1. Topic: C2 Decision Making & Cognitive Analysis

2. Title : **Re-Use of Integrated Dictionary Components for C4ISR Architectures**

3. Author: **Asma T. Ali**

4. Organization: C3I Center, George Mason University

5. Address: System Architectures Laboratory
C3I Center, MSN 4B5
George Mason University
Fairfax, VA 22030-4444

Asma T. Ali
703-993-1725 (v)
703-993-1706 (f)
aali3@gmu.edu

6. POC: Asma Ali

7. This is a student paper; Asma Ali is a first semester PhD. student

Re-Use of Integrated Dictionary Components for C4ISR Architectures[1]

Asma T. Ali
System Architectures Laboratory
C3I Center, MSN 4B5
George Mason University
Fairfax, VA 22030-4444
aali3@gmu.edu

ABSTRACT

The C4ISR Architecture Framework Products can be developed using mapping between Structured Analysis products and the Framework products and also based on mapping between Object Orientation and Framework products [Levis and Wagenhals, 2000 and Bienvenue, Shin and Levis, 2000]. Both of these methodologies for architecture design are adequate to obtain essential and supporting C4ISR products. However, sometimes the architect has to add new capabilities into the existing architecture that contains the products developed using either of the two approaches. If he uses the same approach (either Structured or Object Orientation) to develop the new set of products as was used for the original architecture, then the task of model concordance is not difficult, otherwise it is not easy. This paper discusses the reuse of the components of an Integrated Dictionary developed for the C4ISR products to add new products into the existing architecture. The C4ISR Architecture Framework products are developed using two approaches for a single operational concept, and then the contents of the two integrated dictionaries are compared to find out the similarities and differences.

1. INTRODUCTION

In a rapidly changing world of technology and increased uncertainties, the Department of Defense (DoD) faces an intense challenge to cope with the situation and the development of an interoperable information system. To handle the situation well, and achieve flexibility of interoperability in information systems, the DoD has provided standard for architecture specifications that directly support military operations. These specifications for architecting information systems are Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework Version 2.0 (CAF). The goal is to provide rules, guidance, and product description for developing and presenting architectures to ensure interoperable systems. Another objective is to develop a common unifying approach for different agencies to follow in developing their various architectures. The CAF prescribes four architecture views, the All View, Operational Architecture View, System Architecture View, and the Technical Architecture View. The products are designative by the initials of the view and a product number. For example, they are the AV-1 and AV-2 All View products, nine OV products, 13 SV products, and 2 TV products.

[1] This work was supported by the Office of Naval Research under grant No. N00014-00-1-0267

Although, the CAF provides common definitions, data and references, and describes a set of products to represent three views of an architecture, it does not provide any well defined, and widely accepted processes or guidance to produce those products. However, two approaches, one based on mapping between Structured Analysis (SA) products and the CAF products, and another based on mapping between Object Orientation (OO) and Framework products have been developed by Levis and Wagenhals, 2000 and Bienvenue, Shin and Levis, 2000, respectively. In the former approach, the CAF products are developed using tools and techniques of SA constructs, which identifies the interrelationships among the products. The latter approach demonstrates the development of CAF products using the OO methodology. Both approaches, if carried out properly, carry the same information. The main difference is the difference of focus. The Structured approach is focused on functions and data, while the Object Oriented approach is focused on entities and their interactions [Levis, A. H., Fall 2002].

In many agencies the architect using the CAF products has to deal with a legacy system that contains the products developed using either of the two approaches. When the architect is required to add new capabilities into an existing system, he has to develop new products consistent with the existing products. If the approach to be used in developing new set of products is same as used in the existing product, either SA or OO, then the task of model concordance is not very difficult. Whereas, if the architect has to use OO methodology for developing a new set of products, and the existing products were developed using Structured approach or vice versa, then the task of model concordance is not trivial.

The scope of this work is to make use of the Integrated Dictionary (which is one of the CAF products called "All View" –2(AV-2)) for developing CAF products using either SA or OO approaches. The Integrated Dictionary is an essential CAF product that provides a source for all the definitions for the graphical and tabular representations that comprise the products. The purpose is to find out the possibility of reusing these definitions associated with a set of diagrams developed using one approach (say SA) to develop another set of diagrams using the other approach (say OO). The task is accomplished by developing two sets of CAF products using SA and OO approaches for a single operational concept. The two integrated dictionaries thus developed are then compared to find out the similarities and differences in the definitions.

The remainder of this paper is divided into five sections. Section 2 presents a table and illustrations showing the mapping between CAF and the SA products and the CAF and OO products. The Unified Modeling Language (UML) specification is used for the OO approach. Section 3 illustrates and discusses the operational concept used to develop the products. Section 4 presents a table containing the definitions from the integrated dictionary and discusses the similarities and differences of definitions for the example problem, and Section 5 gives the summary of the work done.

