A Framework & Prototype for Modelling Army Force Generation

Managed Readiness Simulator (MARS), v1.0

Chad Young

Raman Pall

Mike Ormrod
Land Forces Operational Research Team

DRDC CORA TM 2007–54
December 2007

Defence R&D Canada
Centre for Operational Research and Analysis

Land Forces Operational Research Team

A Framework & Prototype for Modelling Army Force Generation

Managed Readiness Simulator (MARS), v1.0

Chad Young

Raman Pall

Mike Ormrod
Land Forces Operational Research Team

This publication contains sensitive information that shall be protected in accordance with prescribed regulations. Release of or access to this information is subject to the provisions of the *Access to Information Act*, the Privacy Act, and other statutes as appropriate.

Distribution is limited to *Defence Purposes Only*

Defense Research and Development Canada – Centre for Operational Research and Analysis

Technical Memorandum

DRDC CORA TM 2007–54

December 2007

Author

Chad Young, Raman Pall, Mike Ormrod

Approved by

D.S. Haslip
Section Head, Land Operations Research

Approved for release by

J.E. Tremblay
Chief Scientist, DRDC CORA

The information contained herein has been derived and determined through best practice and adherence to the highest levels of ethical, scientific and engineering investigative principles. The reported results, their interpretation, and any opinions expressed therein, remain those of the authors and do not represent, or otherwise reflect, any official opinion or position of DND or the Government of Canada.

© Her Majesty the Queen as represented by the Minister of National Defence, 2007

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2007

Abstract

This report provides a comprehensive introduction to the first prototype of the Managed Readiness Simulator (MARS) without delving into the programming details. MARS is designed to provide the user with a collection of modelling constructs that may be combined to track and forecast the Army's ability to generate forces that fulfill government commitments in cases where repetitive, concurrent and/or random tasks compete for personnel, equipment, and facilities resources. An underpinning thrust behind the development of MARS has been the analysis of immediate issues related to the Canadian Army's concept of Managed Readiness. In this role, MARS is to be used to evaluate both capacity and scheduling issues pertaining to the allocation of personnel through the myriad of concurrent activities that take organizational units through a regenerative cycle towards deployment. This report is the second of three companion reports. Data requirements are put into context, concepts related to algorithm design are explained, and ideas critical for applying MARS and creating scenarios are introduced.

Résumé

Le rapport présente une introduction approfondie du premier prototype du programme de simulation de gestion de la disponibilité opérationnelle (MARS) sans entrer dans les détails de programmation. Le MARS fournit à l'utilisateur un ensemble de concepts de modélisation qui peuvent être associés afin de suivre et de prévoir la capacité de l'Armée de terre à mettre sur pied des forces qui respectent les engagements du gouvernement dans les cas où les tâches répétitives, concurrentes et/ou aléatoires se disputent les ressources en personnel, en équipement et en installation. Une orientation sous-jacente au développement du MARS a été l'analyse des questions immédiates liées au concept de la disponibilité opérationnelle de l'Armée de terre canadienne. Dans ce rôle, le MARS doit être utilisé afin d'évaluer à la fois les questions de capacité et d'ordonnancement connexes à l'affectation du personnel à l'aide d'une myriade d'activités concurrentes qui amènent les unités organisationnelles vers un cycle de récupération en vue du déploiement. Ce rapport est le deuxième de trois rapports complémentaires. Les exigences en matière de données sont mises en contexte, les concepts liés au concept d'algorithme sont expliqués et les idées essentielles de l'application du MARS et de la création de scénarios sont présentées.

This page intentionally left blank.

Executive summary

The Managed Readiness Simulator (MARS) is an analytical tool designed to aid in the evaluation of the interacting processes and system demands within the Army's Force Generation system. The aim of this report is to introduce the first MARS prototype, its framework and its general operation without delving into the programming details. It is meant to be accessible to both users and developers in order to:

- put the data requirements for MARS into context;
- understand the concepts behind setting up a scenario;
- introduce general concepts of algorithm design; and
- provide a baseline for continued development.

Context

Within the Army, the colloquial term often used to refer to Force Generation is Managed Readiness (MR). In the current Canadian context, there are three fundamental tenets to MR: The Managed Readiness System (MRS), the Managed Readiness Plan (MRP) and a Whole-Fleet Management approach (WFM)[1]. Collectively, the MRS, MRP, and WFM represent a common frame of reference for the multitude of stakeholders, both tactical and strategic, that must coordinate to produce theatre ready forces with appropriate capabilities in an efficient and time effective fashion.

The current reality is that government commitments and the escalating needs of commanders in theatre continue to increase the demand for continuous operational capability without recourse for an operational pause. This reality, along with other intrinsic issues related to transformation and the aging demographic within the Canadian Forces, have prompted the need to develop and analyse new and/or revised Managed Readiness options. These include the following.

Policy/Process Options

- Adopting new policies for setting deployment length, MRS cycle lengths, and operational tempo limits.
- Adjusting unit assignments and scheduling options within the MRP.
- Altering training requirements and/or related scheduling practices.
- Adjusting how critical resources are to be utilized and/or scheduled.
- Reconsidering human resources guidelines related to certain policies (*e.g.* waivers policies).

[1] In essence, the managed readiness system is a framework to guide the scheduling of many large tasks that must occur concurrently. The Army's managed readiness plan is a particular approach for assigning specific resources to those tasks. The Whole Fleet Management architecture effectively removes "ownership" of certain equipment items such as vehicles from individual units and redistributes these into specialized stocks.

Supply Related Options

- Adjusting the allocation of capabilities between the regular & reserve forces as well as between the Army, Navy, Air Force, and Joint elements.
- "Re-rolling" and/or expanding the capabilities of certain groups of personnel in the establishment.
- Expanding and/or adjusting the organizational structure of the establishment.

Demand Related Options

- Increasing or decreasing the number of lines of operation in the MRP.
- Increasing or decreasing the personnel commitments on each line of operation.
- Setting new conditions on the number of lines of operations that may be filled at any one time.
- Rethinking how commitment levels for personnel should be determined and/or articulated in the future.

Unfortunately, in as much as it is straightforward to devise tactical methods by which to solve immediate force generation problems, it is much more difficult to identify the associated strategic consequences over longer time frames. The dynamic nature of MARS is to make it of particular utility in this regard.

MARS Project Goals

The MARS project was instigated to develop a tool to aid in the investigation of these options. It was designed with a number of functions in mind.

- To provide dynamic gap analysis between supply and demand based on proposed capability demands and existing "Managed Readiness" processes.
- To provide a platform to forecast and understand issues of sustainability in terms of operational capability, resource utilization costs, training and maintenance efforts, and the ultimate quality of fighting forces.
- To facilitate the development, validation, and testing of new and different strategies for resource allocation, activity scheduling, unit synchronization, and functional management policy related to Force Generation.
- To examine the implications of new and existing policies related to Human Resources and materiel.
- To assess potential changes of the structure of forces within the establishment.
- To facilitate "what if?" analyses.

The first stage of MARS development has been to provide a “proof of concept”. More specifically, the thrust behind the first phase in the development of MARS has been to provide a platform for forecasting the interrelated impacts of Task demands on resource utilization in a way that can provide quantifiable assessment of a variety of operational alternatives that may or may not be tied to a specific MRP or MRS.

MARS Implementation

In its implementation, the architecture of MARS is composed of four basic components.

1. A graphical interface that facilitates the creation and management of scenarios.
2. A discrete event simulation kernel.
3. A relational database that stores scenario input and output data.
4. A graphical interface that provides access to a series of postprocessing routines for analysis and interpretation of simulation results.

An introduction to the whole MARS application is provided at the beginning of this report. The focus of remaining discussions centre around the 2nd and 3rd items in the list above, namely the simulation kernel and the relational database. On a conceptual level the simulation kernel and relational database are built on two levels of abstraction. On the first and most basic level of abstraction, individual personnel, equipment, and facilities are grouped together into packages called *Resources*. Resources "flow" between processes called *Activities* that make up larger *Tasks*. On the second level of abstraction individual Activities "flow" between stations in what is termed the Activity lifecycle.

Besides the fact that MARS is a dynamic forecasting tool that can incorporate stochastic effects, one aspect that sets MARS apart from other management tools for force generation is that it is built to enable the concurrent assignment of Units to multiple tasks. This functionality is enabled because at a root level MARS keeps track of individuals not just establishment Units.

MARS Data Requirements

As a consequence of its design, MARS allows the user to create what are called *Task-Templates*. Task-Templates are used to generate individual instances of a Task over the duration of a simulation. Inherent in the creation of a Task-Template is the requirement for a wide range of information, including:

- The hierarchy, position requirements, personnel manning levels, and left-out-of-battle estimates for the establishment;
- A tabulation of significant and/or critical Tasks;
- A breakdown of each Task into smaller Activities;
- The network logic that governs the succession of Activities in each Task;
- The Resource requirements for each Activity;
- Instantiation schedules for each Task (*e.g.* inputs derived from the MRS), and
- Resource allocation schedules/rules for filling Activity requirements (*e.g.* inputs derived from the MRP).

Collectively, these represent the data requirements for the first MARS prototype. Once described by the user, this information is organized and stored within a relational database. This database also holds a collection of temporal data generated when running a simulation.

Together with the input, the output provides the raw data for several post-analysis utilities that are also incorporated into MARS.

MARS Analysis Toolset

Within the first MARS prototype, post-analysis utilities enable the user to assess simulation results from two perspectives, that of a force employer and that of a force generator. Simulation results are easily assessed from a Force Employment perspective to show whether the Army's Field Force can fill continuous demands for theatre-ready units based on existing or proposed managed readiness practices. From a Force Generation perspective, the analysis routines in MARS are designed to analyse how resource utilization is distributed across the Army. Both types of analyses can be conducted at various degrees of aggregation. In future versions of MARS the incorporation of human resource processes such as the recruitment, promotion, posting, attrition, and retirement of personnel is expected to allow for the forecasting of Force Generation issues far into the future.

In its present state, the MARS prototype is intended for "off-line" analysis with direct input from military planning staff in directorates such as the Army G3 (formerly the Directorate of Land Force Readiness (DLFR)). In the future it may be possible to make the algorithms more amenable for "on-line" analysis at operational planning conferences. The aim of the latter would be to provide immediate feedback about pressing issues through direct collaboration with representatives from the Land Force Area Commands, the Land Force Doctrine and Training System, and the Individual Training system. Future directions for MARS are to be adressed on a rolling basis.

Chad Young, Raman Pall, Mike Ormrod; 2007; A Framework & Prototype for Modelling Army Force Generation – Implementation of MARS v1.0 in a Discrete-Event Simulation Environment; DRDC CORA TM 2007–54; Defense Research and Development Canada – Centre for Operational Research and Analysis.

Sommaire

Le programme de simulation de gestion de la disponibilité opérationnelle (MARS) est un outil analytique conçu pour appuyer l'évaluation des processus interactifs et des requêtes de système dans le système de mise sur pied d'une force de l'Armée de terre. L'objectif de ce rapport est de présenter le premier prototype du MARS, son cadre et son fonctionnement général sans entrer dans les détails de programmation. Les utilisateurs et les développeurs doivent y accéder afin de :

–mettre les exigences relatives aux données pour le MARS en contexte ;
–comprendre les concepts sous-jacents à la mise en place d'un scénario ;
–présenter les concepts généraux de la conception d'algorithmes ;
–fournir une base au développement continu.

Contexte

Au sein de l'Armée de terre, le terme courant souvent utilisé pour parler de la mise sur pied d'une force est la gestion de la disponibilité (GD). Dans le contexte actuel canadien, il y a trois principes fondamentaux de GD : le Système de gestion de la disponibilité (SGD), le Plan de gestion de la disponibilité (PGD) et la démarche du système de gestion de l'ensemble du parc de véhicules (SGEPV). Ensemble, le SGD, le PGD et le SGEPV, constituent un cadre commun de référence pour de nombreux partenaires, à la fois tactiques et stratégiques, qui doivent collaborer afin d'outiller des forces prêtes au combat dans le théâtre de façon efficace et rapide.

La réalité actuelle est telle que les engagements du gouvernement et les besoins croissants des commandants dans le théâtre continuent d'accroître la demande pour une capacité opérationnelle continue sans avoir recours à un arrêt opérationnel. Cette réalité, ajoutée aux autres questions intrinsèques liées à la transformation et à la population vieillissante des Forces canadiennes, a créé le besoin d'élaborer et d'analyser de nouvelles options de gestion de la disponibilité ainsi que des options révisées. Nous retrouvons ce qui suit :

Options liées aux politiques et aux processus
–Adopter de nouvelles politiques permettant d'établir la longueur du déploiement, les longueurs de cycle du SGD et les limites du rythme des opérations.
–Ajuster les affectations des unités et les options d'ordonnancement dans le PGD.
–Ajuster les exigences en matière d'instruction et/ou liées aux pratiques d'ordonnancement.
–Ajuster le niveau critique des ressources qui seront utilisées et/ou ordonnancées.
–Reconsidérer les lignes directrices des ressources humaines liées à certaines politiques (p. ex., politiques sur l'exonération).

Options liées à l'approvisionnement
–Ajuster l'attribution des capacités entre la Force régulière et la Réserve, ainsi qu'entre l'Armée de terre, la Force aérienne et les éléments interarmées.
–Réaffecter et/ou étendre les capacités de certains groupes de personnel au sein de l'organisation.

