>Relation (Type = Identifier)</p> <p>Examples:</p> <p>[ENTITY: Car] => (1-TO-1) => [ENTITY: Engine]</p> <p>[ENTITY: Engine] => (1-TO-N) => [ENTITY: Cylinders]</p>

Table 6-2 Translation Table Data-Oriented

Function-Oriented (Dataflow Diagram)	Conceptual Graph Notation
External Entities	Concept (Type = ENTITY) Example: [ENTITY: Patient]
Processes	Concept (Type = PROCESS; Referent = Identifier) Example: [PROCESS: Get Reading]
Data Flows	Concept (Type = DATA; Referent = Identifier) Input Data Flows Linked by Relation (Type = INPUT) Output Data Flows Linked by Relation (Type = OUTPUT) Example: [DATA: Vital Signs] => (INPUT) => [PROCESS: Get Reading] [PROCESS: Get Reading] => (OUTPUT) => [DATA: Reading]
Control Flows	Concept (Type = EVENT; Referent = Identifier) Input Data Flows Linked by Relation (Type = INPUT) Output Data Flows Linked by Relation (Type = OUTPUT) Example: [EVENT: Failure] => (INPUT) => [PROCESS: Get Reading] [PROCESS: Get Reading] => (OUTPUT) => [EVENT: Alarm]
Decomposition	Relation (Type = SubProcess) Example: [PROCESS: Eat] => (SUBPROCESS) => [PROCESS: Chew Food]

Table 6-3 Translation Table Function-Oriented (Dataflow)

State-Oriented (State Transition Diagram)	Conceptual Graph Notation
States	Concept (Type = STATE; Referent = Identifier) Example: [STATE: Idle]
Events	Concept (Type = EVENT; Referent = Identifier) Example: [EVENT: Out of Paper]
Outputs	Concept (Type = DATA; Referent = Identifier) Example: [DATA: Alarm]
Transitions	Demon (Type = TRANS) Example: [STATE: Running] => <<TRANS>> => [STATE: Repair]

	[EVENT: Error] ==> ==> [DATA: Alarm]
Start State	Demon (Type = START) Example: <<START>> ==> [STATE: Idle]

Table 6-4 Translation Table State-Oriented (STD)

State-Oriented (R-NETS)	Conceptual Graph Notation
R-Net Start	Demon (Type = START) Example: <<START>> => [DATA: From Device]
R-Net End	Demon (Type = END) Example: [DATA: From Device] => <<END>>
Action	Concept (Type = PROCESS; Referent = Identifier) Example: [PROCESS: Message Type?]
Stimulus	Concept (Type = DATA; Referent = Identifier) Linked to Action by Relation (Type = Input) Example: [DATA: From Device] => (INPUT) [PROCESS: Message Type?]
Result	Concept (Type = DATA; Referent = Identifier) Linked to Action by Relation (Type = Output) Example: [PROCESS: Message Type?] => (OUTPUT) => [DATA: Message Type]
And	Demon (Type = AND) Example: [Data: Message Type] => <<AND>> => [PROCESS: Store Data] => [PROCESS: Examine Data]
Or	Demon (Type = OR; Annotation = Decision Process) Example: [PROCESS: Range] => <<OR>> => [PROCESS: Tell Nurse] => [PROCESS: Null] => <<END>> Annotation: IF Range = Safe THEN Null ELSE Tell Nurse

Table 6-5 Translation Table State-Oriented (R-NETS)

6.5 Justification for Separate External Tools in the Context Level Data Flow Diagram

The customer has requested that the various external tools (Metrics, Analysis, Resolution, etc.) be consolidated into a single entity to avoid redundancy. This request was carefully considered and a discussion of the issues and the conclusions reached are presented in this section.

There are two primary advantages to combining the external tools together: simplification of the data flow diagrams and elimination of redundancy in the SRS. Currently, the context level data flow diagram shows all of the external tools. This results in a very busy and distracting diagram, reducing its effectiveness. This could be simplified by combining all of these tools together. In addition, the repetition of data common to all these tools could be avoided, making the SRS more concise.

While the diagrams would become more concise, it is uncertain whether this is desirable in this document. Since the external tools will be developed by groups other than the Picasso development team, the SRS will serve as a reference document for these tools. If the tools are combined together, it will be difficult to isolate the data flows associated with particular types of external tools. This could lead to confusion and a lack of interoperability with Picasso.

As to redundancy in the SRS, while there are similarities between external tools, there are also major differences. The metrics tools do not use conceptual graph objects or type hierarchy data. Both the analysis tools and resolutions tools use conceptual graph objects and type hierarchy data, but only the resolution tools can modify those types of data. There are also different metrics log entries associated with each tool. It would still be necessary to list all of these differences as well as others.

The SRS will serve as both a reference document for future development and a communication tool between the customer and the developer. As a reference tool, it makes sense to leave the various external tools separated, since the developers of the external tools only need to reference the data for their particular tool. Since customer communication does not appear to be a problem, the diagrams do not seem to be a major problem. Therefore, after careful consideration, the external tools will be left separated at this time.