2. Mapping Between CAF and Structured Analysis and Object Oriented Products:

The CAF Version 2.0 provides a guideline and a set of products, both essential and supporting, to represent an architecture. But the CAF does not specify a process for developing the architecture views and the associated products. These products are obtainable using SA and OO approaches [Levis, A. H., Fall 2002]. For both approaches the process begins with the creation of an operational concept. In the SA approach the operational concept guides the development of a functional decomposition, the physical architecture composed of system nodes and links, operational nodes and organizational models. The functional decomposition guides the development of the functional architecture [Levis and Wagenhals, 2000]. In Object Oriented approach, some of the CAF products are either essentially equivalent to the UML diagrams or are derivable from them, and, some are not derivable but, they require domain knowledge to complete [Levis, A. H., Fall 2002]. Framework uses graphical presentations, matrices and reports to develop architecture. This paper discusses only those products of the operational and systems architecture views that can be presented graphically. For example, Operational Node Connectivity Descriptions (OV-2), Activity Models (OV-5), Systems Interface Description (SV-1) etc. Table I gives a brief description of the mapping between CAF Operational Architecture view products and the two approaches. Table II lists CAF Systems Architecture view products. Columns 2 and 3 of both Tables I and II show mapping of CAF products with SA and OO approaches, respectively.

Table I: Mapping of C4ISR Operational Architecture View Products developed using Structured Analysis and Object Oriented Approaches

CAF Product	Mapping with Structured Approach	Mapping with Object Oriented Approach
Operational Concept (OV-1) diagram	Create a High level Operational Concept using domain knowledge	Not derivable from UML diagrams. It is developed directly from the domain knowledge base
OV-2 diagram, Operational Node Connectivity Description	Operational nodes are derived directly from Operational concept. Functional decomposition guides the development of needlines and operational activities	Derivable from the UML class diagram
OV-4, Organizational chart	Derived from Operational concept	Derived from Class/Object diagram

Table I (continued):

CAF Product	Mapping with Structured Approach	Mapping with Object Oriented Approach
OV-5, Activity Model	Functional decomposition guides the development of activity model. In its illustration of activity model the Framework uses IDEF0 as the modeling technique	UML activity diagram developed for operational and node classes can be used directly
OV-6a, Rule Model	Functional decomposition guides the development of Rule Model	Directly drivable from the State transition Diagrams for Operational nodes and element classes
OV-6b, State Transition Diagrams	Functional decomposition guides the development of the State Transition description. It is created in the form of State Transition Diagram	UML State Transition Diagram for each object can be used directly
OV-6C, Operational Event/Trace Description	This diagram has to be consistent with the OV-2 and OV-5 diagrams	UML Sequence diagram for operational nodes and element instances can be used directly.
OV-7, Logical Data Model	Derived directly from the Data Model of SA	May be derived from the Class Diagram

Table II: Mapping of C4ISR Systems Architecture View Products developed using Structured Analysis and Object Oriented Approaches

CAF Product	Mapping with Structured Approach	Mapping with Object Oriented Approach
SV-1, System interface diagram	System nodes and links are derived from operational concept	Derivable from the system Class diagram
SV-2, System Communication diagram	Derived from operational concept	Logically similar to SV-1 diagram but, at a lower level of detail.
SV-4, Systems Functionality Description	System entities and components are derived from operational concept and the activity model determines System Functionality description. Graphically it can be represented as activity model such as a data flow diagram	UML activity diagram developed for system node classes can be used directly.

3. OPERATIONAL CONCEPT FOR THE EXAMPLE PROBLEM:

The work done in this paper is based on a fictional FastPass system used at OilCo gas stations, and this system is conceptually based on the Mobil Corporation SpeedPass™ system. However, the fictional example used in this paper does not represent the actual Mobil SpeedPass system. Figure 1 is the OV-1 diagram or the graphical representation of the high-level operational concept. As shown in Figure 1, the driver having FastPass service will pull in front of a Self Serve fuel pump equipped with the FastPass system. If the driver has a FastPass tag, then he will wave the tag in front of the sensor in the pump.

The pump reads the driver's FastPass ID, and sends this ID through a wide area network (WAN) to the oil company's central office that has a database of driver information. The oil company retrieves the driver information and sends it to the financial institution responsible for issuing credit information to the driver through WAN. If the driver's credit account is valid, the financial institution approves the authorization and sends approval to the database office as *true*. In another case, if the driver's credit is not valid the financial institutions sends the approval to the central database office as *false*.

