

A Feasible Program Organizational Architecture Framework

Adel Alblawi, Jerrell Stracener, and Jeffery Williams

Abstract—A feasible program schedule and personnel allocation model may contribute to identifying the final shape of a program organization. If a program organizational design can be shown to be constructed such that it will satisfy the requirements without violating the constraints, then the design may be considered a feasible solution. This paper targets program organizations that are capable of designing and developing complex products. The approach in this paper builds upon the results of the Program Organizational Architecture Framework (POAF). We utilize constraint programming in order to extend the architecture framework and find a feasible design. Important decision variables and constraints are considered in this study. A hypothetical scenario of an ongoing program is presented to demonstrate that a given program organizational design is a feasible solution. This approach enables the program designers to support the decision-making process of implementing an effective program organizational design to manage a complex system and select the "best" program organizational structure.

Index Terms—Constraint programming, complex products, systems engineering management, organizational development.

I. INTRODUCTION

Defining the right program organizational structure to execute the objectives of a complex program is becoming crucial in today's business environment. Leaders cannot only rely on a simple structure or good people to accomplish the work. A system has to be built to support the people to work more effectively and aid the program organization in achieving superior performance [1]. Program is defined as the management and the coordination of multiple projects as one unit to achieving defined goals. For example, a complex program could be the design and development of an aircraft. Organization design is a process of developing the program organization and identifying its structure. The increasing need for organizational design is due to the accelerated changes in stakeholders' requirements, the myriad information that exchanged between teams. Also, the growing size of the program organization or some of the products being produced. Figure 1 depicts various organizations joint as one unit to define a program organization and play some roles in the program organizational structure.

Many factors contribute to the selection of program organizational structure and affect the program organization

performance including strategy, goals, size, and environment. Solving a problem that includes existing factors can be a complicated or time-consuming process. Alternatively, the design problem should focus on candidate variables that can be deterministic and realistic in order to provide a feasible solution space. These variables are often derived from the stated objectives in a program mission statement or to fulfill a stakeholders' needs. The primary objective of organizational design is the construction and the allocation of personnel resources to attain multiple objectives [2]. Therefore, the program organizational design should be based on the definition of required capabilities, personnel capabilities, and personnel ability to flow during the execution of a program. On the other hand, defining the right mix of required personnel, who are responsible for performing activities, and personnel resources characteristics may influence the choice of the program structure [3].

The construction of the program organization and its constituents are developed through the use of an architecture framework. An architecture framework provides the rules for producing an architectural description (or design in this case) of a complex system. A quantitative approach has been developed to evaluate the architectural description and to evaluate whether such a program organizational design is a feasible solution.

A. Related Work

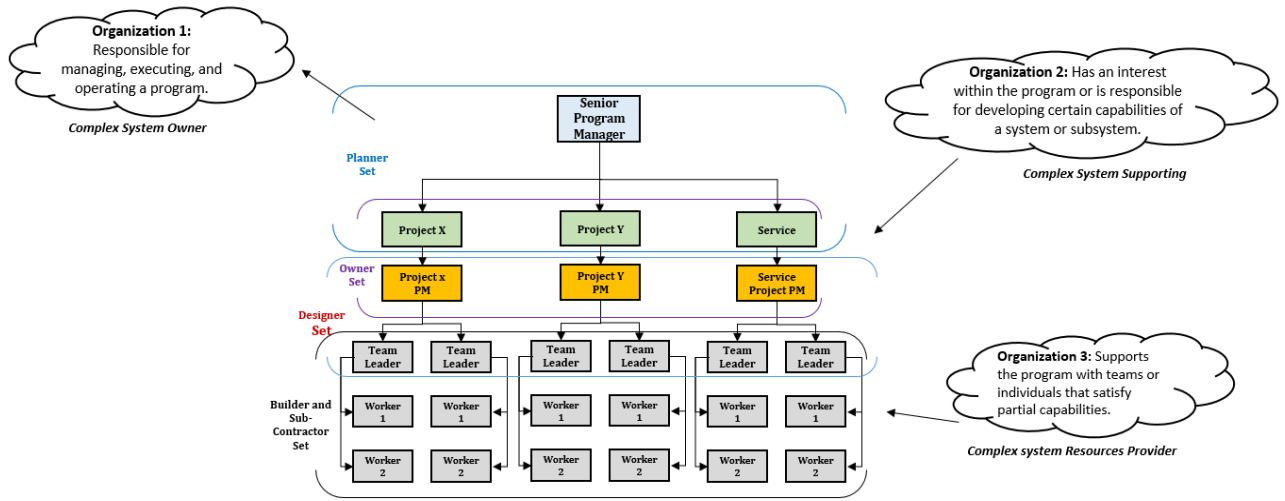
Organization design has been studied in the literature as a decision-making problem. It means the process of designing a program organization involves the decisions to be made and the constraints limiting these decisions. As a result, an organization design problem can be formulated as a mathematical problem that may solve different decision problems. For example, Obel represented an organization design as a linear programming problem that could be used to analyze the effect of resource allocation on different organization structure forms [4]. Arrow and Radner have studied the effect of different structures, information inputs, and allocation processes as a resources allocation mathematical model [5]. Therefore, the assumptions made in these models are fixed decisions and didn't evaluate the dynamics associated with organization design over time.