–Étendre et/ou ajuster la structure de l'organisation.

Options liées aux demandes
–Augmenter ou réduire le nombre de lignes d'opération dans le PGD.
–Augmenter ou réduire les engagements du personnel pour chaque ligne d'opération.
–Établir de nouvelles conditions concernant le nombre de lignes d'opération qui peuvent être remplies en même temps.
–Réexaminer de quelle manière les niveaux d'engagement du personnel devraient être déterminés et/ou organisés dans le futur.

Malheureusement, même s'il est assez simple de diviser les méthodes tactiques utilisées pour résoudre les problèmes immédiats de mise sur pied d'une force, il est beaucoup plus difficile de cerner les conséquences stratégiques associées à long terme. La nature dynamique du MARS vise à créer un outil particulier à cet égard.

Buts du projet du MARS

Le projet du MARS a été entrepris afin de développer un outil permettant d'examiner ces options. Il a été conçu avec un certain nombre de fonctions en tête.

–Fournir une capacité d'analyse dynamique des écarts entre l'approvisionnement et les demandes basés sur les demandes de capacité proposées et les processus de "gestion de la disponibilité" en place.
–Fournir une plate-forme permettant de prévoir et de comprendre les questions de durabilité en termes de capacité opérationnelle, de coût d'utilisation des ressources, d'effort d'instruction et de maintenance et de qualité ultime des forces de combat.
–Favoriser le développement, la validation et la mise à l'essai de nouvelles stratégies pour l'attribution des ressources, l'ordonnancement des activités, la synchronisation des unités et de la politique de gestion fonctionnelle de mise sur pied d'une force.
–Examiner l'incidence des nouvelles politiques et des politiques déjà en place liées aux ressources humaines et au matériel.
–Évaluer les changements possibles de la structure des forces au sein de l'organisation.
–Favoriser les analyses de type " et si ? ".

La première étape de développement du MARS consistait à fournir une " validation de principe ". En particulier, la direction derrière la première étape dans le développement du MARS a été de fournir une plate-forme visant à prévoir les répercussions interdépendantes des demandes de travail sur l'utilisation des ressources de façon à pouvoir effectuer une évaluation quantifiable d'une variété d'options opérationnelle qui peuvent ou non être associées précisément à un PGD ou un SGD.

Mise en place du MARS

Pour cette mise en place, l'architecture du MARS est composée de quatre éléments de base.

1. Une interface graphique qui favorise la création et la gestion de scénarios.
2. Un programme de simulation d'événements discrets.
3. Une base de données relationnelle qui stocke des données de scénarios d'entrées et de sorties.
4. Une interface graphique qui fournit un accès à une série de routines post-traitement pour l'analyse et l'interprétation des résultats de simulation.

Une introduction à l'application complète du MARS se trouve au début de ce rapport. Le reste des discussions portera sur les points deux et trois susmentionnés, donc le programme de simulation et la base de donnée relationnelle. Au niveau conceptuel, le programme de simulation et la base de données relationnelle sont conçus en fonction de deux niveaux d'abstraction. Selon le premier niveau de base d'abstraction, chaque membre du personnel, l'équipement et les installations sont groupés dans des ensembles nommés les " ressources ". Les " ressources " circulent entre les processus nommés les " activités " qui forment la majorité des tâches. Selon le deuxième niveau d'abstraction, les " activités individuelles " circulent entre les stations, ce que nous pourrions nommer le " cycle de vie d'une activité ".

Bien que le MARS soit un outil de prévision dynamique qui peut insérer des effets stochastiques, l'un des aspects qui distinguent le MARS des autres outils de gestion pour la mise sur pied d'une force est sa construction permettant les affectations simultanées des unités à diverses tâches. Cette fonction est activée parce qu'à la base le MARS tient compte des personnes, non seulement des unités de l'organisation.

Exigences en matière du MARS

Grâce à sa conception, le MARS permet à l'utilisateur de créer ce que nous nommons des " modèles de tâche ". Ces modèles de tâche sont utilisés pour produire des cas individuels pour une tâche au cours d'une simulation. Une vaste étendue d'information est essentielle à la création d'un modèle de tâche, et doit comprendre ce qui suit :

–La hiérarchie, les besoins en postes, les niveaux de dotation en personnel et les évaluations des laissés hors de la bataille pour l'organisation ;
–Un tableau des tâches importantes et/ou essentielles ;
–Une analyse de chaque tâche selon de plus petites activités ;
–La logique des réseaux qui dirigent la succession des activités pour chacune des tâches ;
–Les besoins en ressources pour chaque activité ;
–Le calendrier des instanciations pour chaque tâche (p. ex., entrées dérivées du SGD) ;
–Les calendriers/règles d'attribution des ressources afin de remplir les exigences des activités (p. ex., entrées dérivées du PGD).

Dans l'ensemble, ces points représentent les exigences relatives aux données pour le premier prototype du MARS. Une fois fournie par l'utilisateur, cette information est organisée et stockée dans une base de données relationnelle. Cette base de données comprend aussi une collection de données temporelles produites pendant l'exécution d'une simulation.

Accompagnées des entrées, les sorties fournissent des données brutes pour plusieurs utilitaires post-analyse qui sont aussi ajoutés au MARS.

Boîte à outils d'analyse du MARS

Avec le premier prototype du MARS, les utilitaires post-analyse permettent à l'utilisateur d'évaluer les résultats de la simulation selon deux points de vue, dont celui de l'utilisateur d'une force et celui du responsable de la mise sur pied d'une force. Les résultats de la simulation sont facilement évalués d'un point de vue de l'utilisateur d'une force afin de montrer si les forces de campagne de l'Armée de terre peuvent répondre aux demandes continues des unités prêtes au combat selon des pratiques de gestion de la disponibilité existantes ou proposées. Du point de point du responsable de la mise sur pied d'une force, les routines d'analyse dans le MARS sont conçues afin d'analyser comment l'utilisation des ressources est répartie à la grandeur de l'Armée de terre. Les deux types d'analyses peuvent être menés à divers niveaux de regroupement.

Dans les prochaines versions du MARS, nous prévoyons que l'intégration des processus de ressources humaines (le recrutement, la promotion, l'affectation, l'attrition et la retraite du personnel) permettra de faire des prévisions quant aux questions de mise sur pied d'une force à très long terme. À l'étape actuelle, le prototype du MARS est conçu pour l'analyse " hors ligne " avec une saisie directe des planificateurs militaires dans les directions comme le G3 de l'Armée de terre (anciennement Disponibilité opérationnelle de la Force terrestre). Dans le futur, il devrait être possible de rendre ces algorithmes davantage aptes aux analyses " en ligne " lors des conférences de planification opérationnelle. L'objectif de cette dernière option serait de pouvoir fournir immédiatement une rétroaction sur les questions urgentes grâce une collaboration directe avec les représentants du Secteur de la Force terrestre, du Système de la doctrine et de l'instruction de la Force terrestre et du Système d'instruction individuelle. Les directions futures pour le MARS seront examinées de façon régulière.

Chad Young, Raman Pall, Mike Ormrod; 2007; A Framework & Prototype for Modelling Army Force Generation – Implementation of MARS v1.0 in a Discrete-Event Simulation Environment; DRDC CORA TM 2007–54; RDDC – Centre pour la recherche et l'analyse opérationnelles.

Table of contents

Abstract i

Résumé i

Executive summary iii

Sommaire vii

Table of contents xi

List of Figures xiv

1 Force Generation & MARS 1

1.1 Aim 1

1.2 Force Generation & Managed Readiness 2

1.3 The Managed Readiness System (MRS) 2

1.3.1 The Managed Readiness Plan (MRP) 4

1.3.2 Managed Readiness Perspectives 6

1.4 Development of the Managed Readiness Simulator (MARS) 7

1.4.1 The MARS Architecture 8

1.4.2 MARS Data Requirements 9

1.4.3 The MARS Graphical User Interface (GUI) 11

1.4.4 The Simulation Kernel 19

1.4.5 The Relational Database 19

1.4.6 The Analysis Interface 19

1.5 MARS Kernel Design: Two Fundamental Design Concepts 24

1.5.1 The Resource-Activity-Task Paradigm 24

1.5.2 The Activity-Station-Lifecycle Paradigm 26

2 Tasks, Activities & Resources 28

2.1 Defining a Task-Template - An Overview 28

2.2 The Task Instantiation Schedule 30

2.3 Elements, Element-Options & Element-Option Schedules 31

2.4 Activity Definition 33

2.4.1 Activity Attribute Categories 35

2.4.2 General Attributes 35

2.4.3 Temporal Attributes 37

2.4.4 An Attribute to Alter the Status of the Parent Task 38

2.4.5 Internal Routing Attributes 38

2.4.6 External Routing Attributes 41

2.4.7 Attributes to Set Resource Status 42

2.5 Resource Selection Attributes 47

2.5.1 Defining a Resource Supply 47

2.5.2 Defining A Resource Requirement/Demand 49

2.5.3 Constructing a Resource Selection Schedule 50

2.5.4 The Roto Mechanism 51

2.5.5 The Stealing Mechanism 57

2.5.6 Setting Resource Requirements & Selection Schedules/Rules within an Activity 62

2.6 Closure 63

3 The Activity Lifecycle 65

3.1 The Lifecycle of an Activity – An Overview 65

3.2 Viewing the Activity Lifecycle from the Perspective of a Resource . 67

3.3 Station 1: Task Generation/Instantiation 69

3.4 Station 2: The Global Wait Queue 78

3.5 Station 3: Resource Acquisition 78

3.5.1 Resource Requirements 78

3.5.2 The Resource Selection Mechanisms 81

3.5.3 The Roto Mechanism 83

3.5.4 The Stealing Mechanism 85

3.5.5 Assigning Individuals to Positions 87

3.6 Station 4: Activity Processing 93

3.7 Station 5: Resource Routing & Activity Dissolution 94

3.8 Closure 94

4 Future Development 96

References 98

Annexes 99

A The Whole Fleet Management System (WFM) 99

B List Of Abbreviations 100

Distribution list 103

List of Figures

1 Managed Readiness in the Context of Army Transformation 3

2 An example Managed Readiness System (MRS) 4

3 A Sample Managed Readiness Plan (MRP) 5

4 Basic illustration of MARS architecture & dataflow. 8

5 MARS constructs used to model Managed Readiness scenarios 10

6 The main MARS Control Panel Interface 12

7 The Establishment Builder Interface 14

8 The ORBAT Builder Interface 15

9 The Task Builder Interface 16

10 The Basic Resource Scheduler Interface 17

11 The Advanced Resource Scheduler Interface 18

12 Illustrative MARS Output (Force Employer Perspective). 21

13 Illustrative MARS Output (Force Generator Perspective). 22

14 Illustrative MARS Output (Unit Perspective). 23

15 An illustration of the Resource-Activity-Task paradigm 25

16 An illustration of the Activity-Station-Lifecycle paradigm 26

17 A Task-Template, Task Instantiation Schedule & Instantiation Time Line . . 29

18 The Hierarchical Breakdown of a Task into Elements and Activities. 32

19 The Element-Option Concept 34

20 General attributes of an Activity. 36

21 Temporal attributes of an Activity. 37

22 Altering the Status of a Task in order to Transfer Resources 39

23 Pictorial illustration of an Activity 40

24 External Routing 41

25 External Routing to indicate Temporal Dependence 42

26 Attributes that serve to alter the status of individuals in a Resource are attached to Feeder and Finder nodes. 43

27 Participation Type Example 44

28 Qualifier Status: An On-Call Example 45

29 Status Definitions and Timing 46

30 Finder Node Information 48

31 The Roto Concept 52

32 The Roto Mechanism. 54

33 An example using the Stealing Mechanism 58

34 Stealing Mechanism Inputs 59

35 A Pictorial View of the Activity Lifecycle. 66

36 Data Structure used to store Task Instantiation Schedules 70

37 Timing Parameters 71

38 Relational Data Structure used to store Task-Template Hierarchy 72

39 Data Structure used to store Element-Option Schedules 73

40 Relational Data Storage Structure for Activities. Part 1. 74

41 Relational Data Storage Structure for Activities. Part 2. 75

42 Relational Data Storage Structure for Activities. Part 3. 76

43 Relational Data Storage Structure for Specific Tasks/Elements/Activities. . . 77

44 The Relational Data Structure used to store Status Definitions: Doing-Status. 79

45 Relational Data Structure used to Store Status Definitions: B&E Status. . . . 80

46 Relational Data Structure used to store Resource Requirement Information . 82

47 Relational Data Structure used to Store Roto Mechanism Information. 84

48 Relational Data Structure used to Store the Establishment. 85

49 Relational Data Structure used to Store Personnel Information. 86

50 Relational Data Structure used to Store Stealing Mechanism Information. . . 88

51 Sorting candidate individuals to fill positions in a Resource 89

52 Relational Data Structure used to track the movement of a Resource. 91

53 Relational Data Structure used to store ORBAT information. 92

1 Force Generation & MARS

The end of the Cold War and the downsizing of the Canadian Forces in the early 1990's saw the beginning of a trend characterized by an ever-increasing deployment tempo. This trend, which persists to the present day and is exemplified by current missions in the Greater Middle East, continues to place a heavy strain on the Army. This fact, coupled with the popular belief that a decision to pull Canadian Troops out of Afghanistan would ultimately result in a redeployment to one of many other venues, is just one of the driving factors causing the Canadian Army to re-evaluate and develop new strategies for Force Generation. The development of the Managed Readiness Simulator (MARS) is an important initiative in this regard. Several other allied countries are also pursuing the development of similiar analytical tools (*e.g.* [1, 2, 3]).