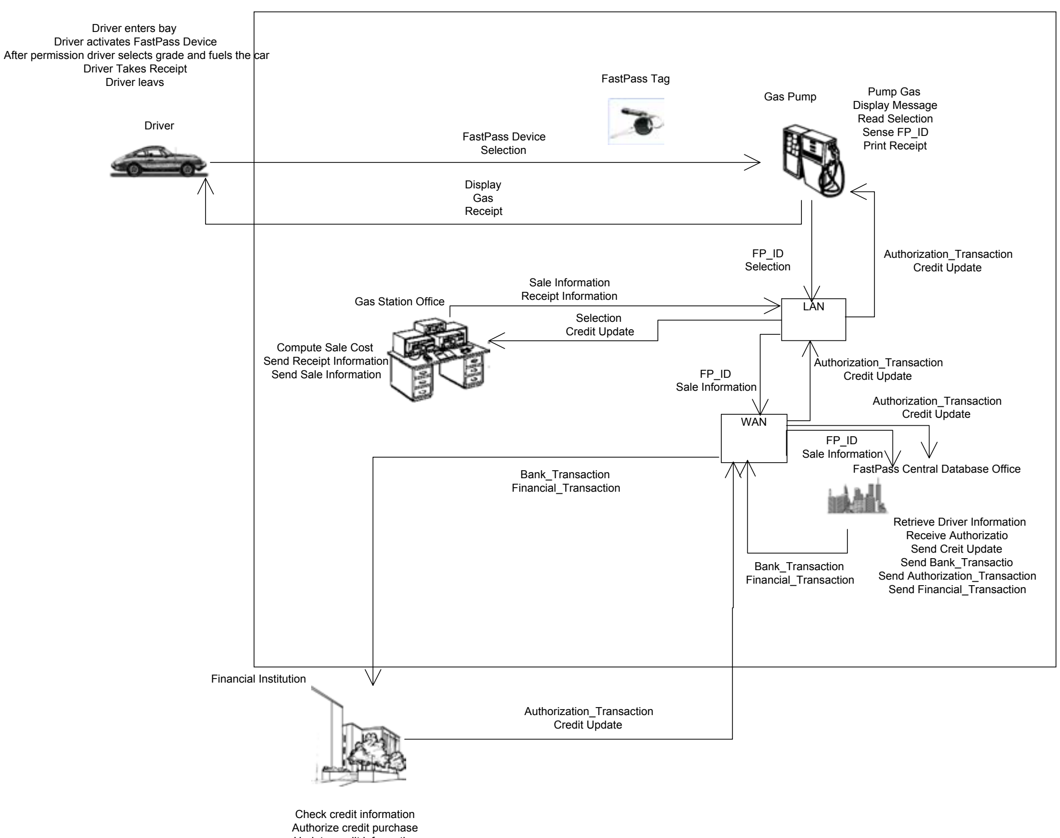

Figure 1. FastPass System Operational Concept

Upon receiving the authorization, the FastPass Central Database Office sends the authorization information to the pump. The pump determines the type of approval. If it is *true* the pump displays a message to the driver to select the gas grade and amount. If the approval is *false*, the pump displays a message to see the attendant and generates an error. Upon receiving the message from the pump for selecting the gas grade and amount, the driver makes a selection. The pump reads the selection and dispenses gas. After dispensing the gas the pump sends the sale information to the gas station office through a local area network (LAN) at the gas station. The gas station office calculates the cost and sends it to the central database office and the pump. The pump prints the receipt for the driver. The central database office sends this information to the financial institution that updates the driver's credit account and sends this information back to the central database office. The central database office updates driver's and gas station database and forwards the updated information to the gas station office, which updates its ledger account.

4. Definitions Of CAF Products for the Fastpass System Example

To examine the similarity and differences between CAF products that are developed using the two approaches, two architectures were created using the System Architect 2000 tool. Both were based on the same operational concept described in Section 3. One was done using the SA tool set and the other was done using the OO tools. In both cases, the System Architect 2000 tool created an Integrated Dictionary which contained the definitions of every element of every product in the architecture. These element definitions were then compared.

4.1 *Definitions of Operational Architecture view Products*

Tables III contains definitions of the CAF Operational Architecture views products developed using SA and OO approaches for the FastPass system example. These definitions are derived from the two Integrated data dictionaries developed during the process. Column 1 of the tables lists name of the CAF product developed. Since in the OO methodology some of the CAF products are derived directly from the UML diagrams, column 1 names those UML diagrams, too. For example, as mentioned in Table I, the UML Activity diagram can be directly used as an activity model, or OV-5 diagram. Column 1 in Table III lists the name of that product as "OV-5/UML Activity diagram". Column 2 of the table lists the definitions of the terms used in all products. For instance the "OV-2" diagram is consists of "Operational Nodes", "Needlines", "Information Exchange" etc., and they are listed in column 2. In case where UML diagrams are used directly for C4ISR products, column 2 shows the names of the terms used in the UML diagrams, too. For example, the "ICOMs" of the "OV-5" diagram map with the "Message Flows" between objects in the UML activity diagram. Column 2 shows these terms as ICOMs/Message Flows. Column 3 and 4 list terms of the CAF products when they are developed using SA and OO concepts for the FastPass System example.

Table III: CAF Product definitions for FastPass System Example

CAF Product	Definition	Definition of the CAF product developed using Structured Analysis approach for FastPass system	Definition of the CAF product developed using Object Oriented approach for FastPass system
Operational Node Connectivity Description (OV-2 diagram)	Operational Nodes	Driver	Driver
		Fast Pass Central Database	Fast Pass Central Database
		Financial Institution	Financial Institution
		Gas Station Office	Gas Station Office
		Pump	Pump
	Information Exchange	FastPass Device	FastPass Device
		Selection	Selection
		Display	Display
		Receipt	Receipt
		Receipt Information	Receipt Information
Command Relationships chart (OV-4 diagram)	Organizational Units	Driver	Driver
		Gas Station	Gas Station
		FastPass Central Office	FastPass Central Office
		Financial Institution	Financial Institution
Activity Model/UML Activity diagram (OV-5)	Operational Activities	Operate FastPass System	
		Validate Account	
		Operate Pump	
		Manage Sales	
		Present FastPass Tag	Present FastPass Tag
		See Display Message	See Display Message
		Select Gas Grade & Amount	Select Gas Grade & Amount
		Pump Gas	Pump Gas
		Take Receipt	Take Receipt
		Display Message	Display Message for Gas Selection
			Display Message to see attendant
		Sense FastPass	Sense FastPass
		Dispense Gas	Dispense Gas
		Print Receipt	Print Receipt