On the other hand, information flow within an organization's entities can impose an ample set of constraints on the organization design. Also, some of these constraints are very complex and difficult to model. Consequently, constraint programming has been widely used to model different decision-making problems. For example, Bourdais et al. have utilized constraint programming to make decisions about health care staff planning [6], and Henz was able to formulate an efficient round-robin college basketball schedule as a constraint program model [7].

Manuscript received December 1, 2015 ; revised March 10, 2016.

Adel Alblawi and Jerrell Stracener are with Southern Methodist University (SMU), USA (e-mail: aalblawi@smu.edu, jerrell@engr.smu.edu).

Jeffery Williams is with Securaplane Technologies, USA.



Program Organizational Structure

Fig. 1. An example of a program organization.

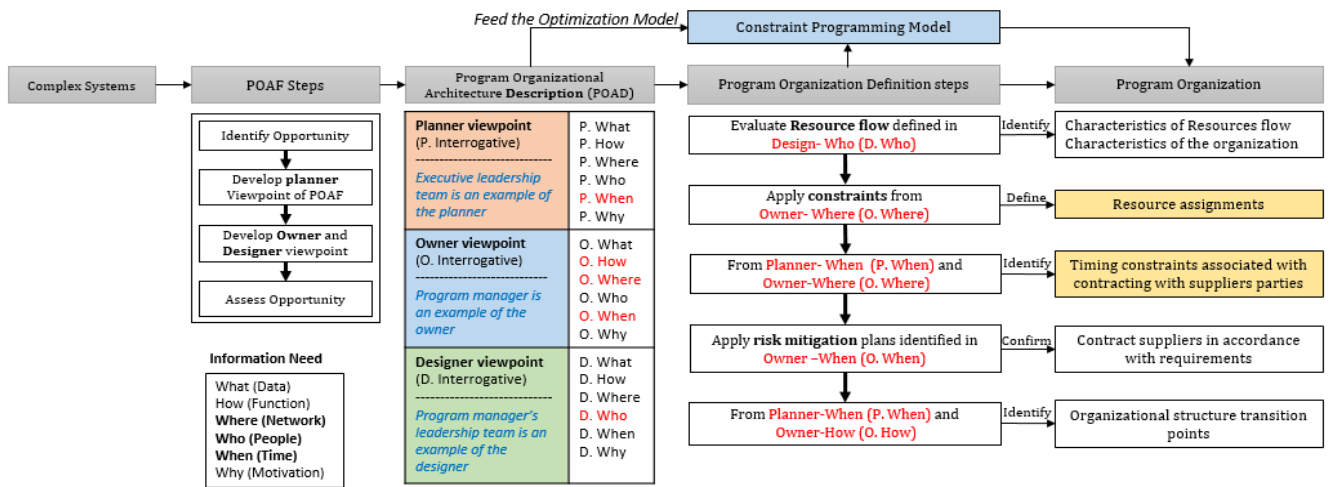


Fig. 2. Program organization definition process-adapted from [8].

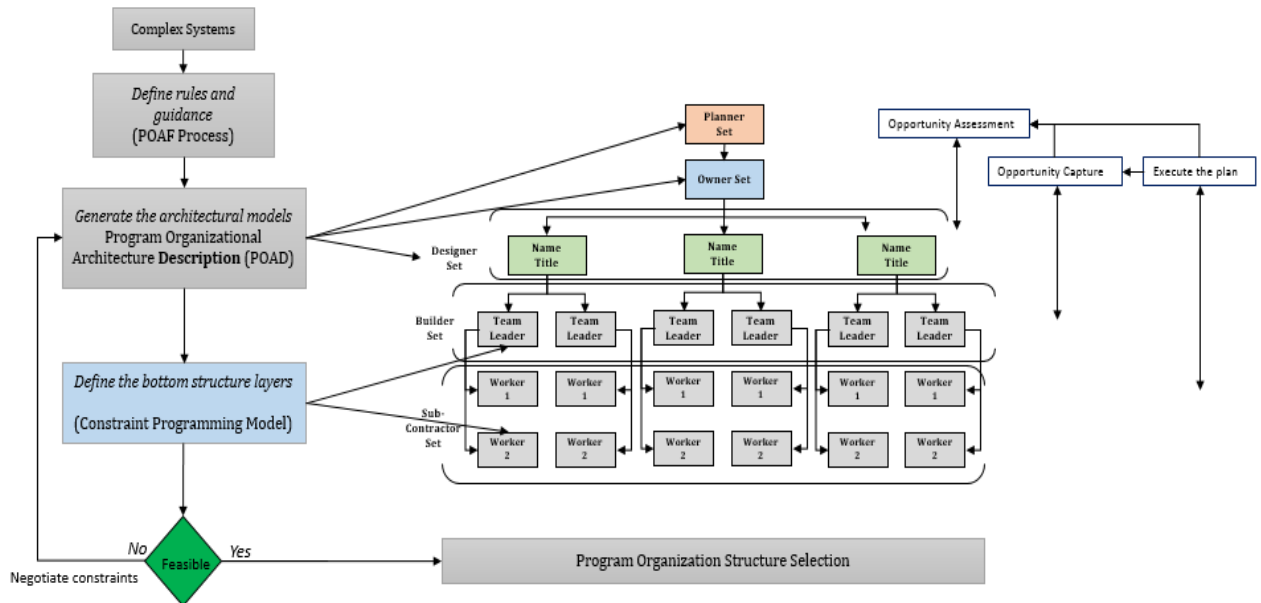


Fig. 3. A generic program organizational structure are matched to POAF major steps.