An important aspect of MARS is that it is designed to provide the user with a collection of modeling constructs that may be combined to track and forecast the Army's ability to generate forces that fulfill government commitments in cases where repetitive, concurrent and/or random Activities compete for personnel, materiel (*i.e.* equipment and vehicles), and facilities resources. An underpinning thrust behind the development of the first MARS prototype has been for the analysis of immediate issues related to the Canadian concept of Managed Readiness (MR). In this role, MARS is to be used to evaluate both capacity and scheduling issues pertaining to the allocation of personnel through the myriad of concurrent activities that take organizational units through a regenerative cycle towards deployment. An underlying goal in the design of MARS has been to produce a descriptive tool that is amenable to a broad range of scheduling, resource allocation, and related capacity problems that need to be solved in a transparent way with a human-in-the-loop. The first MARS prototype represents the first step toward this goal.

1.1 Aim

This report is the second of three companion reports about the first MARS prototype [4, 5]. The aim of this report is to provide an introduction to the first MARS prototype, its framework and its general operation without delving into the programming details. It is meant to be accessible to developers and users to:

1. Put the data requirements for MARS into context;
2. Understand the concepts behind setting up a scenario;
3. Introduce general concepts of algorithm design;
4. Provide a baseline for continued development.

In the next section a brief background about Force Generation and the Canadian concept of Managed Readiness is presented to provide a context to some of the terminology and examples used in the remainder of this report. Concluding this section is a basic outline of the MARS architecture, a brief discussion concerning data requirements and MARS outputs, and an introduction to the Graphical User Interface (GUI) that is currently in development.

1.2 Force Generation & Managed Readiness

Force Generation is a categorical term that fits on a continuum between that of Force Development and Force Employment. A sustainable force generation system encapsulates methods to reconstitute, organize, manage and renew forces to:

- Maintain a continual readiness posture;
- Achieve a sustainable level of capability in prolonged operations; and
- Preserve a flexible level of contingency.

The framework for implementing force generation processes in the Canadian Army is known colloquially as Managed Readiness. It has three main aspects.

1. A predictable training, deployment, and reconstitution scheme called the Managed Readiness System (MRS).
2. A cyclical assignment schedule called the Managed Readiness Plan (MRP).
3. A Whole Fleet Management (WFM) approach that is manifest in the implementation of a pooled equipment architecture.

As illustrated in Figure 1, Managed Readiness is at the heart of the recent plans to transform the Army to deal with the geopolitical realities of modern warfare and peace making that underscore a requirement to balance increasing demands of a complex threat environment, quick response, and decreasing or limited resources. Basic MRS and MRP concepts are introduced in the next subsections. The WFM approach is discussed in Annex A.

1.3 The Managed Readiness System (MRS)

The first tenet of Managed Readiness is known as the Managed Readiness System (MRS). It is a scheme for training, deploying and reconstituting forces that is intended to provide a prescribed set of capabilities to one or more theatres of operation in a way that is sustainable over periods spanning many years. At present the MRS is based on a three year rotation with six phases that may be best illustrated in terms of the hexagon in Figure 2. Organizational units within the Army's Field Force are considered to flow around the hexagon in a repeating cycle. In theory, certain Activities are to occur at each phase of the cycle. These are explained below ([6]).

Phase 1 - Recovery/Reconstitution: Personnel enter this phase following re-deployment from a theatre of operation. During this phase they are subject to post-deployment leave allowances and waivers policies that restrict the kind of Activities that they can be expected to perform. In this phase personnel are expected to take required career courses and/or begin certain individual training courses. The turnover of personnel is expected to be high in this phase as units are reconstituted via internal transfers and external postings.

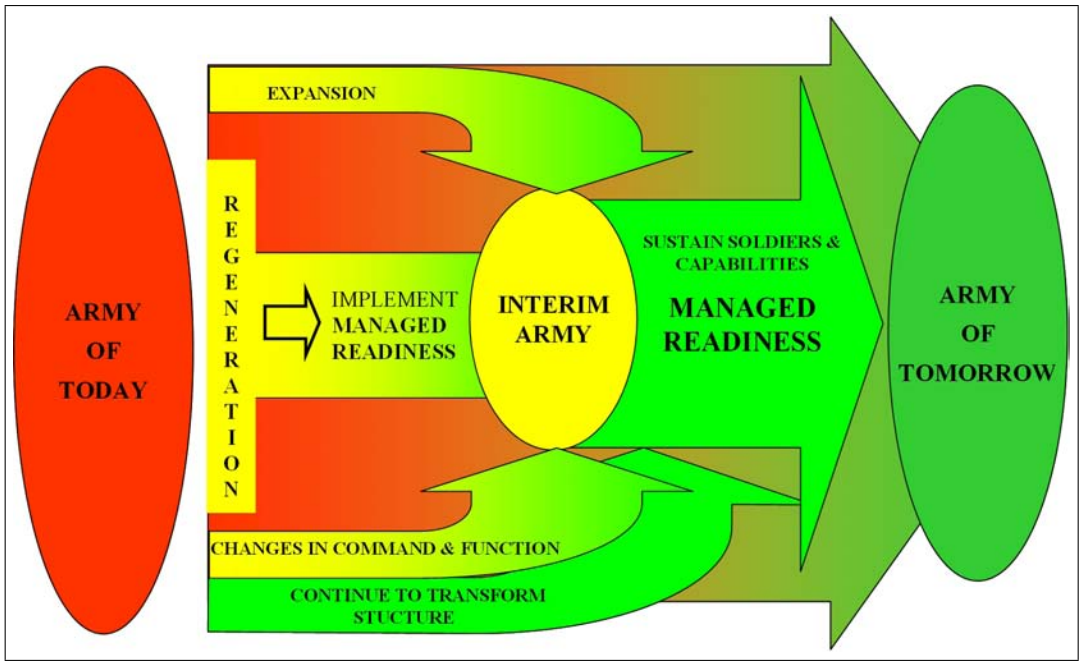


Figure 1 *: Managed Readiness in the Context of Army Transformation. Further details about the Managed Readiness concept in relation to Transformation initiatives instigated in 2004 are contained in Ormrod, Young & Pall [4].*

Phase 2 - Regeneration/Support: During this phase, turnover is supposed to be limited to non-key personnel as particular units commence with collective training exercises (Training Levels 2 - 4 [6]) in preparation to be on-call for potential surge and/or emergency tasks like Non-Combat Evacuation Operations (NEO) and Disaster Assistance Response Teams (DART). In this phase individuals and/or collective teams of personnel may also be engaged in career courses, sovereignty exercises, small unit exchanges, transformational Activities, and High Readiness (HR) support exercises.

Phase 3 - Regeneration/Support: In phase 3 certain organizational units are assigned to participate in surge tasks, participate in transformational assignments, support HR training, or remain on call for emergency tasks. Collective training exercises (Training Levels 2 - 4 [6]) also continue in this phase.

Phase 4 - High Readiness Training/Support: In this phase units begin on a "road to high readiness" in preparation for a potential deployment. Collective training exercises (Training Levels 2 - 4) are conducted with high priority. Support to national/international training opportunities (including Brigade level training exercises (BTE's)) and exchange training (for small units) are encouraged where applicable in this phase.

Phase 5 - High Readiness Training: In this phase Mission Specific Task Forces (MSTFs) are formed and large scale (Training Level 5 [6]) collective training occurs at the Canadian Manoeuver Training Centre (CMTC). Theatre Mission Specific Training

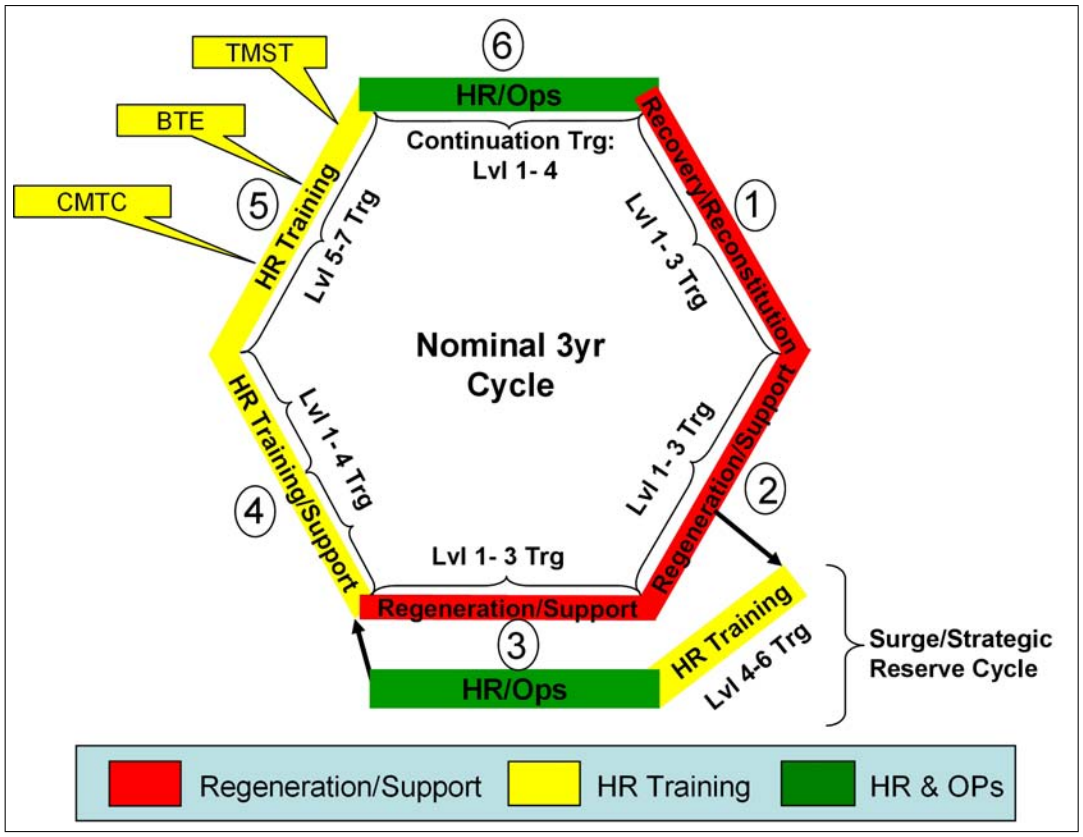


Figure 2 : *An example Managed Readiness System (MRS).*

(TMST) and extra Task specific exercises are also conducted in this phase. The main focus on this phase is preparation for deployment, or potential deployment, to a theatre of operation.

Phase 6 - Deployment/High Readiness Reserve: All personnel in this phase are expected to be deployed, or ready to deploy, on short notice to a major theatre of operation. Personnel not deployed may participate in collective training exercises and provide support to mission specific courses and the individual training system.

Of course, as this is a simple outline of the MRS, Activities within and between each of the phases may be rearranged to account for potential resource constraints and/or extra surge demands.

1.3.1 The Managed Readiness Plan (MRP)

The second tenet of Managed Readiness is the construction of a cyclic assignment schedule called the Managed Readiness Plan (MRP). In a sense, the Managed Readiness Plan is "impressed" on the Managed Readiness System. It specifies when particular Units are to

enter and leave the Deployment/High Readiness Phase (*i.e.* phase 6) of the MRS. A sample illustration of a simplified MRP is presented in Figure 3.

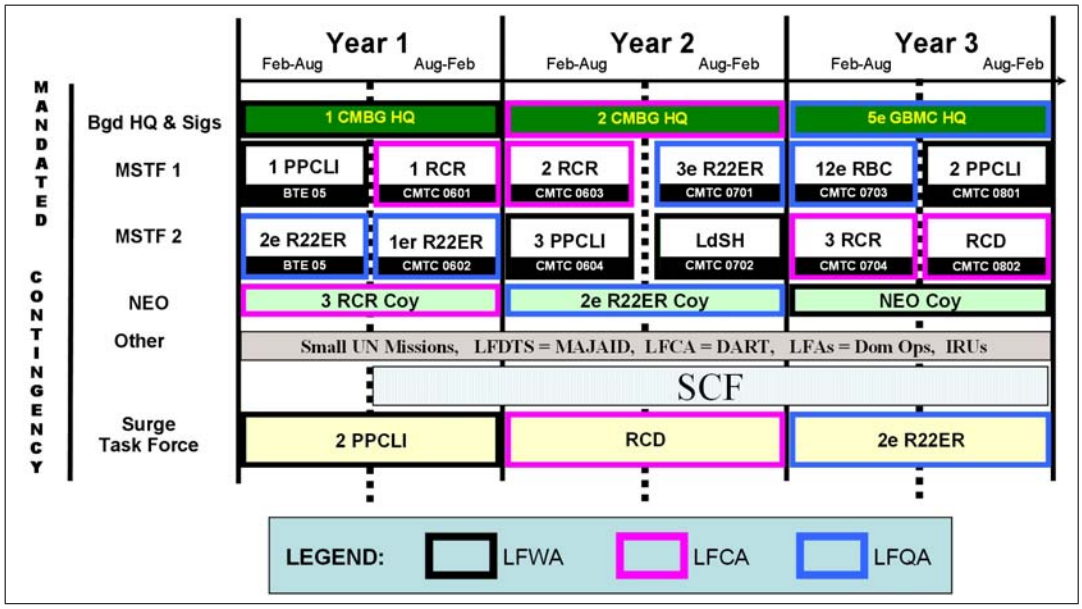


Figure 3 *: A Sample Managed Readiness Plan (MRP) [6]*

Across the top of the plan is a timeline. Demarcated down the left of the plan are specific "Lines of Operation" (LineOps). The LineOps depicted here are consistent with the continuous capability expectations[2] for the Army. These LineOps include: 1 Brigade HQ and Signal Squadron, the equivalent of 2 Mission Specific Task Forces (MSTF), 1 Non-Combat Evacuation Operation Element (NEO), a Strategic Surge Reserve, a Strategic Task Force with Marine type capabilities (SCF), a Disaster Assistance Relief Element (DART), a Domestic Operations Force (DomOps), and a Major Aid Disaster Element (MAJAID). Of these, the MSTFs and the Brigade HQ are expected to demand the greatest number of resources on a continual basis.