Table III (continued):

CAF Product	Definition	Definition of the CAF product developed using SA approach for FastPass system	Definition of the CAF product developed using OO approach for FastPass system
Activity Model/UML Activity diagram (OV-5)	ICOM/Message Flows	FastPass Device	FastPass Device
		Display	Display
		Receipt	Receipt
		Receipt Information	Receipt Information
		Driver Information	
		Gas Price	
			[Approval= True]
			[Approval=False]
Operational State Transitional Description / UML State Transition diagram (OV-6b)	Diagram State	Pump is Idle	Providing FastPass
		Validating Credit	Selecting Gas
		Dispensing Gas	Pumping Gas
		Computing Cost of Sale	Taking Receipt
		Printing Receipt	Pump is Idle
			Sensing FastPass
			Requesting Gas
			Dispensing Gas
			Providing Sale Information
			Printing Receipt
Operational Event/Trace Description/ UML Sequence diagram (OV-6C)	Nodes/Objects	Driver	Driver
		FastPass Central DatabaseOffice	FastPass Central Database Office
		Financial Institution	Financial Institution
		Gas-Station Office	Gas-Station Office
		Pump	Pump
	Object State	Present FastPass Tag	Present FastPass Tag
		See Display Message	See Display Message
		Select Gas Grade & Amount	Select Gas Grade & Amount
		Pump Gas	Pump Gas
		Take Receipt	Take Receipt

Table III (continued):

CAF Product	Definition	Definition of the CAF product developed using Structured Analysis approach for FastPass system	Definition of the CAF product developed using Object Oriented approach for FastPass system
Logical Data Model/UML Class diagram (OV-7)	Entities/ Association Classes	FastPass Device	FastPass Device
		Driver Database	Driver Database Credit Card database (modeled as aggregate classes for the class FastPass Central Database Office)
		Credit card database	
		Financial Transaction	Financial Transaction
		Authorization Transaction	Authorization Transaction
		Display	Display
		Selection	Selection
	Operational Node/ Classes	Driver	Driver
		Pump	Pump
		Gas Station Office	Gas Station Office
		Financial Institution	Financial Institution
		FastPass Central Database Office	FastPass Central Database Office
	Relationship	Defines	
		Included in	
		Required for	
		Triggers	
		Does	
		Used to compute	
		Leads to	
		Produces	

As shown in Table III, many definitions of the products developed using two different approaches match each other. The definitions for “Operational Nodes”, “Information Exchange”, “Organizational Units”, “Operational Activities”, “Object State”, and “Entities/Association Classes” map with each other. The reason is that the CAF products using both SA and the OO approaches were developed from the same operational concept. Figure 2 illustrates mapping between High Level Operational Concept, Operational Node Connectivity Description, and the UML Class diagram.