B. Motivation

The motivation for this paper is in response to an ongoing effort to investigate whether the design of a program

organization created for the purpose of designing and developing a product can be optimized. In order for a design to be optimized, a mathematical approach needs to be established to generate a feasible solution. The feasible

solution of a program organization relies on design description (e.g., program constraints) and the information available in POAF. POAF helps to capture and define the information needed by different stakeholders around the interrogatives who, where, when, how, and what in order to accomplish a set the objectives. It also contributes to bridging the gap between the problem domain and the solution space by selecting the right criteria and evaluating the possible alternatives. We present a systematic approach in order to find a feasible solution for a program organizational design problem through the application of the constraint programming technique. This technique can solve and handle complex types of constraints without requiring an objective function. CP finds the solutions that rapidly support the decision-making by reducing the number available choices and continuing the improvement process so a better solution can be found.

The paper is organized as follows. Section II gives in detail the two level models and the discussion of the CP model variables and constraints. We present a hypothetical program to evaluate its effectiveness in Section III. Finally, we discuss the conclusion and future work in Section V.

II. DESCRIPTION OF THE PROBLEM SPACE

A. What Is the POAF?

POAF is an architecture framework created for the design of a program organization. Alblawi et al. define the program organization as *“an organization that is designed for a specific purpose. These organizations are typically contained within a larger enterprise that has both tactical (short-term) and strategic (long-term) goals of which only a subset is addressed by the program’s organization”* [8]. POAF views the program organization as an open social system and defines the required activities (i.e., capabilities) to match the personnel’s capabilities over the resource flow timeframe. The framework produces 18 viewpoint models including the resource analysis workbook and integrated master schedule. These architectural artifacts collect the information needed to address the interrogatives taken from the Zachman Framework and construct customized POAF viewpoint models from DoDAF models. For instance, O.Why would refer to the Owner’s viewpoint and the model for the interrogative “Why”. Figure 2 depicts an overview of a program organization definition process.

B. Problem Statement

For a generic program organization, POAF is able to divide the structure into a 5 layered structure as shown in Figure 3. These layers represent a set of stakeholders with some concerns. For instance, the first layer represents the planner viewpoint and may consist of the executive leaders and the program managers for a program. The planner is concerned with the program scope and its impact on a bigger enterprise. The teams and individuals who are responsible for the development of the products within the program constraints are defined on the fourth and fifth layers: “Builder and Sub-Contractor Sets.” The structure of the bottom two layers changes frequently as a result of unforeseen events such as replacing individuals. These layers are determined by the description of the work to be performed, resource requirements, and the resource loaded Integrated Master

Schedule (IMS). Then, constraint programming techniques are employed to evaluate the feasibility of the constraints that imposed on the Program Organizational Architecture Description (POAD) models.

The process model of a defense system life cycle as well as commercial (with respect to their complexity) consists of decision points with multiple gating milestones. The target of this work may be applied in any phase. However, this paper focuses on the design and development phase of a complex system and its internal sub-phases as a result of reengineering activity.

C. Program Organizational Definition

POAD is made up of a collection of architectural artifacts. These artifacts, along with their relationships to one another, may be used to design a program organization. The artifacts are collectively documented as an architectural description (i.e., POAD) to address all known constraints. However, POAD doesn’t result in the final definition of the program organization. The information available in POAD must be evaluated to determine the final definition of the program organization. The process are showing in Figure 2 under the program organization definition step.

The information needed to formulate the decision problem is integrated and organized as a big model in blocks 2 and 3. Therefore, this paper investigates the appropriate assignment of personnel resources to program activities, which defines a feasible program schedule. Some authors view this problem as a Resource-Constrained Project Scheduling Problem (RCPSP) [9]. Personnel resources are one of the most important components of an organization that drives an organization to succeed or fail. In addition, the resultant architectural description is based on identifying the required capabilities, constructing the program activities, and then assigning the needed personnel to perform these activities. As previously stated, determining the number of personnel would influence the choice of the final organizational structure. The rest of the paper explains in detail the variables and methodology needed to create a feasible program organizational design.

III. CONSTRAINT PROGRAMMING MODEL

Constraint programming is a platform which handles computational systems that can be described mathematically. Constraint programming is best suitable for solving problems whose variables are restricted by values derived from requirements [10]; for example, the personnel characteristics are derived from the assessment of an organization internal capabilities or historical data. A constraint programming model can be presented as a Constraint Satisfaction Problem (CSP). The general approach to CSP begins with the decision variables and constraint identification, space search reduction, and the utilization of modeling language needed to find a feasible solution. A feasible solution of CSP requires the assignment of each decision variable value from its domain and the satisfaction of each constraint in the problem. The problem input data structure has an enormous impact on of how fast the model can be solved [11]. If a user organize the data in an efficient way, the time a solver takes to find a solution will be fast. A CSP is define as triple $\{X, D, C\}$ where X represents a set of variables $X = \{X_1, \dots, X_n\}$ that

can take a value from its domain $D = \{D_1, \dots, D_n\}$ such as $X_i \in D_i$ with respect to a set of constraints $C = \{C_1, \dots, C_m\}$ that restricting the assignment of values that variables can take. The definition of the constraint is the cross product between $D_i \times D_j \dots$ where C_{ij} is the possible constraint if $C_{ij} \subseteq D_i \times D_j$ [12]. For example, in a scheduling problem starting time of activity i represents a variable (X_i) for $i = 1, 2, 3$ that can take any values in the time horizon domain $= [0, 12]$ of a project with no precedence constraint is violated between X_1 and X_2 .