Each line in the MRP indicates which Army Units are responsible to fill deployment demands. For example, the MRP in Figure 3 indicates that the International Brigade HQ is to be supplied in sequence by the HQ SubUnits of `1`, `2` and `5e Brigades`. The Units responsible for providing personnel to fill the headquarters Element of the first MSTF are: `1 PPCLI`, `1 RCR`, `2 RCR`, `3e R22ER`, `12e RBC` and `2 PPCLI` respectively. Units that are responsible for providing personnel to fill other parts of an MSTF (*e.g.* artillery, reconnaissance, and engineering elements) are not indicated but are usually outlined in more detailed versions of the MRP.

[2]Some of the expectations set out in Figure 3 are also to be shared with other parts of the CF.

1.3.2 Managed Readiness Perspectives

One perspective from which to understand the amalgam of MR concepts is to consider that, in order to deal with the ever increasing demands, the strategy employed by the Army has been:

1. To set a capability target and a contingency threshold for continual deployment, and
2. To develop a systematic methodology by which to produce and validate forces to meet these targets.

The continual capability targets are embodied in the 2 MSTF's and the Brigade HQ in Figure 3. The contingency threshold is represented by the NEO, DART, and other Surge Lines. The systematic methodologies to meet these targets are embodied in the MRS, MRP and WFM. Unfortunately, the current reality is that government commitments and escalating demands from commanders in theatre have pushed demands above the continual capability target without recourse for an operational pause. This reality, along with other intrinsic issues related to transformation and the aging demographic within the Canadian Forces, have prompted the need to develop and analyse new and/or revised MR options. In this regard some potential alternatives include:

1. Adjusting the allocation of capabilities between the regular & reserve forces as well as between the Army, Navy and Air Force;
2. "Re-rolling" and/or expanding the capabilities of certain groups of personnel in the establishment;
3. Expanding and/or adjusting the structure of the establishment;
4. Increasing/decreasing the number of lines or operation in the MRP;
5. Increasing/decreasing the personnel commitments on each line of operation;
6. Setting new conditions on the number of operation lines that may be filled at any one time;
7. Rethinking how commitment levels should be determined and articulated in the future;
8. Adopting new policies for setting deployment length, MRS cycle lengths, and operational tempo limits;
9. Adjusting unit assignments and scheduling options within the MRP;
10. Altering training requirements and/or related scheduling practices;
11. Adjusting how critical resources are to be utilized and/or scheduled; and
12. Reconsidering guidelines related to certain human resources policies *(e.g.* waivers policies).

These alternatives, as well as others, may be easily categorized in terms of those that target the "supply" of resources[3], those that target the "demand" for resources[4], and those that target the "processes or policies" by which the supply of resources is organized and utilized to meet demands[5]. However, in as much as it is straightforward to devise tactical methods

[3]Items 1 – 3 in the list above.
[4]Items 4 – 7 in the list above.
[5]Items 8 – 12 in the list above

by which to solve immediate force generation problems, it is much more difficult to identify the associated strategic consequences over longer time frames. The dynamic nature of MARS is to make it particularly useful in this regard.

1.4 Development of the Managed Readiness Simulator (MARS)

An introduction to the history and phased development approach of the MARS project is the subject of the first report in this series [4]. In short, the MARS project was instigated to develop a tool with a number of functions in mind. They were:

- To provide dynamic gap analysis between supply and demand based on proposed capability demands and existing "Managed Readiness" processes;
- To provide a platform to forecast and understand issues of sustainability in terms of operational capability: resource utilization costs, training and maintenance efforts, and the ultimate quality of fighting forces;
- To facilitate the development and testing of new and different strategies for resource allocation, activity scheduling, unit synchronization, as well as alternate policies for Force Generation;
- To examine the implications of new and existing policies pertaining to human resources and materiel management;
- To assess change options for the structure and organization of the establishment; and
- To facilitate "what if?" analyses.

With respect to the latter, MARS might be used to answer generic questions like those below.

- What will happen if a particular Force Generation strategy is implemented?
- Will this action or strategy have the desired effect over the short term or long term?
- How will this action or strategy promulgate through the interrelated parts of the production system?
- What are the related costs and risks?
- What are the potential mitigation options for unwanted impacts?
- How well do particular mitigation methods work?

To produce a model with these capabilities requires the collection, integration, and simulation of information spanning across the full continuum from Force Development (FD), through Force Generation (FG), to Force Employment (FE). With this in mind an inherent part of MARS development has been to reevaluate the scope and focus of this project on a rolling basis.

The focus of the first phase of development has been to produce an expandable prototype with enough functionality to model how a static[6] Field Force responds to resource demands

[6] In this context the term "static" infers that the demographic within the establishment remains unchanged over time.

for HR training and operational deployments as per the current versions of the MRS and the MRP. Specific attributes and limitations of the prototype will become evident from discussions in sections 2 and 3. The next stages of development are to be jointly decided upon by military partners, such as the Army G3 staff, and the Land Forces Operational Research Team (LFORT) once the first MARS prototype can produce preliminary results that merit confidence and demonstrate the potential benefits of a simulator that is flexible, dynamic, and has an expandable set of diagnostic outputs.

1.4.1 The MARS Architecture

The first MARS prototype is designed to be an integrated application that allows a user to create, store, run, and analyse different FG scenarios. A pictorial illustration of the MARS architecture is presented in Figure 4. Conceptually, it is composed of four main components.

1. A GUI that facilitates the development and management of scenarios;
2. A discrete event simulation kernel;
3. A relational database that stores scenario input and output data;
4. A GUI that provides access to a series of postprocessing routines for analysis and interpretation of simulation results.

To accomplish the aim of this report, discussions are focused around parts 2 and 3 of the MARS architecture, namely the simulation kernel and the relational database. As will be explained in section 1.5, issues with regard to these are discussed together in relation to how they relate to the two fundamental concepts that underpin the design of MARS.

Although this document is intended to focus on the simulation framework and data structure of the first MARS prototype, these aspects are best understood in the context of the whole MARS application. The remainder of this section provides a short overview of MARS as it exists in its current state of development. Fundamental ideas about the data requirements for MARS, Task-Templates, the GUI for user input, discrete event simulation, the relational database, and the analysis interface are described next.

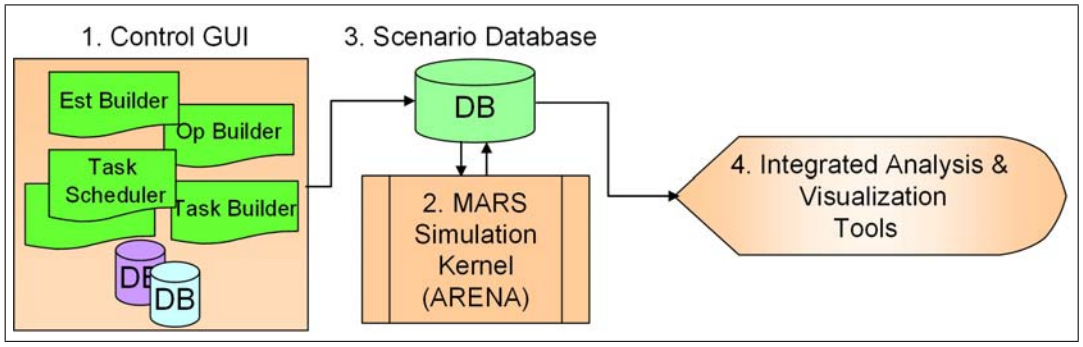


Figure 4 : *Basic illustration of MARS architecture & dataflow.*

1.4.2 MARS Data Requirements

MARS allows for the input of information to model a variety of FG paradigms. The data required to set up a scenario within MARS can be segregated in terms of supply, demand and process inputs.

Supply Inputs

- The hierarchy, position requirements & manning levels within an organization (*e.g.* Army establishment);
- The Whole Fleet stock architecture & Supply levels;
- The Facilities Pools.

Demand Inputs

- A tabulation of important Tasks;
- A breakdown of each Task into smaller Activities;
- The network logic that governs the succession of Activities in each Task;
- The Resource requirements for each Activity.

Process Inputs

- Instantiation schedules for each Task (MRS Inputs); and
- Resource Allocation Schedules/Rules for filling Activity requirements (MRP Inputs).

Figure 5 contains an illustration of how these constructs may be combined within MARS. Shown at the top of the figure are several Task-Templates, each of which is made up of several Activities. Each Task may be generated repeatedly (*i.e.* instantiated) over the duration of a simulation according to an instantiation schedule. Within each Task the incumbent Activities are arranged in a network. Each Activity may require one or more Resources as per a specific set of predefined requirements. Individuals to fill Resource requirements are selected from the establishment, materiel fleets, or facilities pools based on a set of prescribed allocation schedules/rules. Once acquired, Resources may be "passed" between Activities as prescribed via the network logic.

Although this is just a simple description of how MARS operates, it can be used to emphasize some very important concepts that set it apart from other management and analysis tools. For example, the following issues, among others, may be of practical note to military planners.

- Many different Tasks can be scheduled to occur concurrently within a scenario.
- Tasks can be scheduled to occur on a regular deterministic basis or at random.
- The layout of Activities within a Task is allowed to change each time it is instantiated (*e.g.* different “rotos” of the same Task can be created and instantiated.).

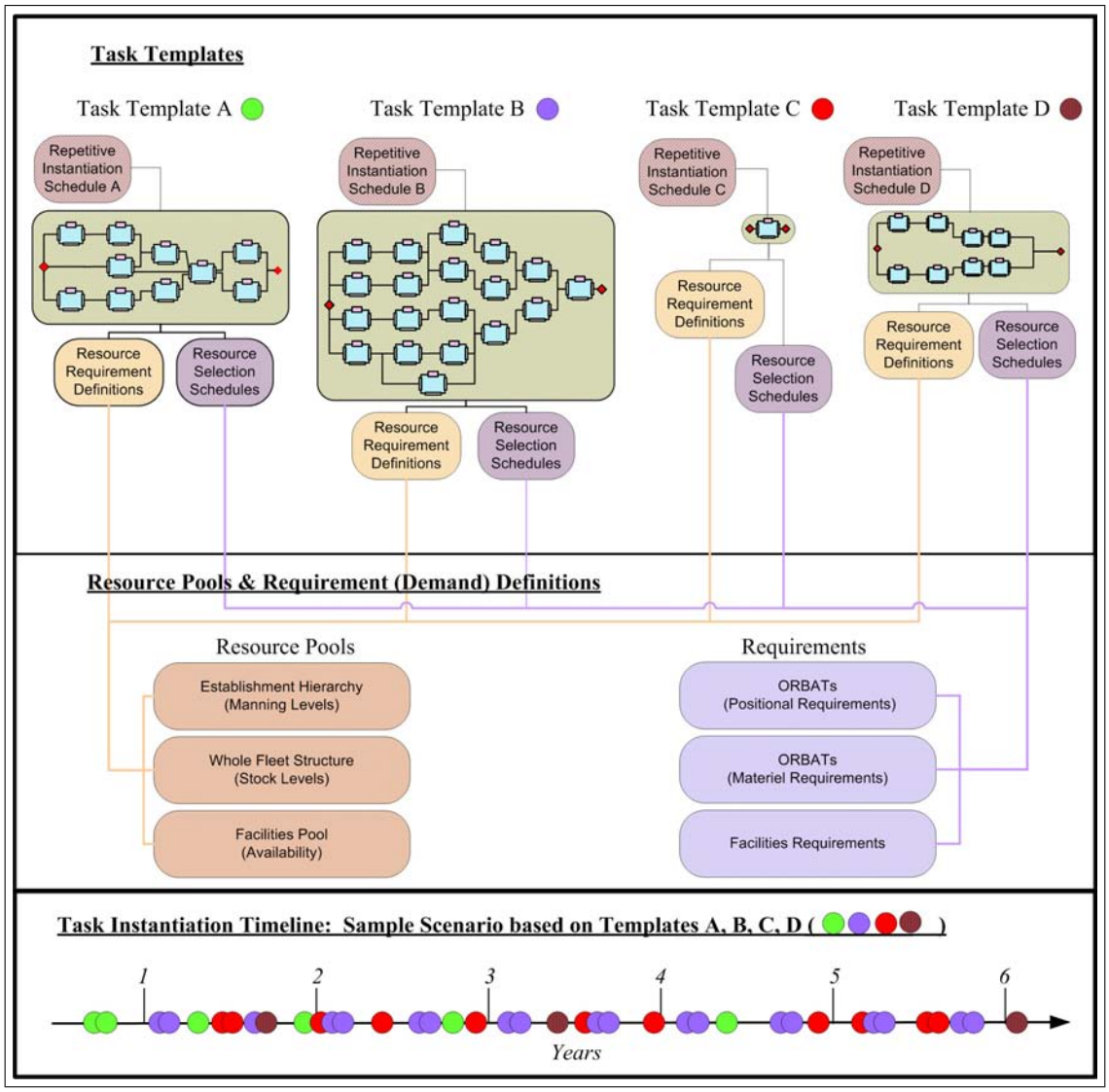


Figure 5 : *An illustration of the constructs that are combined to model Managed Readiness scenarios within MARS. At the top of the figure four Task-Templates are shown. Each of these is designed by the user. A Task-Template may contain any number of Activities that are arranged in terms of an Activity Network. The temporal dependence and flow of Resources between dependent Activities is implicit in the network layout. Task-Templates can be instantiated repeatedly within a scenario based on a user prescribed Instantiation Schedule. Instantiation Schedules can be completely deterministic or stochastic. A sample time-line indicating the occurrence of each instantiation over a 6 yr period is shown at the bottom of the figure. The middle portion of the figure concerns the acquisition of Resources. Each Activity within a Task-Template is linked to a series of Resource Requirement Definitions. These Resource Requirement Definitions contain specific details about the individual position requirements for each Activity. Specifically each Resource Requirement Definition embodies one or more ORBATs. Also linked to each Activity within a Task-Template are a series of Resource Selection Schedules. The parts of the establishment, Whole Fleet, and/or Facilities pools that are responsible for supplying individuals to fill Resource Requirements are laid out in these Resource Selection Schedules. In general, one Resource Selection Schedule must be defined for each Resource Requirement.*