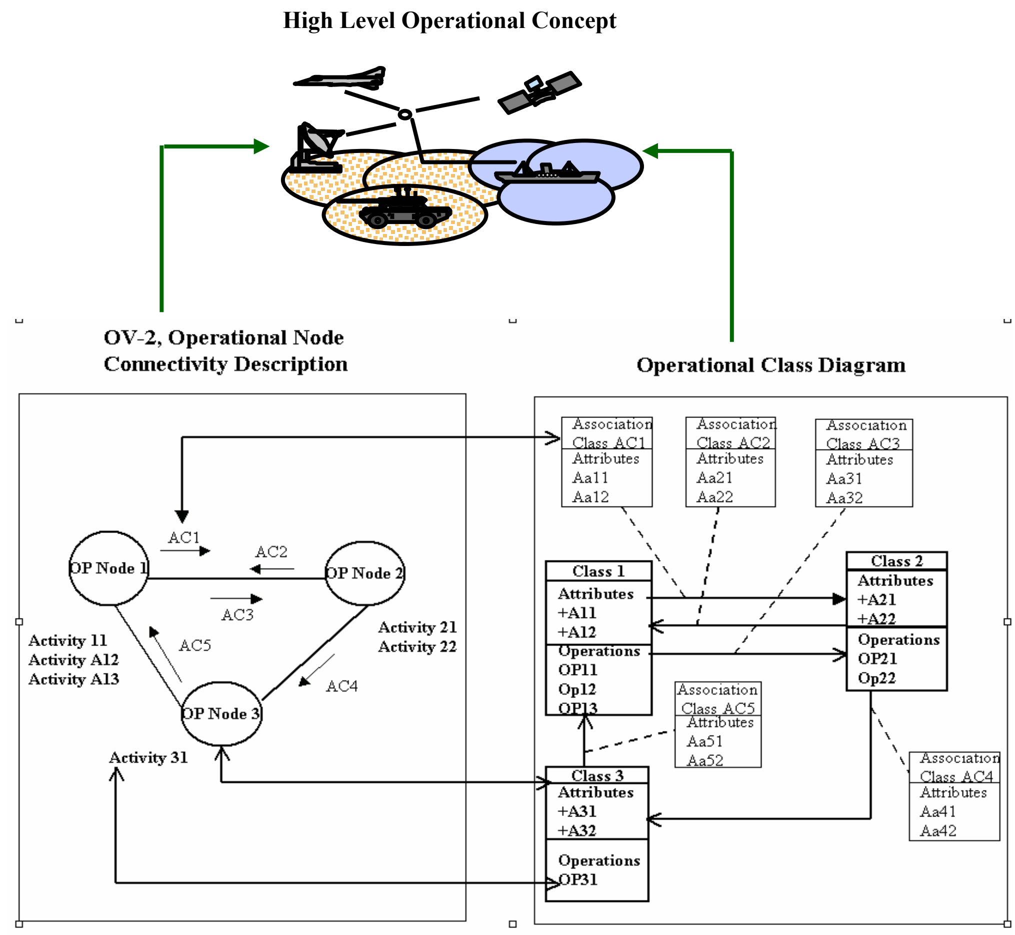

Figure 2: Mapping between Operational Concept, Operational Node Connectivity Description, and UML Class Diagram

As shown in Figure 2, the UML class diagram and the Operational Node Connectivity description are derived from the same operational concept. The "Classes" of the UML diagram, e.g. *Class 1, Class 2,* map with the "Operational Nodes", *OP Node 1*, *OP Node 2,* of the Operational Node Connectivity Description. The "Association Classes" *Association Class AC1, Association Class AC2* map with the "Information Exchange", and the "Operations" *OP11*, *OP12* of the classes map with the "Operational Activities" *Activity11, Activity 12* of the OV-2 diagram. Figure 3 illustrates the mapping between the Activity Model (OV-5) diagram, the UML Activity diagram and the Operational Node Connectivity Description. As shown in the figure the activities (operations) of the "Classes" in the UML Activity diagram, activities of the operational Nodes and the activities of the child diagram in the OV-5 match with each other. Similarly the "Message Flows" between the activities in the Class diagram map with the "Information Exchange", and the "ICOMs" of the activity model.

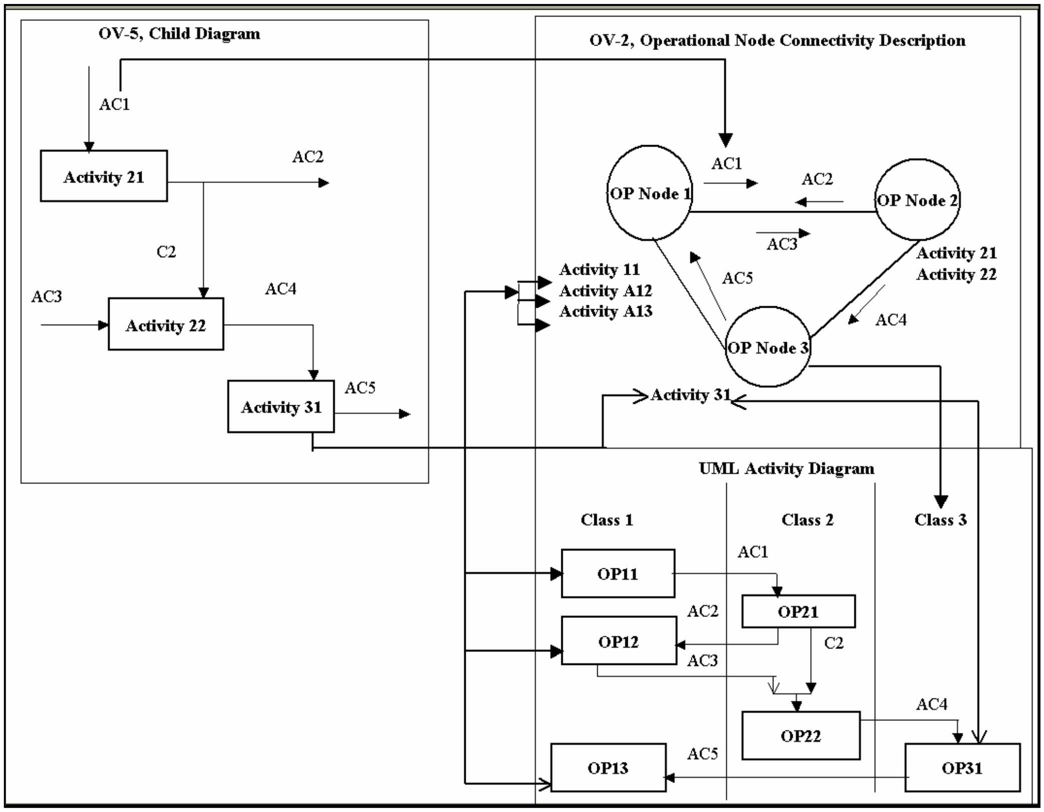

Figure 3: Mapping between Activity Model (OV-5) diagram, Operational Node Connectivity Description, and the UML Activity diagram.