A. Program Organizational Decision Variable Selections

For program activities and schedule design, there are a variety of decision variable selections that can be defined. However, this selection is strategically associated with the stakeholder's preferences and the available information needed to formulate the model as shown in Table 1. The information is deterministic with well-known characteristics. Decision variables could be the activity start, end, and duration and has a domain (i.e., the set of all possible values that can be assigned to a variable).

TABLE I: LIST OF DECISION VARIABLE — ADAPTED FROM [8]

Decision Variable (X_i)	Variable Domain (D_i)	Description	POAD Input Models
Activity: Start (A) End (A) Duration (A)	Earliest start time and the latest start time or time-windows and duration of the activity	They represent the start time, end time, and duration of an activity. They identify the position of an activity in time. Duration within two activities is $Duration(A) = End(A) - Start(A)$	O. When D. How O. Who Historical Data
Personnel Resource: Resources (A_i, R)	$Res(A_i, R) \in [0, 1, \dots, A_i]$	This variable represents the candidate personnel resources assigned to compatible activities.	P. When O. When D. Where D. Who
Domain Filtering: Alternative (A_i, R_i)	$[R_1, \dots, R_i]$ Processing (A_i)	It represents the type of assigned personnel resource (internal, external, or contracted) with matched personnel skills to execute program activities	O. Who O. When

B. Program Organizational Design Constrains

Resource-Constrained Project Scheduling Problems have been studied in the literature under different constraints to solve a real problem. This led to the development of a constraints library to model various situations. In the problem addressed here, the modeled constraints are based on the program organization needed to define the team configurations and on assigning them to activities while considering the timing and skill constraints needed to accomplish activities. Table 2 shows a list of the program organization constraints.

C. Methodological Assumptions

The focus of this paper is on defining a feasible solution for a program organizational design by evaluating the information and constraints available in POAD models. It is assumed that POAF defines the constraints related to the schedule and personnel assignments. It is also assumed that the personnel needed to execute a set of activities are already trained and ready to join the program. Finally, the activity

duration, precedence, and personnel needed are based on the historical data or similar activities from previous experience.

TABLE II: PROGRAM ORGANIZATION CONSTRAINTS — ADAPTED FROM [8]

Constraint (C_i)	Description	POAD Input Models
<i>Precedence Relation</i>	These constrains model the relations between activities $end(A_i) \leq start(A_{i+1})$	D. How D. When
<i>Alternative</i>	This constraints ensures an activity is executed by using a capable personnel resources.	P. Who
<i>Personnel Resources Usage</i>	This constraint ensures the required number of resources doesn't exceed the available resources. $\sum_{A=1}^n Resource(A_i, R) \leq Available(R)$	P. Where P. Who
<i>Personnel Resources Skills</i>	This constraint ensures the activities must be fulfilled by personnel with the needed skills and experience for the work they will be doing.	P. Where P. Who O. When
<i>Personnel Resources</i>	This constraint ensures the assigned personnel can execute only one activity at a time. It also considers the worker transition time.	
<i>Program Life Cycle Phases</i>	This constraint ensures activities within each phase of the system life cycle are executed before the proceeding to the next phase.	P. Where P. When

IV. H A HYPOTHETICAL SCENARIO

The application of POAF and its extended constraint programming model is demonstrated through a use of a hypothetical program example. In the example, a program suffers from a communication breakdown between its organizations which leads to a need for a reengineering activity. This activity forces a designer (e.g., architect) to revisit the program process models and personnel resources to reshape its final organizational structure. In this example, we are interested in defining the program organization and recommend the "best" structure for the design and development phase.

A. Program Organizational Architecture Description

POAD is a result of POAF which contains all the information related to the program organization description. POAD has six models per viewpoint. Stakeholders' views are grouped into three viewpoints: Planner, Owner, and Designer as shown in Figure 2. For the example program described in this paper, the model D.Where identifies the constraints associated with the assignment of resources to activities. Also, the model O.Where identifies the timing constraints associated with the constructing of these resources. P.When decomposes the program in this example into three sub-phases: (1) concept development, (2) requirement analysis, (3) preliminary design, and (4) detailed design phase. The major activities involved in executing these phases are developed in O.How and D.How. For lack of space, we present some of the POAD models that relate to the constraint satisfaction problem formulation process. Figure 4 shows some of POAD models that constructed for this example. Finally, SysML is the standardized modeling language utilized in the example to generate the POAD models