6.6 Axiom CASE Tool

The Axiom CASE tools was very useful to our group in a number of ways. It supports a structured analysis method. It also uses Backus-Naus Format (BNF) to describe and create the data items. It provides an integrated drawing tool to create flow diagrams. It allows a P-specs, C-specs, and data items to be defined. It supported traceability by creating trace items and linking these trace items to P-specs, C-specs, and data items. It can validate child flow diagrams to parent process, providing a verification that all the inputs and outputs are used in child process. It is capable of validating P-specs, C-specs, and the data dictionary. It supplies a template that supports the DOD-2167A SRS standard. It can also generate an SRS directly to a printer or to a MS Word compatible file

There were a few disadvantages to the Axiom CASE tool. It did not support multiple users. Only one person could work on a particular project file at a time and there was no way to merge files from multiple users. Another problem was that there was a large amount of learning necessary before one could become proficient with the two. Only two members of our team learned to use the tool. Another problem was that the version available to us was a demo version and that the limitations of that demo version were not made clear to us when we received the tool. We discovered limitations in the number of processes we could enter into a data flow diagram well after we had started using the tool. This could have caused a large investment to become useless if we had needed to add more processes.

Overall, using Axiom benefited the team by organizing and validating the structure of our SRS and integrating text and data flow diagrams. We would recommend this tool to others.

Index

A

Action_Taken 40, 111
Actor 40, 111
Add_Request 40, 111
ambiguity 5, 111
Analysis tools 5, 112
Analysis_Data_Query 41, 111
Analysis_Log_Entry 10, 41, 111
Analysis_Report 41, 71, 111
Analysis_Request 42, 71, 112
Ancestor_Descendant_Query 112
Ancestor_Descendent_Query 42
Annotation 40, 42, 112
Associated_CASE_Tool 22, 39, 43, 112
Author 43, 112
Axiom CASE Tool 133

C

CASE tool 4
CASE Tools 112
CASE_Tool_Report 44, 71, 112
CASE_Tool_Request 44, 72, 112
CASE_Tool_View 9, 44, 112
CASE_Tool_View_Query 45, 113
CG_Object 31, 32, 45, 113
CG_Object_Name 45, 113
CG_Object_Query 31, 35, 46, 113
Command_Name 46, 113
Concept 47, 113
Conceptual Graph Objects 32
Conceptual Graph Translation Tables 127
Configuration_Data 22, 47, 113
Confirmation_Report 47, 113
Context 48, 114

D

Dashed_Directed_Arc 48, 114
Data Dictionary Hierarchy 71
Date_Time_Info 48, 114
Delete_Request 49, 114
Demon 49, 114
Directed_Arc 49, 114
Direction 50, 114

E

Entry_Name 50, 114
Error_Report 50, 114
External Interface Description 8
External_Tool_Configuration 50, 72, 114

I

incompleteness 5, 115
inconsistency 5, 115

L

Label 51, 115
Launch_Analysis_Tool 16, 51, 72, 115
Launch_CASE_Tool 16, 51, 72, 115
Launch_External_Tools 115
Launch_Input_Translator 52, 115
Launch_Metrics_Tool 16, 52, 72, 115
Launch_Output_Translator 53, 115
Launch_Resolution_Tool 16, 53, 72, 115
Launch_Tool_Request 22, 27, 53, 115
Launch_Translator 16, 54, 72, 116
Levels 54, 116
Line_Of_Identity 54, 116
Link_Object 55, 116

M

Manage_External_Tools 116
Manage_Metrics_Log 116
Manage_Reqs_Views 116
Manage_Type_Hierarchy 36, 116
Metrics tools 5, 117
Metrics_Data_Request 10, 41, 55, 72, 117
Metrics_Log 56, 73, 117
Metrics_Log_Entry 16, 38, 56, 117
Metrics_Report 56, 74, 117
Metrics_Request 57, 74, 117

N

Negation 57, 117
Non_Link_Object 57, 117

P

Parent_Child_Relation 57, 118
Picasso 3, 5, 6, 118
 components 4
 system 3
Picasso_Data 9, 10, 14, 16, 58, 118
Picasso_Data_Request 9, 10, 14, 58, 75, 118
Picasso_Report 15, 58, 76, 118
Picasso_User_Request 9, 59, 76, 118
Problem_Description 59, 119
Problem_Type 60, 119
Process Launch_External_Tools 26
Process Manage_Metrics_Log 37
Process Manage_Reqs_Views 30
Process Retrieve_CG_Object_Data 35
Process Update_Configuration_Data 25

R

Referent 60, 119
Relation 60, 119
Requirements_View 31, 32, 33, 34, 35, 60, 119
Requirements_View_Data 61, 77, 119

Requirements_View_Info..... 61, 119
 Requirements_View_Info_Query 62, 119
 Resolution tools5, 63, 120
Resolution_Data_Query 62, 119
Resolution_Log_Entry 10, 62, 120
Resolution_Report..... 63, 79, 120
Resolution_Request 63, 79, 120
 Retrieve_CASE_Tool_View_Request 31, 63, 120
 Retrieve_CG_Object.....120
 Retrieve_Original_View 34, 120

S

Store_Requirements_View121
Store_View_Request..... 22, 64, 121

T

Tool_Launch_Info 64, 121
Tool_Name 65, 121
Tool_Type..... 65, 121
Translator_Report..... 65, 79, 121
Translator_Request 66, 79, 121

Translators 5, 121
 Type.....122
Type_Hierarchy_Data36, 37, 66, 122
Type_Hierarchy_Query..... 36, 67, 122
Type_Name..... 67, 122
Type_Query..... 68, 122

U

Update_Configuration_Data122
Update_Ext_Tool_Config_Request..... 22, 68, 122
User_Command_Entry..... 68, 123
User_Command_Query..... 69, 123
User_Command_Results..... 69, 123
User_Location..... 69, 123
User_Name..... 70, 123
 Users 4, 123

V

Version..... 70, 123
View_Location 71, 123
View_Name..... 71, 124