The mapping illustrated in Figures 2 and 3 is evident from the definitions of the terms listed in Table III for FastPass example. The "Operational Nodes" and the "Classes" have the same definitions like, *Driver*, *Pump,* etc., Similarly, the "Information Exchange", "Message Flows", and the "ICOM"s have identical definitions as *FastPass Device*, *Selection*, and *Display*. The definitions of the "Operational Activities" for the OV-5 child diagram and activities (operations) of the "Classes" for the UML Activity diagram match with each other. For example, the activities *Present FastPass Tag*, *Pump gas,* etc. are identical across products.

In some cases there are certain definition that are either not present in one of the two dictionaries or they do not match. For the CAF OV-5 product "Activity Model/UML Activity diagram", the definitions of the "Operational Activities" in column 3 are *Operate FastPass System*, *Validate Account*, *Operate Pump*, *Manage Sales*, these terms do not match with any term in column 4 containing definitions for the OO approach. The reason is in the current approach for developing UML activity diagram the concept of hierarchy or functional decomposition is not used, and therefore, the activities (operations) of the classes are the same as the lower level activities in the activity model developed using the SA approach. Also, as shown in Table III, column 3 has the definition *Display Message*, whereas, column 4 has definitions *Display Message for gas selection* and *Display message to see attendant*. The difference is because in the UML activity diagram, the two display messages are modeled at the decision point, while, in the activity model developed using SA approach a decision point is not modeled, but rather the decision about *Display Message* is given in the rule model. The rule model for the activity *Display Message* states that if the approval for authorization is true, then the pump should display the message for gas selection, and if the approval is false then the pump should display message for seeing the attendant. This explanation is also valid for the ICOM definitions, [Approval= True], and[Approval= False] in column 4 of Table III. These two definitions are not present in the data dictionary for SA approach since the decision point in OV-5 diagram is not modeled, but at such point, the rule model explains the decision to be taken by the pump. Moreover, the ICOM definitions *Driver Information*, and *Gas Price* are not present in the OO data dictionary, because, these two definitions come as input to the activities from the data stores, and the UML activity diagram for Operational Nodes does not model the aggregate classes that behave as data stores.

In the SA approach, one State Transition diagram (OV-6b) is developed for the entire architecture, whereas, in OO methodology, state transition for each object/class is developed, separately. Since the approach used in both methodologies is different, the definitions used for the states in one data dictionary may also differ from the other. As shown in column 3 of Table III, the definitions for sates of the system *Pump is Idle*, *Validating Credit*, and *Dispensing gas* do not match totally with the definitions of the states for each object in column 4. For example in column 4 the definitions *Providing FastPass*, *Selecting Gas*, *Pumping Gas*, and *Taking Receipt* are various states of the class *Driver*.

In the OO approach, Class diagram for operational classes can be used directly as Logical Data Model (OV-7), whereas, in the SA approach, OV-7 can be created using the IDEF1X or Entity Relationship Diagram formalisms. As shown in Table III many definitions of the "Entities" map with the definitions of the "Association Classes" of the Class diagram like *FastPass Device*, and *Display*, *Selection*. Whereas, a few entities like *Driver Database* and *Credit Card Database* are modeled as aggregate classes in the UML class diagram, and they behave as a "data store" that contain information about the driver and his credit card. Also, as shown in Table III, column 3 has definitions for the "Relationship" between entities in the Logical Data Model, whereas, column 4 does not have such definitions because, in the UML Class diagram the relationships between the classes are not named.

4.2 *Definitions of Operational Architecture view Products*

Table IV contains definitions of the CAF System Architecture views products developed using both the SA and the OO approaches. Column 1 of the table lists names of the CAF products, column 2 of the table lists the definitions of the terms used in all products and columns 3 and 4 give the names of the definitions from the SA and the OO Integrated Dictionaries, respectively.

Table IV: CAF System Architecture View products definitions for FastPass System Example

CAF Product	Definition	Definition of the CAF product developed using Structured Analysis approach for FastPass system	Definition of the CAF product developed using Object Oriented approach for FastPass system
System Interface Description (SV-1)	System Node	Driver	Driver
		FastPass Central Database	FastPass Central Database
		Gas Station Office	Gas Station Office
		Financial Institution	Financial Institution
		Pump	Pump
	System Elements	Pump Control Unit	Pump Control Unit
		Gas Dispenser	Gas Dispenser
		Key Pad	Key Pad
		Sensor	Sensor
		Monitor	Monitor
		Printer	Printer
		Gas Station Control Unit	Gas Station Control Unit
		Ledger	Ledger
		Gas Price	Gas Price
		Calculator	Calculator
		FastPass Database Control Unit	FastPass Database Control Unit
		Driver Database	Driver Database
		Financial Institution Control Unit	Financial Institution Control Unit
		Credit Card Database	Credit Card Database
	System Data Exchange	FastPass Device	FastPass Device
		Display	Display
		Receipt	Receipt
		FastPass_ID	FastPass_ID
		Authorization Transaction	Authorization Transaction
		Dispensed Gas Data	Dispensed Gas Data
		Selection	Selection
			[Approval=True]
			[Approval=False]