TABLE III: PROGRAM ACTIVITY CHARACTERISTICS

Activity	Duration	Required Capability	Minimum Skill Level	Maximum Skill level	Resources Quantity
1	5	SE, SOFE, ME, EE	(1,1, 1, 1)	(5,5, 5, 5)	(1,1,1,1)
2	5	SE, SOFE, ME, EE	(2,1, 1, 1)	(5,5, 5, 5)	(1,1,1,1)
3	10	SE, SOFE, ME, EE	(2,1, 1, 1)	(5,5, 5, 5)	(1,1,1,1)
4	10	SE, SOFE, ME, EE	(2,1, 1, 1)	(5,5, 5, 5)	(1,1,1,1)
5	5	SE, SOFE, ME, EE	(2,1, 1, 1)	(5,5, 5, 5)	(1,1,1,1)
6	0	SE, SOFE, ME, EE	(2,1, 1, 1)	(5,5, 5, 5)	(1,1,1,1)
7	10	SE, SOFE, ME, EE	(1,1, 3, 1)	(5,5, 5, 5)	(1,1,1,1)
8	5	SE, SOFE, ME, EE	(1,1, 2, 1)	(5,5, 5, 5)	(1,1,1,1)
9	10	SE, SOFE, ME, EE	(1,1, 1, 3)	(5,5, 5, 5)	(1,1,1,1)
10	0	SE, SOFE, ME, EE	(2,1, 1, 1)	(5,5, 5, 5)	(1,1,1,1)
11	6	SE, SOFE, ME, EE	(1,1, 1, 1)	(5,5, 5, 5)	(2,2,2,2)
12	10	SE, SOFE, ME, EE	(1,1, 3, 5)	(5,5, 5, 5)	(2,2,2,2)
13	5	SE, SOFE, ME, EE	(1,1, 3, 5)	(5,5, 8, 10)	(2,2,2,2)
14	0	SE, SOFE, ME, EE	(2,3, 2,5)	(8,8, 8, 10)	(1,1,1,1)

B. Mathematical Model Description

A major step in defining the program organization is to evaluate whether the constraints identified in the POAD models will obtain a feasible solution. These constraints consists of resource constraints (e.g., resource availability) and constraints on program activities (e.g., precedence constraints). For this problem, a program has to establish a plan to schedule 14 activities and assign them to resources. The objective of the model is to find a feasible solution (e.g., the completion of activities within a scheduled timeframe) without constraint violations. In addition, the model will consider several types of constraints such as skill requirements and time to fill up an open position.

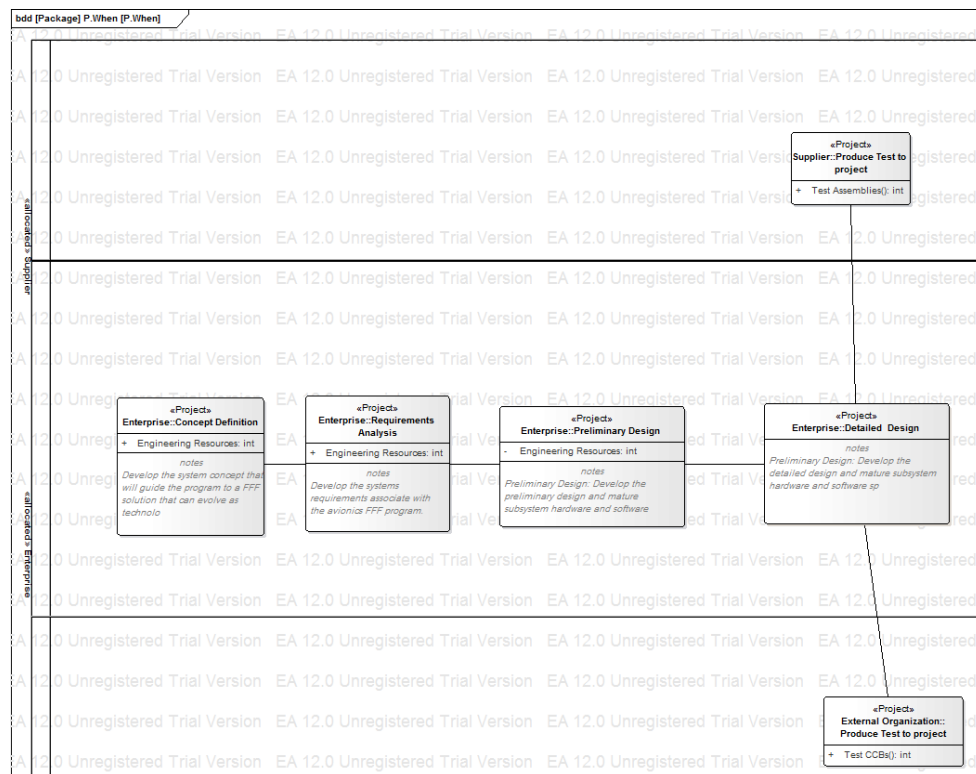
1) Data set

The data related to constructing this model comes from two sources. First, the estimation of the activity duration and resources requirements can be defined from a historical data. Second, the desertion of activity tasks and required capability are derived from the information available in the POAD models. Figure 5 shows the activity graph that is built from different POAD models.