- The Resource requirements within an Activity need not match the organizational structure of any particular part of the establishment (*e.g.* the organizational structure of military forces sent to a theatre of operations do not need to match the organizational structure of units in garrison.).
- The rules used to select the parts of the establishment that are responsible for supplying individuals to fill Resource requirements can change each time a Task/Activity is instantiated.
- The group of individuals within any particular unit, subunit, or lower structural branch within the establishment, can be engaged in more than one Activity at a time (as long as an individual's personal utilization level does not surpass 100%).
- Each individual can be given different attributes over the course of a simulation to indicate eligibility for participation in particular kinds of Tasks or Activities (*e.g.* waivers & Left out of Battle (LOB) attributes can be accounted for).
- Different versions of the establishment with different manning levels can be created and/or imported for use within different scenarios.
- Stochastic aspects (*i.e.* the likelihood that certain events will occur) can be included.

More detail about these is provided in sections 2 and 3.

1.4.3 The MARS Graphical User Interface (GUI)

The GUI is composed of several categorical components that are accessible through what amounts to a main control panel. The layout of the main control panel is illustrated in Figure 6. Along the left hand side of the panel is a tabulation of scenarios that have been created. Different icons are used to indicate those scenarios that have been run through a simulation and those that have not. Information about a selected scenario is presented in the centre of the main control panel. Along the bottom of the panel are controls that allow the user to run a scenario through the simulator. On the right side of the panel are buttons that provide the user access to integrated utilities for creating and modifying a scenario. Each of these are described below.

Resources/Organizations Button group : The Resources/Organizations button group in Figure 6 contains utilities to create and organize Resource pools and Resource Requirements (*e.g.* ORBATs for deployment to theatre). It contains 2 functional buttons: the Organization Builder and Personnel[7].

- The Organization Builder contains two sub-utilities, the Establishment Builder and the ORBAT builder. A screen-shot of the Establishment Builder Utility is shown in Figure 7. In this figure the hierarchy of the Field Force is shown on the left hand panel. On the right hand panels are shown specific details about the positions contained in selected Units (in this case, `LFAA/2RCR/ACOY/1PL`). A

[7]The "New Org Builder" button is in development and at the time of writing is not functional.

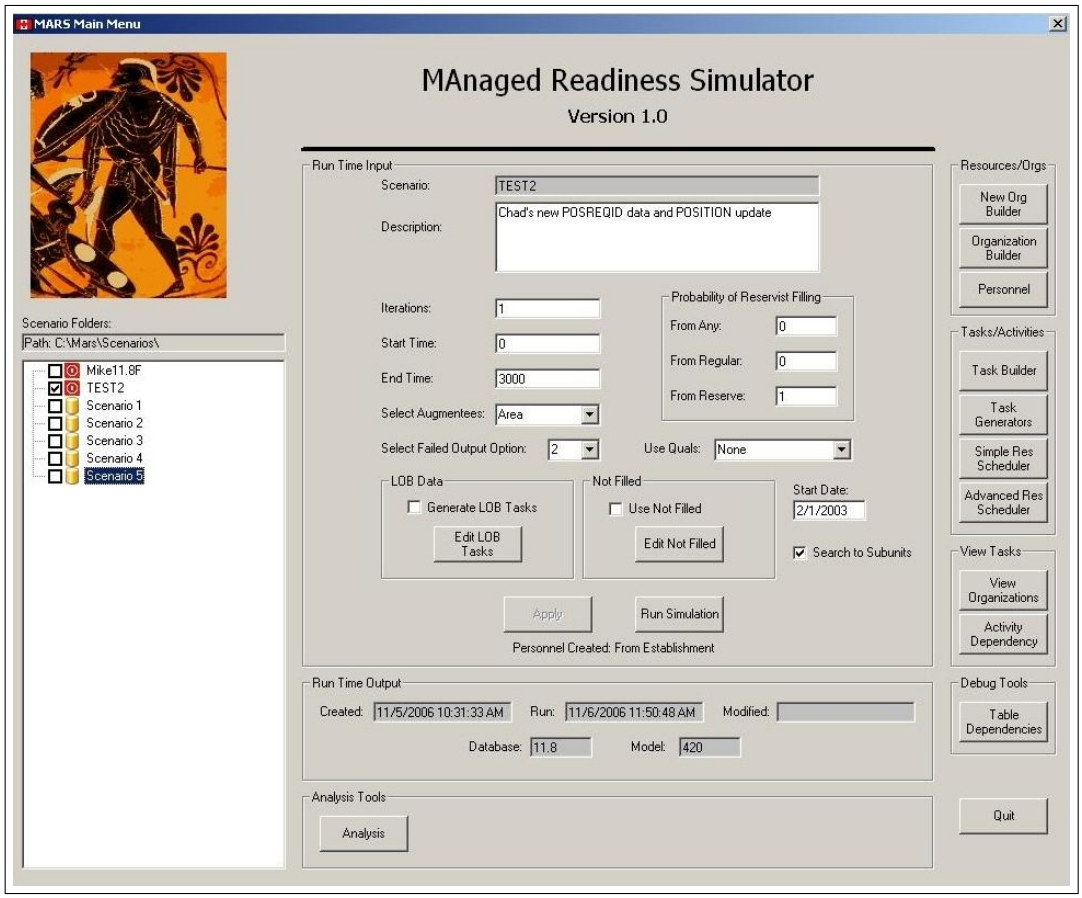


Figure 6 : *The main MARS Control Panel Interface. Aspects of this interface are described in section 1.4.2.*

screen-shot of the ORBAT builder is shown in Figure 8. In this figure the hierarchical structure of different ORBATs are displayed on the left hand panel. On the right hand panels specific details about selected parts of the ORBAT are depicted. In this example is shown position information about `MSTF/SURV SQN/HVY SURV TP/1 HVY RECCE PLT/1 HVY RECCE DET`. Of course, these ORBATs can be altered to test different concepts within a MARS scenario. Note that in Figure 8, the designation `417:SurvTp Comd` is a position requirement label. This label represents a link to a set of Rank, MOC, and Qualification requirements. Other MARS utilities enable the user to create these requirements.

- The Personnel button provides access to a utility that allows the user to connect personnel to certain positions within the establishment[8]. In this regard, the first MARS prototype provides two options. The first option allows the user to "create" individual personnel for every position within the establishment. The second option allows the user to import personnel information from an external data source. The first option can be used to represent a "best case scenario" where the establishment is completely manned with no shortfalls or excess. The second option can be used to capture actual manning levels. As MARS is developed, options that allow the user to create some individuals and import information about others are to be incorporated. Indeed, accurately accounting for manning levels is an important aspect of any MARS simulation.

Tasks/Activities Button group The Tasks/Activities button group provides access to utilities that allow the user to create Task-Templates, Task Instantiation Schedules, Activities, and Resource Selection Schedules/Rules. There are four buttons: Task Builder, Task Generators, Simple Resource Scheduler and Advanced Resource Scheduler.

- The Task Builder Button opens a utility that allows the user to create Task-Templates. An illustrative screen-shot of this utility is shown in Figure 9. On the left hand panel are individual Task-Templates. Each Task-Template can be broken down hierarchically into individual Activities. The hierarchical structure of a Task-Template is described in section 2. Highlighted in Figure 9 is an Activity called `CMTC Training`. Details about this Activity are presented in the right hand panel. Discussions about Activity attributes are presented in section 2.
- The Task Generator Button opens a utility that allows the user to create Task Instantiation Schedules. As will be described in section 2, these instantiation schedules may contain random and deterministic components.
- The Basic Resource Scheduler allows the user to schedule units as per a particular MRP. By creating different schedules, MRP options can be analysed. An illustrative screen-shot of this utility is presented in Figure 10.
- The Advanced Resource Scheduler provides enhanced capabilities that are not available with the Basic Resource Scheduler. Among other things, the Advanced Resource Scheduler allows for the scheduling of individual SubUnits within the establishment. The layout of this utility is illustrated in Figure 11.

[8] In MARS, each establishment position *can* contain (or be attached to) more than one individual if so desired.

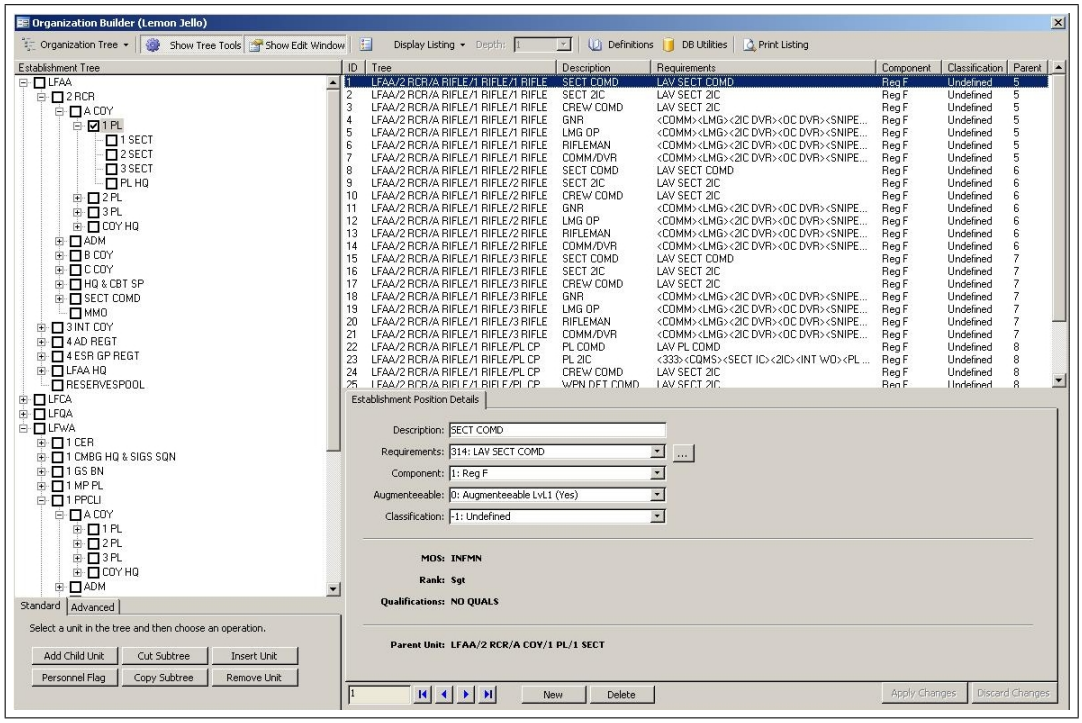


Figure 7 : *The Establishment Builder Interface. The MARS prototype interface for creating and modifying the hierarchy of Formations, Units, SubUnits, and Positions within the establishment. On the left hand panel is the establishment breakdown. On the right hand panel are specific details about a selected Unit within the establishment. In the case shown here, details about the list of positions within* `LFAA/2RCR/ACOY/1PL` *are shown. When running a MARS scenario, different parts of the establishment can be marked as available or unavailable. Additionally, as MARS is developed, the ability to model the transformation of the establishment by setting "expiry dates" on some branches and "installation dates" on other branches is to be investigated.*