Table IV (continued):

CAF Product	Definition	Definition of the CAF product developed using Structured Analysis approach for FastPass system	Definition of the CAF product developed using Object Oriented approach for FastPass system
Systems Communication Description (SV-2)	Communication Nodes	WAN	WAN
		LAN	LAN
		Microwave	Microwave
		Pump communication unit	Pump communication unit
		Gas Station Communication unit	Gas Station Communication unit
Systems Functionality Description (SV-4)	System Functions	Sense FastPass	Sense FastPass
		Display Message	Display Message for gas selection
			Display Message to see attendant
		Read Grade	Read Grade
		Read Amount	Read Amount
		Print Receipt	Print Receipt
		Compute cost of sale	Compute cost of sale
		Update Account	Update Account
		Retrieve driver Information	Retrieve driver Information
		Receive Authorization	Receive Authorization
		Request charge	Request charge
		Receive credit update	Receive credit update
			Present FastPass Tag
			Select gas grade and amount
			Pump gas
			Take receipt
			Validate account
			Update Credit
	Data Store/Aggregate Classes	Driver Database	Driver Database
		Gas Price	Gas Price

As listed in Table IV, the definitions of the "System Nodes", "System Elements", and "System Data Exchange" in both approaches match with each other. The definitions of the "System Nodes" are the same as the definitions of the "Operational Nodes". In the OO approach, the System Interface Description is derived from the Systems Class diagram. The mapping between the two diagrams is shown in Figure 4.

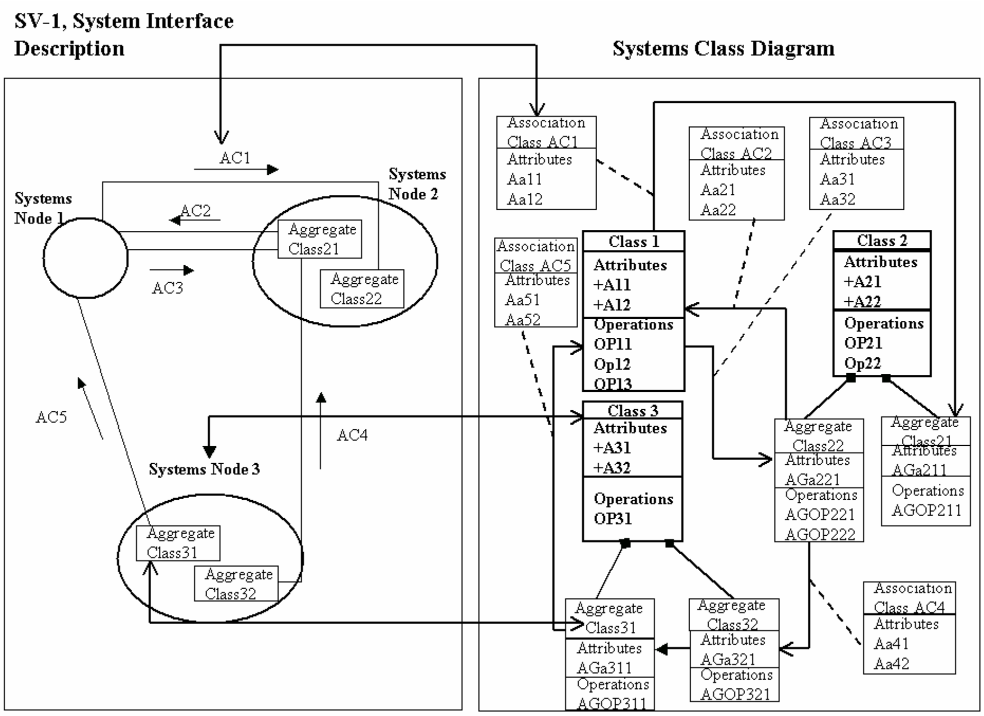

Figure 4: Mapping between System Interface Description (SV-2) and UML Class diagram for Systems Classes

As shown in Figure 4, the "System Classes" *Class 1*, *Class 2*, etc. map with the "System Nodes" *Systems Node 1*, *Systems Node 2*,, etc., the "Aggregate Classes" *Aggregate Class31*, *Aggregate Class32* map with the "System Elements", and the "Association Classes" match the "System Data Exchange". For the FastPass example, *Printer*, *Pump Control Unit*, and *Gas Price* are the "Aggregate Classes" for the classes *Pump*, and *Gas Station Office* respectively in the UML Class diagram for the System Classes. These terms match with the "System Elements" for the "System Nodes" *Pump* and *Gas Station Office*. The definitions for the "Communication Nodes" in both SA and OO dictionaries are the same.

In the SA approach the System Functionality Description (SV-4) is illustrated using either a Data Flow diagram or an Activity Model. In the OO methodology, UML Activity diagram can be used directly as SV-4 diagram. For this paper the author has used Data Flow diagram to model the Systems Functionality Description in the SA approach. Figure 5 shows the mapping between the UML Activity diagram and the Data Flow diagram.