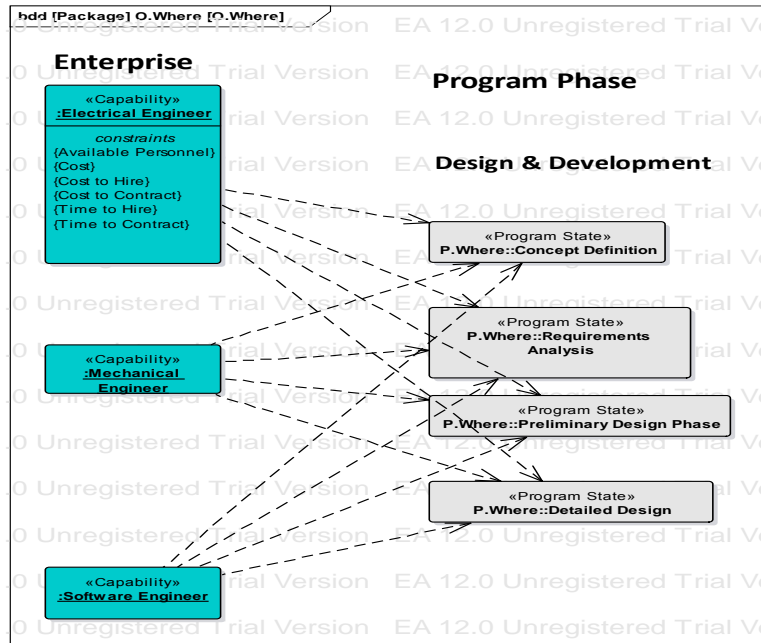
The activity characteristics are shown in Table 3. The activity requires different organizational capability (e.g., Systems Engineering) along with different skill level requirements to perform an activity. The available resources that are able to execute the activities with their skill levels are summarized in Table IV.

TABLE IV: PROGRAM CAPABILITTES MAPPED TO RESPRCES CHARACTERISTICS

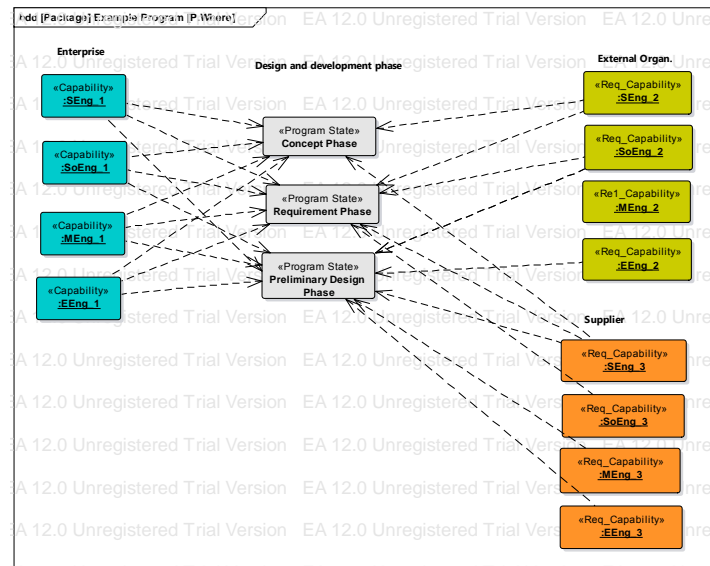
Required Capability	Resource Name	Resource Type	Minimum Skill Level	Maximum Skill Level	Maximum Available
SE	SEng_1	Enterprise	2	5	3
SOFE	SoEng_1	Enterprise	2	5	3
ME	MEng_1	Enterprise	2	5	3
EE	EEng_1	Enterprise	2	5	3
SE	SEng_2	Direct Hire	3	8	2
SOFE	SoEng_2	Direct Hire	3	8	2
ME	MEng_2	Direct Hire	3	8	2
EE	EEng_2	Contractor	3	8	2
SE	SEng_3	Contractor	5	10	1
SOFE	SoEng_3	Contractor	5	10	1
ME	MEng_3	Contractor	5	10	1
EE	EEng_3	Contractor	5	10	1



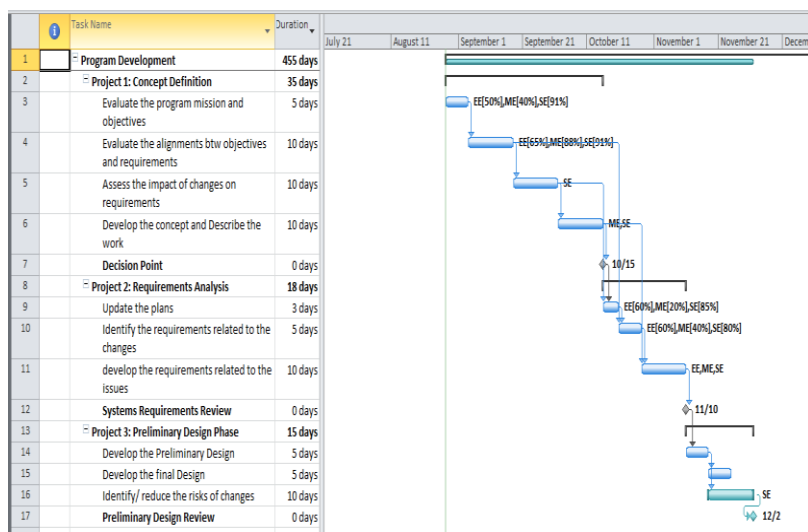
P. When: It identifies the project portfolios that are used to manage the program.



O. Where: This diagram identifies the first set of constraints and objectives that must be considered when developing the program teams. It built upon P. Where.



P. Where: It identifies the organizations that will provide the necessary capabilities.c



D. When - The detailed schedule of the program. This built upon the output of the O. How, O. Where, O. When, O. who, D. How, D. Where, D. When, D. Who, and Resource Analysis Workbook

Fig. 4. Selected POAD models that follow the specifications proposed by POAF.