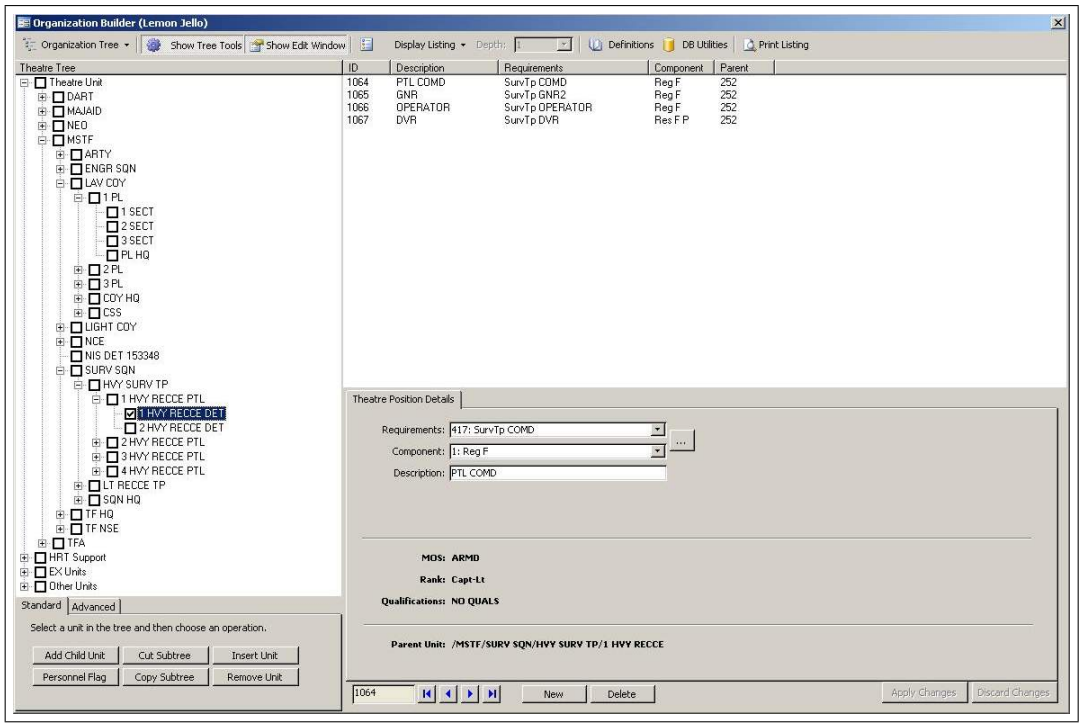


Figure 8 : *The ORBAT Builder Interface. The MARS interface for creating and modifying ORBATs. On the left hand panel are listed several ORBATs that may be expanded to show specific details on the right hand panel. These ORBATs, or branches within each ORBAT, are used to define the Resource Requirements for individual Activities in MARS. The user can create any number of different ORBATs for use within a particular MARS scenario.*

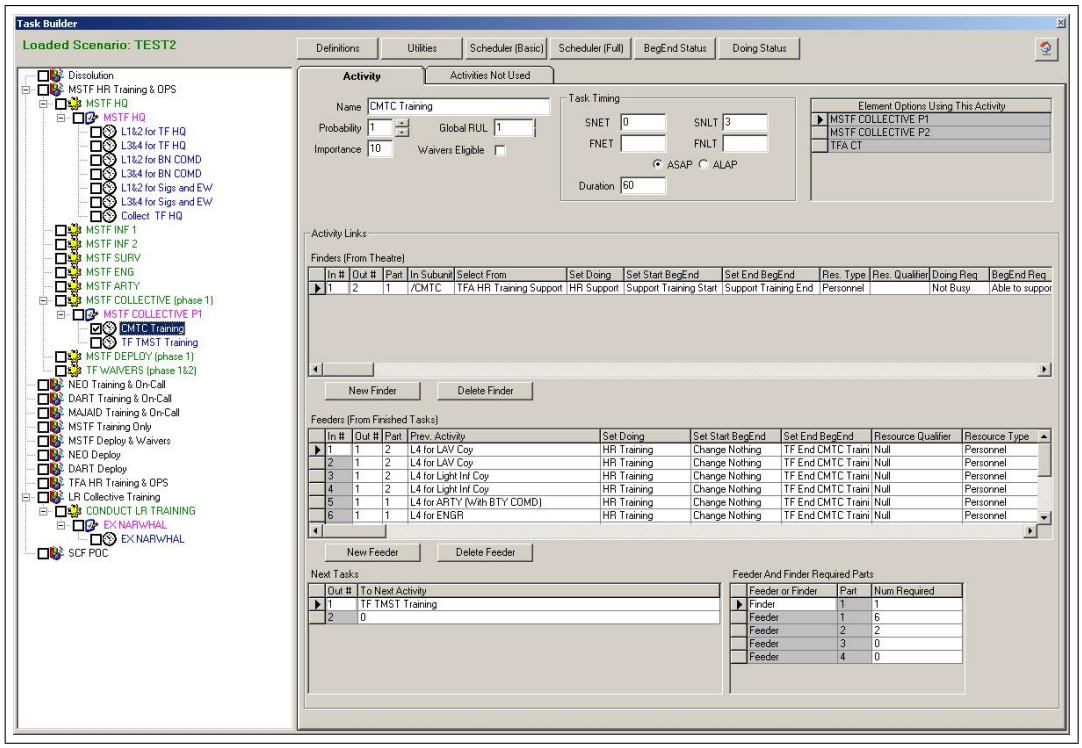


Figure 9 : *The Task Builder Interface, the MARS prototype interface for creating and modifying Task-Templates for use within a MARS scenario. On the left hand panel is a listing of individual Task-Templates. Specific examples in this list are Task-Templates labeled:* `MSTF HR Training & OPS`, `NEO training & On-Call`, *and* `DART Training & On-Call`. *Each Task-Template can be broken down hierarchically into individual Activities. The hierarchical structure of a Task-Template is described in section 2. Highlighted in the figure above is an Activity within the* `MSTF HR Training & OPS` *Task-Template called* `CMTC Training`. *Details about this Activity are presented in the right hand panel. Discussions about Activity Attributes are presented in section 2*

.

View Tasks Button group Referring back to Figure 6, the two buttons labeled View Organizations and Activity Dependency open "read-only" utilities that allow the user to check and collectively assess a scenario before setting it to run.

Debug Tools Button Group The button(s) in this group open utilities that are specifically for algorithmic debugging. As MARS become more mature, these utilities will not be presented to the user as part of the main control panel.

Once the design of a scenario is complete, the buttons along the bottom of the main panel in Figure 6 allow the user to run (or re-run) a scenario and analyse results using a series of graphical analysis utilities. The publication of documentation to instruct users on the MARS GUI is to follow upon development of future versions.

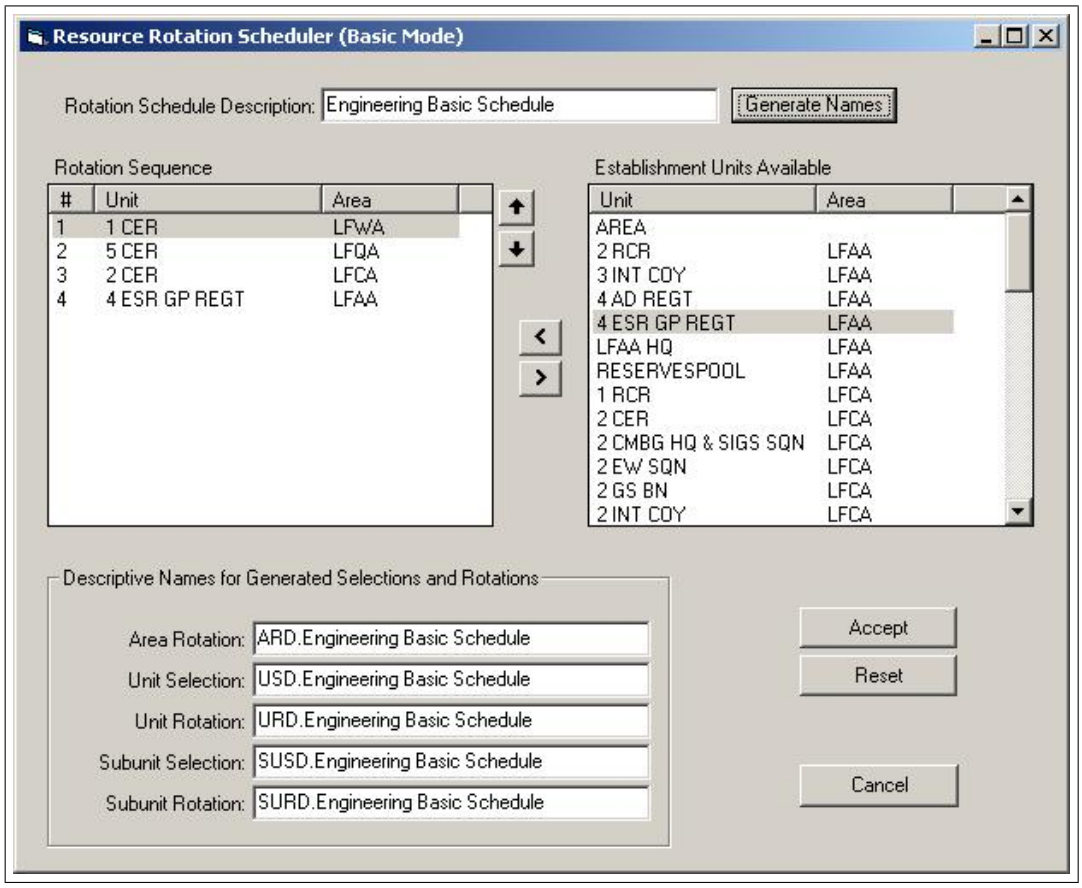


Figure 10 : *The Basic Resource Scheduler interface. In this example the schedule for a particular Engineering resource is to cycle through four Units in a Rotation. As shown, the rotation sequence is:* `1 CER`, `5 CER`, `2 CER`, *and* `4 ESR GP RGT`. *Sequences like this can be designed to coincide with a particular MRP. Resource Schedules are explained in section 2.5.3.*

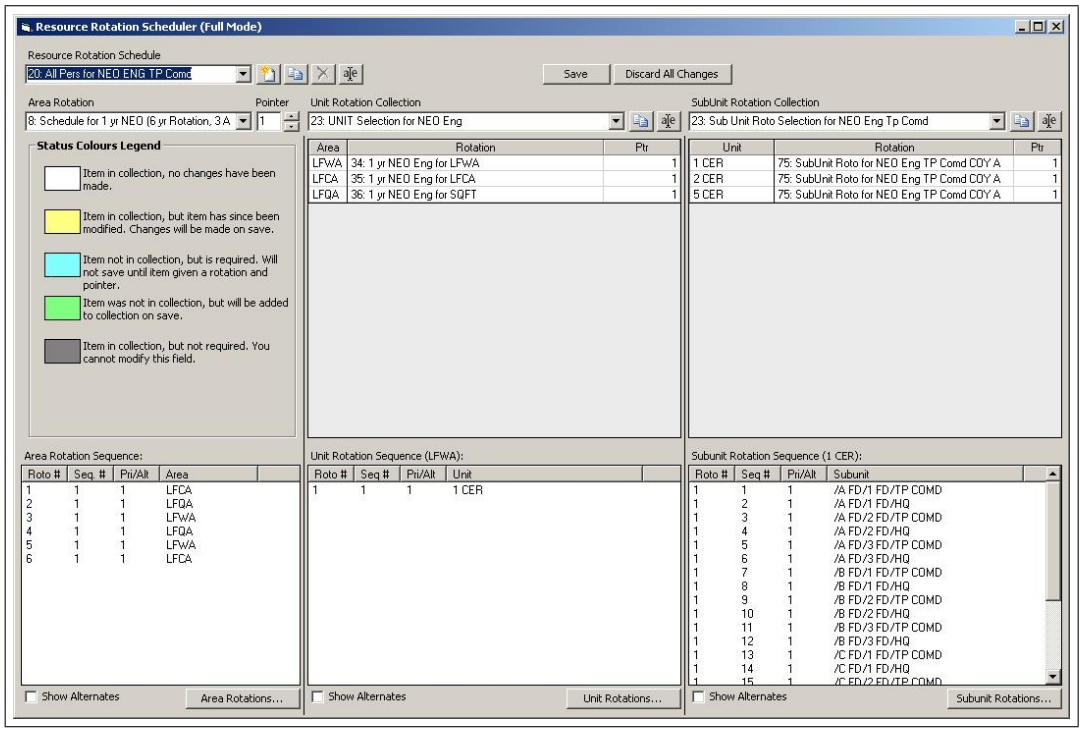


Figure 11 : *The MARS prototype interface for creating Advanced Resource Schedules. As with the Basic Resource Scheduler, the Advanced Resource Scheduler can be used to create schedules that coincide with any MRP. However, whereas the Basic Resource Scheduler is designed to allow the user to designate Unit Rotations, the Advanced Scheduler is designed to allow the user to create SubUnit schedules. This utility is in the beginning stages of development. Note that the window contains 6 panels that are organized into 3 columns. The first column contains information about Area scheduling, the second column contains information about Unit Scheduling, and the third column contains information about SubUnit Scheduling. It is expected that a user would use the Basic Resource Scheduler to create Area and Unit schedules and then use the Advanced Resource Scheduler to create SubUnit schedules. Concepts behind the creation of schedules are explained in section 2.5.3.*

1.4.4 The Simulation Kernel

As depicted in Figure 4, the simulation kernel is the computational engine of MARS. It is built as a discrete event simulation in the *Arena v10.0* simulation environment. Brief introductions to some of the general ideas behind discrete event simulation have been described in Pall *et. al.* [5] and Isbrandt & Zegers [7]. A comprehensive reference on the subject has been written by Law & Kelton [8]. The strategy behind the design of the kernel is the subject of chapters 2 and 3. Algorithmic implementation details pertaining to the MARS kernel are presented in the third report of this companion series [5].