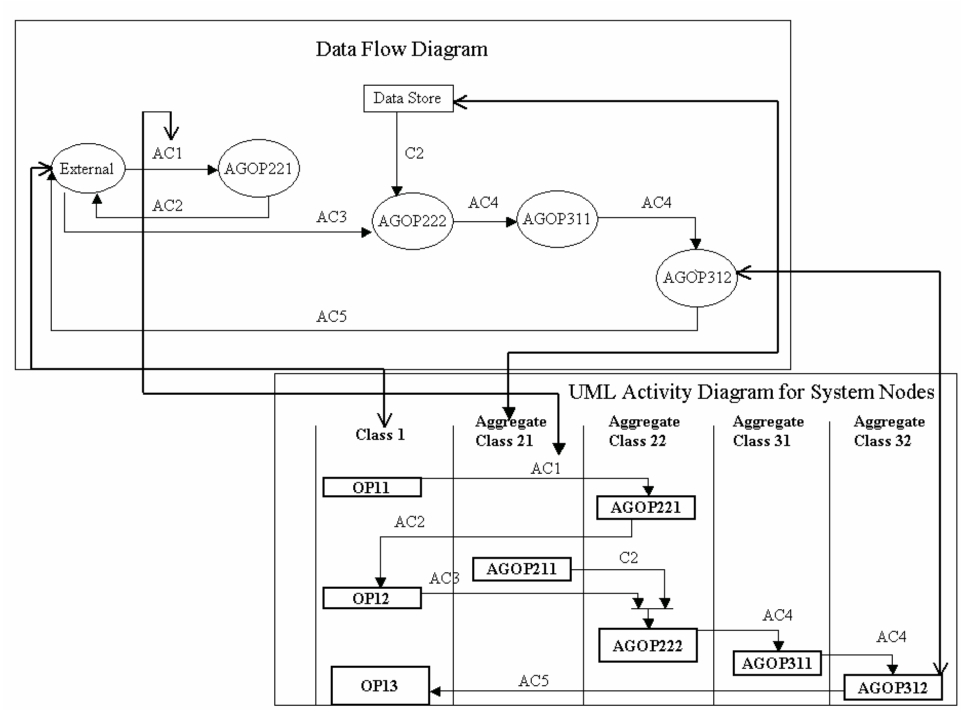

Figure 5: Mapping between UML Activity diagram and the Data Flow diagram

The mapping shown in the Figure 5 is also evident in the definitions of the terms listed in Table IV. The activities (operations) of the classes map with the "System Functions" such as *Sense FastPass*, *Read Grade*, and *Read Amount*. The definitions *Present FastPass Tag*, *Select Gas Grade and Amount*, and *Take Receipt* are functions of the *Driver* Class/System node that is external to the system. The data flow diagram models the external system node but not its functions, and therefore, the definitions of functions of the node external to the system are not presenting SA dictionary. Figure 5 also shows the mapping between the "Data Store" and the Aggregate Classes. In FastPass system, the definitions of the "Data Store" map with the definitions of the "Aggregate Classes" such as *Gas Price* and *Driver Database*.

When C4ISR products were developed for FastPass system using SA and OO approaches, the two data dictionaries contained numerous definitions for Operational Architecture and System Architecture view products. All these definitions are not listed in Tables III and IV. However, sufficient definitions have been listed to show the similarities and the differences between these definitions.

5. Summary

This paper discussed the reuse of definitions contained by an Integrated Dictionary of the CAF products. FastPass system example is used to develop CAF products using Structured Analysis and Object Oriented approaches. The components of the two Integrated Dictionaries were compared to find out the similarities and the differences among the terms used. The contents of the two Integrated Dictionaries show that most of the terms are identical, and they can be reused when the products are to be developed using either of the two approaches. However, there are certain differences among those terms that occur because of the difference in the two architecture development techniques. Thus, an architect will have to use experience and domain knowledge to "fill in the blanks" when re-using products from one architecture in another.

ACKNOWLEDGMENT

The support and guidance provided by Dr. Lee W. Wagenhals, Research Associate Professor, at George Mason University are greatly appreciated.

REFRENCES

1. Levis Alexander H., and Wagenhals Lee, W., C4ISR Architectures: I. Developing a Process for C4ISR Architecture Design, Systems Engineering 3, 2000.

2. Bienvenu Michael, P., Shin Insub, and Levis Alexander H, C4ISR Architectures: III An Object oriented Approach for Architecture Design, Systems Engineering 3, 2000.

3. C4ISR Architecture Framework Version 2.0, C4ISR Architecture Working Group, Department of Defense, Washington, DC, December 18, 1997.

4. Levis, Alexander H., Lecture Notes on Advances in Architecture (Rev. Feb. 2002), Spring 2002, http://viking.gmu.edu/http/504g/index.html

5. Levis, Alexander H., Lecture Notes on C4ISR Architecture Framework Implementation (Rev. Sep. 2002), Fall 2002, http://viking.gmu.edu/http/syst621f01/Syst621.htm

6. Wagenhals Lee W., Haider Sajjad, and Levis Alexander H., Synthesizing Executable Models of Object Oriented C4ISR Architectures, submitted to Journal of Systems Engineering.