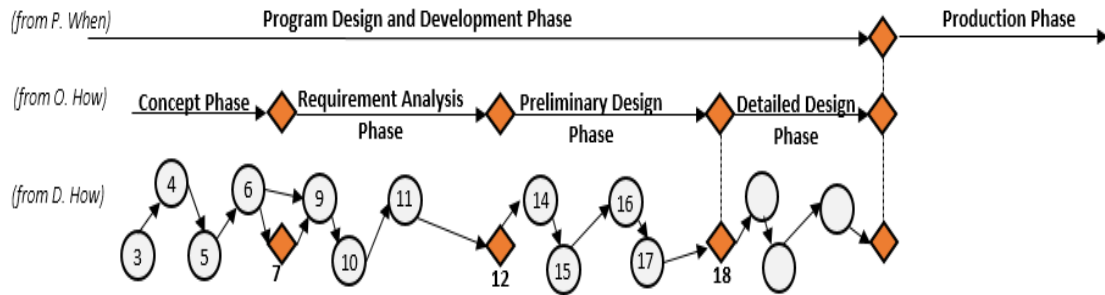


Fig. 5. Program activities and decision points that execute the phases and sub-phases.

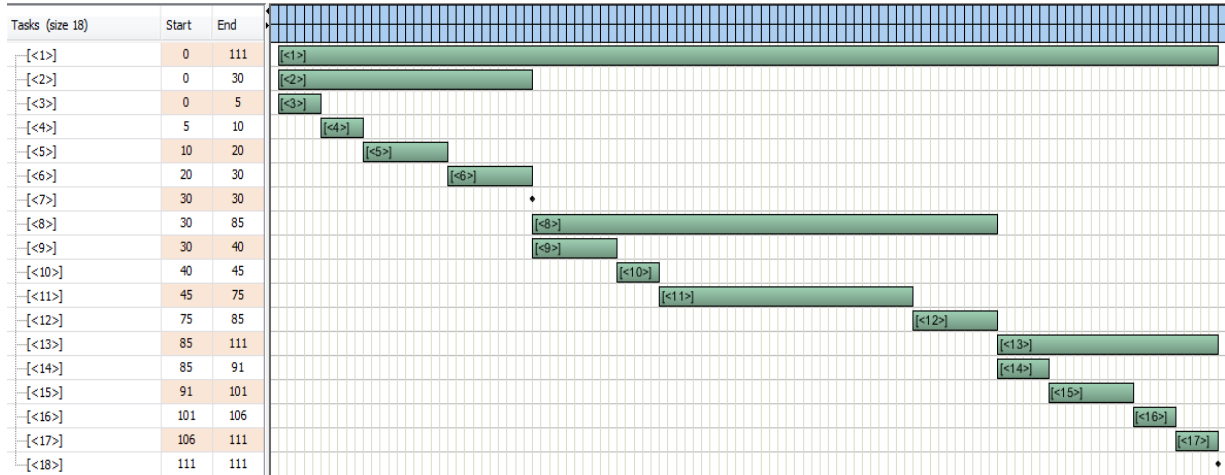


Fig. 6. Program feasible schedule.

2) Results

The program models are coded in an Optimization Programming Language (OPL) format. Then, the model is solved in IBM ILOG CPLEX Optimization Studio. The model was able to find a feasible solution. However, the solution was selected among other possible solutions and there is no guarantee the selected one is the best solution. To find the optimal solution, a solution needs to be evaluated while considering an objective function (e.g., minimizing the personnel cost). Fig. 6 is a presentation of a feasible schedule.

Since the cost of acquiring additional human resources and the transition time between organizations were not considered in this model, the needed capabilities should be supplied by outside resources. This will provide the type of personnel that a program organization needs. Table V shows the team configuration for required capability of activity 1 and activity 2.

TABLE V: TEAM CONFIGURATION FOR ACTIVITY 1 AND ACTIVITY 2

Activity	Required Capability	Quantity Needed	Resource Name	Resource Type
1	SE	2	(SEng_2, SEng_3)	(Direct Hire, Contract)
	SOFE	2	(SOEng_2, SOEng3)	(Direct Hire, Contract)
	ME	2	(MEng_2, MEng_3)	(Direct Hire, Contract)
	EE	2	(EEng_2, EEeng_3)	(Direct Hire, Contract)
	SE	2	(SEng_2, SEng_3)	(Direct Hire, Contract)
2	SOFE	2	(SOEng_2, SOEng3)	(Direct Hire, Contract))
	ME	2	(MEng_2, MEng_3)	(Direct Hire, Contract)
	EE	2	(EEng_2, EEeng_3)	(Direct Hire, Contract))
	SE	2	(SEng_2, SEng_3)	(Direct Hire, Contract)

C. Program Organizational Structure Determination

Bazaraa, *et al.* define a feasible solution as “a set of variables X_1, \dots, X_n satisfying all of the constraints” [13]. A feasible solution to a CSP is the assignment of all variables with a values such that no constraint is violated [14]. If a solution can satisfy all constraints imposed on the design, the organization design may be feasible. For the specified program example discussed in the previous section, a feasible design would select a structure that allows for the use of an outside organization to execute the activities that enable the required capabilities. Virtual organizational structure is the best when there is a need for temporary partnering with an external organization to deliver the final results [2].