1.4.5 The Relational Database

As is evident from the list of data requirements in section 1.4.2, the creation of a scenario within MARS requires a large amount of information – from the definition of resource supplies, to the specification and scheduling of demands, and the prescription of Tasks and Activities. In the first MARS prototype all of this information is stored within a relational database that is implemented inside Microsoft Access. A new database is created each time a new scenario is created by the user. Upon creation, certain tables inside the database are cleared and re-populated with input prescribed through the GUI.

The relational database also provides the framework for organizing and storing the collection of output data generated by the simulation kernel. The collection of scenario input and simulation output within the relational database provides the foundation upon which routines within MARS can provide for the automatic analysis of scenario results.

In its implementation, there are a series of overarching algorithms that control the interaction of the simulation kernel and the relational database. For the first MARS prototype these algorithms operate to:

1. Export certain data from the relational database to intermediate unformatted text files,
2. Prompt the start of the simulation kernel and pass information about the location of the text file input, and
3. Import the output from the simulation kernel to the database.

Section 3 contains additional detail about the tabular structure of the relational database.

1.4.6 The Analysis Interface

The analysis interface consists of a series of graphical tools to synthesize the results of a simulation. The bulk of the analysis routines behind these tools are encoded in the Microsoft Visual Basic programming language to operate on data inside the relational database. The culmination of these routines are a series of graphical and tabular outputs that present scenario forecasts from a variety of different perspectives. Figures 12 through 14 are examples of analysis details that are obtainable with the first MARS prototype.

Illustrated in Figure 12 are sample results from an example (non-realistic) scenario. They are presented in a way that might be of greatest relevance to a force employer. On the left of the interface the user can select a series of Tasks, Activities, and/or Resource Requirements for which the analysis is to be carried out. For each of these, a corresponding stacked bar is shown in the chart on the right. In a general sense, the ability of the Army to fill a prescribed set of Resource demands is most easily assessed with this kind of plot. Each bar indicates the number of positions filled from scheduled primary units, alternate units, and reserve units. Also illustrated are those positions that remain unfilled when the resource acquisition schedules prescribed for the scenario are used. Although not shown here, a break-down of information about the positions that remain unfilled or are filled by augmentees or reserves can also be obtained via the GUI. The presentation of the bar graph in Figure 12, as with most others, can be altered and exported to other applications.

Illustrated in Figure 13 are sample results from an illustrative scenario that are presented from the perspective of a Force Generator. As shown, MARS allows the user to navigate through the establishment and select particular formations, units, subunits, etc. Also selectable are the Task types, Activity types[9], Rank Ranges, and Occupation designations for which the analysis is to be carried out. The graph shown in Figure 13 indicates the number of persons from a particular part of the establishment that are occupied with the preselected types of Tasks or Activities over a selected time period during the simulation. Also shown on the graph is the upper bound on the number of individuals with the selected attributes that occupy positions in the selected location of the establishment over the selected time frame. By comparing the number of individuals in a particular Unit with the number of individuals that are actually busy at any given instant in time, the efficiency with which resources are utilized can be ascertained.

Depicted in Figure 14 is a Gantt chart that, in addition to being of importance to a force generator, is expected to be of particular interest to specific Formations/Units within the establishment. It indicates the occurrence of specific Activities that occupy their personnel. Time is depicted along the horizontal axis of the chart and specific Activities are indicated along the vertical axis of the chart. Although not shown here, the number of personnel participating in each Activity can be obtained interactively from the GUI. Tools that indicate how individuals or groups of individuals move from Activity to Activity are also available.

Of course, these are just a sampling of some of the kinds of outputs that are available from the first MARS prototype. From a technical viewpoint, none of the tabulated outputs are especially new. They represent simple data summaries. The power of MARS is that the user is provided the ability to perform analysis by filtering and viewing results at various levels of aggregation – from the analysis of pooled resources to that of an individual or small groups of individuals. It is well known that "the devil is in the details" when it comes to resource utilization and capacity issues and the ability to "drill down" through various levels of detail is expected to be helpful for diagnosing particular issues that may not be apparent when aggregated at a high level.

[9]Note that the user is provided the ability to create different Task/Activity types when defining a scenario.

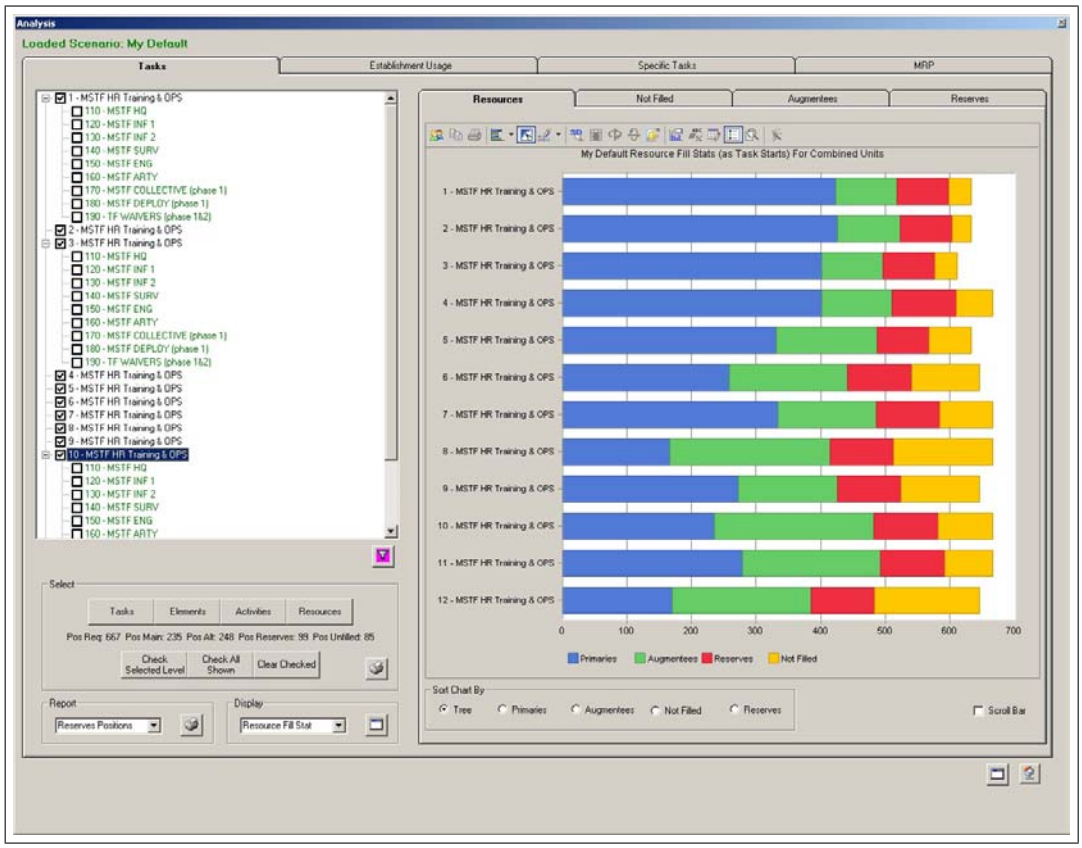


Figure 12 : *Illustrative MARS Output (Force Employer Perspective) for an example scenario. Resource fill statistics are presented for selected Tasks over the course of a simulation. On the stacked bar plot is shown the population of individuals filling Resource demands that were filled from scheduled Units, Backup Units (also known as augmentee Units) and Reserve Units. Also shown are the proportion of positions that remain unfilled based on the scheduling and acquisition rules prescribed by the user. A series of other analysis tools that allow the user to analyse results in more detail are also available.*

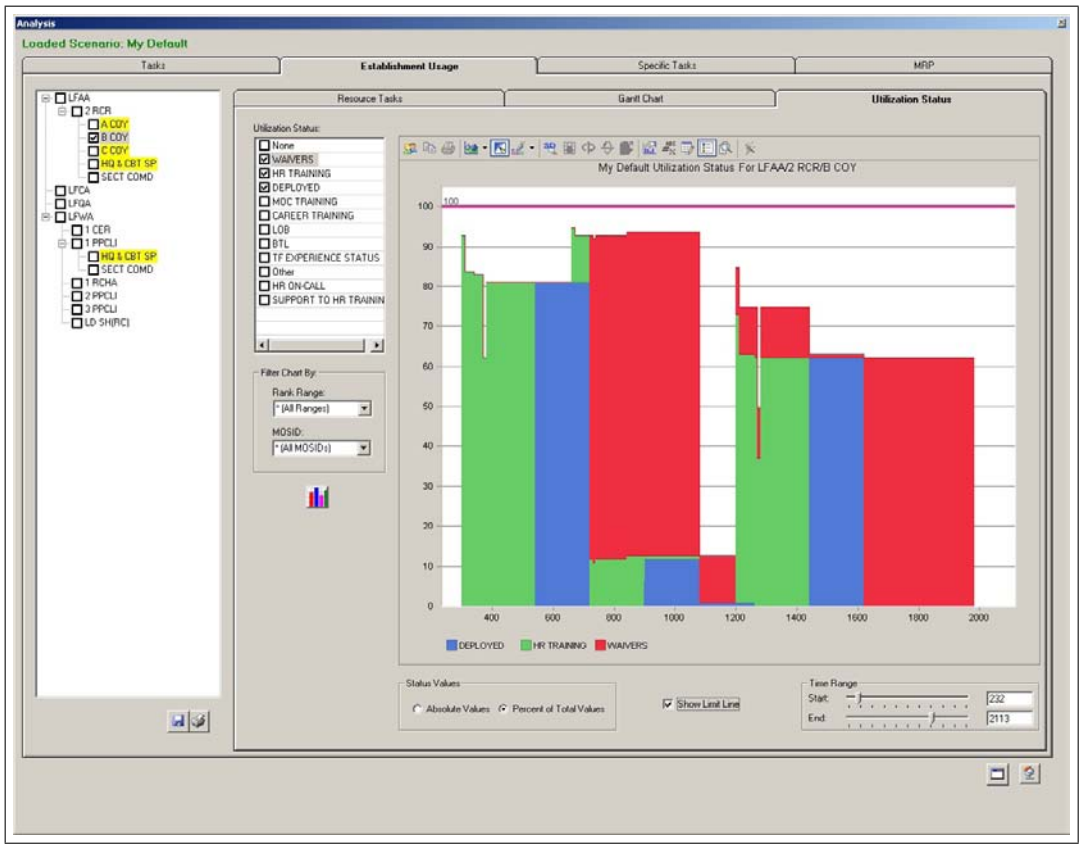


Figure 13 : *Illustrative MARS Output (Force Generator Perspective) for an example scenario. The stacked line graph with area shading indicates the proportion of personnel from selected units within the establishment that are occupied with certain kinds of Tasks/Activities over a selected time period. A series of other analysis tools that allow the user to analyse these results in more detail are also available.*

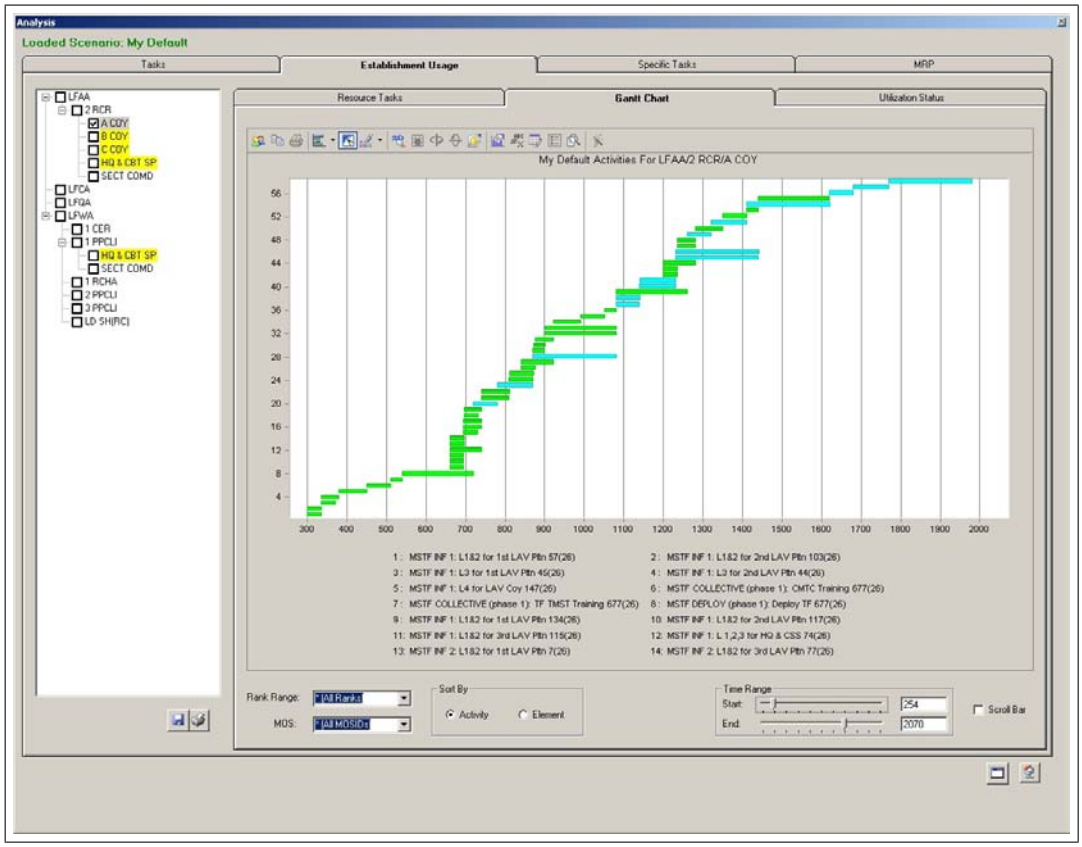


Figure 14 : *Illustrative MARS Output (Unit Perspective). The Gantt chart provides details about when individuals within a particular establishment unit are occupied with particular Activities. This utility is interactive. Information (not shown here) can also be obtained about the number of individuals participating in a specific Activity.*

Since the focus of development for the first MARS prototype has been the development of a “proof of concept” framework, it is important to note that the family of analysis tools currently incorporated in MARS is only a sampling of what is possible. MARS is designed to integrate data that might not otherwise be integrated with other analysis tools, and therefore it is expected to have appeal to a variety of potential stakeholders across the Canadian Forces. As such, results are compiled from a variety of perspectives. The Force Employment, Force Generation, and Unit perspectives noted above are examples. As MARS is developed, a wider family of analysis tools are to be incorporated[10]. The preparation of documentation regarding the MARS analysis interface is to follow the development of future versions of the MARS.