V. CONCLUSIONS AND FUTURE WORK

In this paper, quantitative technique is used to extent the POAF and determine the final shape of a program organization structure. The constraint program is used to evaluate the constraints imposed on the design of a program organization. Organization domain of a complex system development activities are based on defining the team, individual, to execute a program. Analyzing the constraints imposed on the program schedule and personnel resources may help to identify the lower levels of the program structure. The necessary information to build the mathematical model was extracted directly from POAF models.

Yet, the methodology seeks to find a better, more optimal solution. Decisions in a program organization should determine the most effective way of evaluating how to allocate resources at the optimal cost and minimize schedule slippage. Other factors need to be discussed as well. For example, how the leadership role and management style

define the organization structure.

REFERENCES

- [1] J. R. Galbraith, *Designing Organizations: Strategy, Structure, and Process at the Business Unit and Enterprise Levels*, 3rd Edition, John Wiley & Sons, Incorporated, 2014, ch.1, pp. 2-3.
- [2] R. Duncan, "What is the right organization structure? Decision tree analysis provides the answer," *Organizational Dynamics*, vol. 7, pp. 59-80, Winter 1979.
- [3] R. Ledbetter, *Organizational Structure: Influencing Factors and Impact in the Grand Prairie Fire Department*, National Fire Academy, 2003.
- [4] B. Obel, "On organizational design: From a linear programming point of view," *Journal of Management Studies*, vol. 15, no. 2, pp. 123-137, 1978.
- [5] K. J. Arrow and R. Radner, "Allocation of resources in large teams," *Journal of the Econometric Society*, pp. 361-385, 1979.
- [6] S. Bourdais, P. Galinier, and G. Pesant, "A constraint programming application to staff scheduling in health care," in *Proc. 9th International Conf.* 2003, pp. 153-167.
- [7] M. Henz, "Scheduling a major college basketball conference-revisited," *Operations Research*, vol. 49, pp. 163-168, 2001.
- [8] A. Alblawi, J. Stracener, and J. Williams, "A conceptual methodology to create a feasible and effective program organization design," in *Proc. American Society for Engineering Management: ASEM*, Indianapolis, IN, 2015.
- [9] T. Browning and A. Yassine, "Resource-constrained multi-project scheduling: Priority rule performance revisited," *International Journal of Production Economics*, vol. 126, no. 2, pp. 212-228, 2010.
- [10] K. Apt, *Principles of Constraint Programming*, Cambridge University Press, 2003.
- [11] M. Schulz and A. Eisenblätter, "Solving frequency assignment problems with constraint programming," *Technische Universität Berlin, Institut für Mathematik*, 2003.
- [12] B. Smith, "A tutorial on constraint programming," University of Leeds, School of Computer Studies, 1995.
- [13] M. Bazaraa, J. Jarvis, and H. Sherali, *Linear Programming and Network Flows*, John Wiley & Sons, 2011.
- [14] S. Brailsford, C. Potts, and B. Smith, "Constraint satisfaction problems: Algorithms and applications," *European Journal of Operational Research*, vol. 119, no. 3, pp. 557-581, 1999.

Adel Alblawi received the B.S. degree from University of Toledo in 2008 in mechanical engineering, and the dual M.S. degree from Southern Methodist University in Operations Research and Systems Engineering.

He is currently a Ph.D. candidate studying systems engineering at the Bobby B. Lyle School of Engineering, Southern Methodist University (SMU). He is also a lecturer at Shaqra University, Saudi Arabia.

Adel current research interests include organizational design, systems architecture, optimization, and managing complex products. He is currently a student member of ASEM, IEEE, and PMI.

Jerrell Stracener is a professor of practice and a founding director of the Southern Methodist University (SMU) Systems Engineering Program (SEP). He teaches graduate-level courses in systems analysis methods and applications. He is the SMU lead senior researcher in the DoD-sponsored Systems Engineering Research Center (SERC). Prior to joining SMU full time in January 2000,

He was employed by LTV/Vought/Northrop Grumman. He conducted and directed systems engineering studies, analysis, and directed systems reliability and supportability programs, on many of the nation's most advanced military aircraft. He was cofounder and leader of the SAE Reliability, Maintainability and Supportability (RMS) Division (G-11).

Dr. Stracener earned PhD and MS degrees in mathematical statistics from SMU and a BS in math from Arlington State College (now the University of Texas at Arlington). His current research interests include reliability, mission reliability modeling and analysis, and systems engineering. He is an SAE Fellow and AIAA associate fellow.

Jeffery Williams has over 40 years of systems engineering experience. He conducted weapon systems and mission analysis for the US Air Force, Navy, and Marine Corps.

He was the first system architect for the Theater High Altitude Air Defense (THAAD) Battle Management Command, Control and Communication (BMC3) system. He also functioned as the principal systems engineer for the fielding of the Common Hardware and Software (CHS) equipment developed by Litton Data Systems for the US Army Dr. Williams has been working as a systems engineer for L-3 Communications Link Simulation & Training, EFW, Bell Helicopter, and the Train Dynamic Systems Division of New York Air Brake.

Dr. Williams is currently working as a manager systems engineering at Securaplane Technologies. Dr. Williams earned PhD degree from SMU in Applied Science and earned both his bachelor and master's degree in mathematics from the University of West Florida.