1.5 MARS Kernel Design: Two Fundamental Design Concepts

The kernel of MARS is built on two levels of abstraction. Both levels have a common theme – each is based upon the idea that a particular type of "entity" is considered to flow between one or more "stations" which are arranged as part of a larger causal "network". What makes the levels of abstraction different are the items that these "entities", "stations", and "networks" represent.

1.5.1 The Resource-Activity-Task Paradigm

The first level of abstraction provides a framework for understanding the inputs required to create a scenario within MARS. It is therefore of particular importance to a prospective MARS user. On this level "entities" are represented by "Resources", "stations" are represented by "Activities", and the "networks" that define the causal dependence of Activities are called "Tasks".

In practical terms, as alluded to above, each Resource is defined to represent a group of individual people, materiel or facilities. Within a simulation, Resources are drawn from different parts of the establishment, equipment fleet or facilities pool to participate in one or more Activities. Each Activity serves to achieve a small part of the collective objective of a large Task. A sample illustration of this **Resource-Activity-Task** paradigm is presented in Figure 15 where it is shown that several Resources simultaneously flow through a network of Activities that compose a larger Task.

Many different Tasks may be defined to occur concurrently. In addition, each Task may be instantiated repeatedly over the course of a simulation. An example of a large Task might be that of the "High Readiness Training" regimen that precedes deployment to a theatre of operation. According to the current version of the MRS, this Task is to take approximately 12 months and is to be instantiated repeatedly every 6 months. A full discussion outlining the details of the first level of abstraction is the subject of section 2.

[10]It is expected that the second version of MARS shall strive to incorporate specific or general tools to model/analyse the interlinked dynamic effects of individual training, recruitment, promotions, postings, attrition, and retirement.

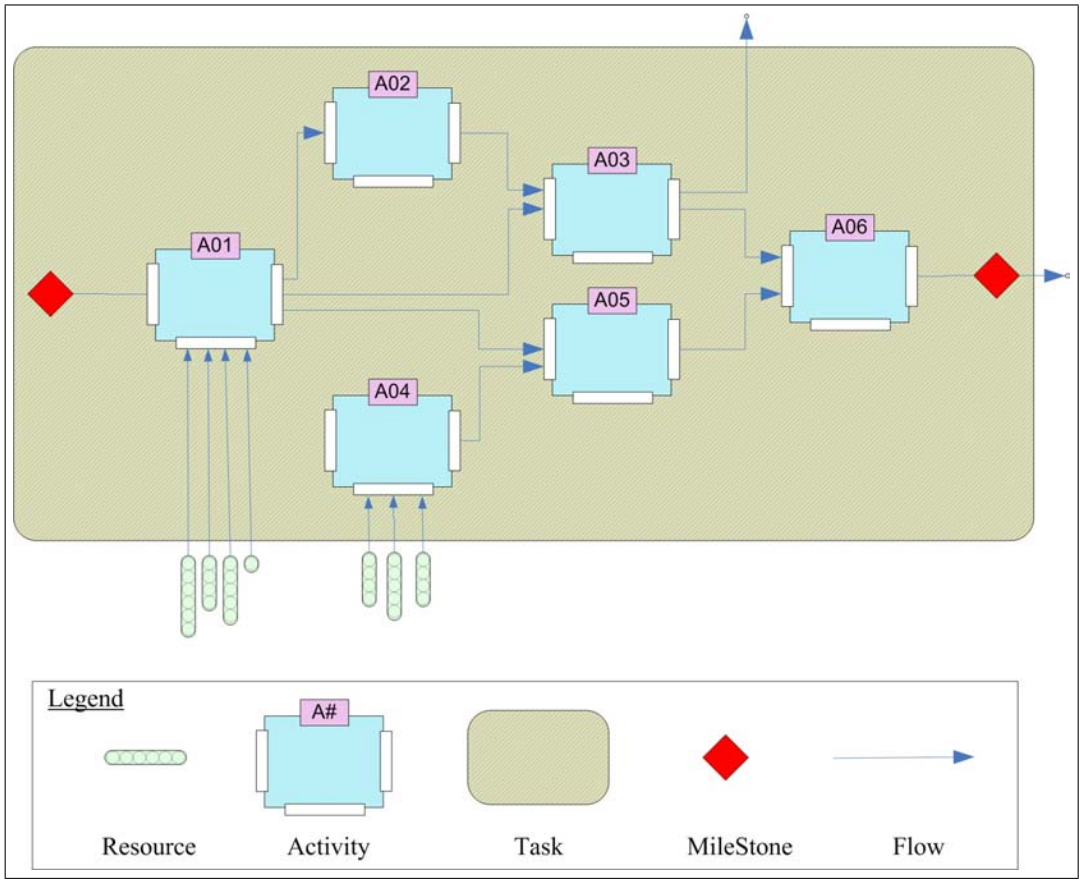


Figure 15 : *A network of Activities. Resources "flow" between Activities in the network. In the Resource-Activity-Task Paradigm the "entities" are Resources, the "stations" are the Activities, and the "network" is the Task. This paradigm is the subject of section 2.*

1.5.2 The Activity-Station-Lifecycle Paradigm

The second level of abstraction within MARS encapsulates the machinery for how it instantiates Tasks and models the acquisition and flow of Resources between Activities. On this level of abstraction, which is of particular relevance to the analyst, "entities" are "Activities", that flow between "stations" within a "lifecycle". This **Activity-Station-Lifecycle** concept is illustrated in Figure 16.

Activities are created at the beginning of the lifecycle, and at each progressive station certain algorithmic functions are executed. Functional specifics at each station in the lifecycle depend on the attributes of the Activity itself. During the tenure of an Activity in the lifecycle, the status and/or attributes of personnel, materiel, and/or facilities that participate in an Activity may be altered. These status changes can impact upon how individuals within each Resource may be assigned to concurrent and future Activities. At the end of the lifecycle

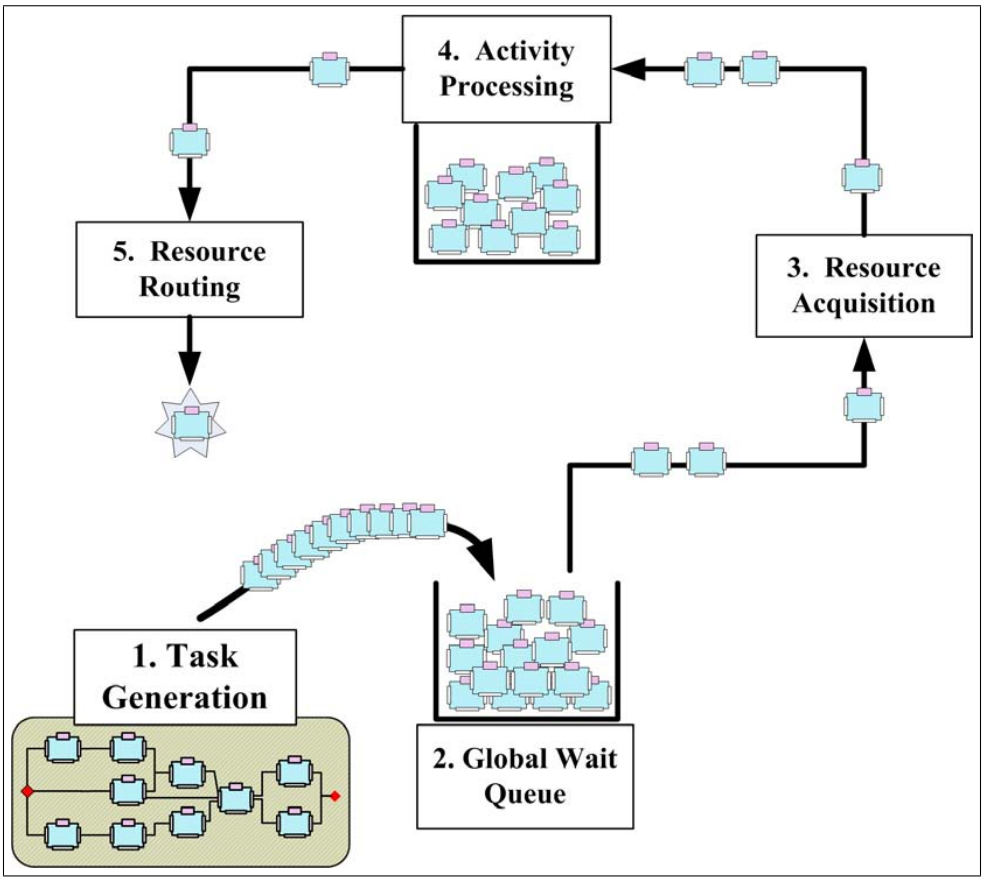


Figure 16 : *A Pictorial View of the Activity Lifecycle. Activities "flow" from station to station. In this lifecycle "entities" are represented by Activities that flow between the numbered stations shown above. A detailed discussion of the Activity-Station-Lifecycle paradigm is presented in section 3.*

the Activity is dissolved.

An important aspect of the lifecycle is how individual personnel, material and facilities are acquired to fulfil the Resource requirements of each Activity. Simulating the acquisition of Resources ultimately allows MARS to forecast Resource utilization levels and predict the Army's ability to complete scheduled Tasks. In fact, predicting the Army's ability to complete Tasks and dynamically determining the utilization of personnel, materiel, and facilities across the Army, are the fundamental strategies by which the sustainability of managed readiness alternatives can be predicted with MARS. A full discussion of this topic is presented in section 3.

2 Tasks, Activities & Resources

As introduced in section 1.5.1 and illustrated in Figures 15 and 16, MARS is designed around two levels of abstraction. The first level of abstraction embodies a Resource-Activity-Task paradigm and is the subject of this section. In the Resource-Activity-Task paradigm the term "Resource" refers to a collection of personnel, materiel, or facilities. Resources are considered to flow between "Activities". These Activities represent smaller processes within a larger network that defines a "Task". This section is expected to be of greatest interest to a prospective MARS user because it characterizes the way that different Managed Readiness concepts may be input for testing within MARS.

2.1 Defining a Task-Template - An Overview

MARS is designed to provide the user with tools to define "Task-Templates"[11]. Once defined, a Task-Template is used within the MARS simulation kernel to repeatedly create individual instances of a Task based on a user defined Task Instantiation schedule. This idea is illustrated in Figures 5 and 17. Using the current MRS as a reference, the network of Activities contained within the HR Training regimen is one example of what can be defined using a Task-Template. Given the nominal frequency and duration for operational deployments, a Task-Template designed to represent HR training could be instantiated twice every six months as per a user prescribed Task instantiation schedule. Other current Canadian Army examples of what might be instantiated through the design of a Task-Template include the training and/or deployment of: a Non-combat Evacuation Operation team (NEO), a Disaster Assistance Response Team (DART), or a Special Contingency Force (SCF). Each of these serves to *produce* or *exercise* a *capability* to provide a *service*. In fact, this is a useful and practical definition of what a Task-Template represents[12].

The Activities embodied in a Task-Template are the vehicles by which MARS simulates the selection and processing of resources. Specifically, each Activity encapsulates defining particulars about required resources and how these resources are to be received, collected and forwarded to dependent Activities. Resources are categorized into three types: personnel, materiel, and facilities. Different than other models where a Resource represents an individual person, piece of equipment, or facility, the resources within MARS may represent collections of personnel, materiel or facilities. Several resources may participate in a single Activity as per a set of Resource requirements.

Resource requirements are part of the definition of an Activity and, as each Resource may contain a collection of individuals, each Resource requirement may contain a collection of position requirements. For example, a personnel Resource requirement can represent the need to fill a group of positions in a Unit where each position has a defining set of

[11] Task-Templates are very similiar to classes within the object-oriented approach to computer programming.

[12] MARS is not limited to modeling the large Tasks typically associated with the MRS. It is designed to be equally adept at modeling smaller Tasks that include only a single Activity. In fact, by modeling large collective Tasks in concert with smaller Tasks a better understanding of the overarching impacts of potential scheduling conflicts and resource utilization policies can be obtained.

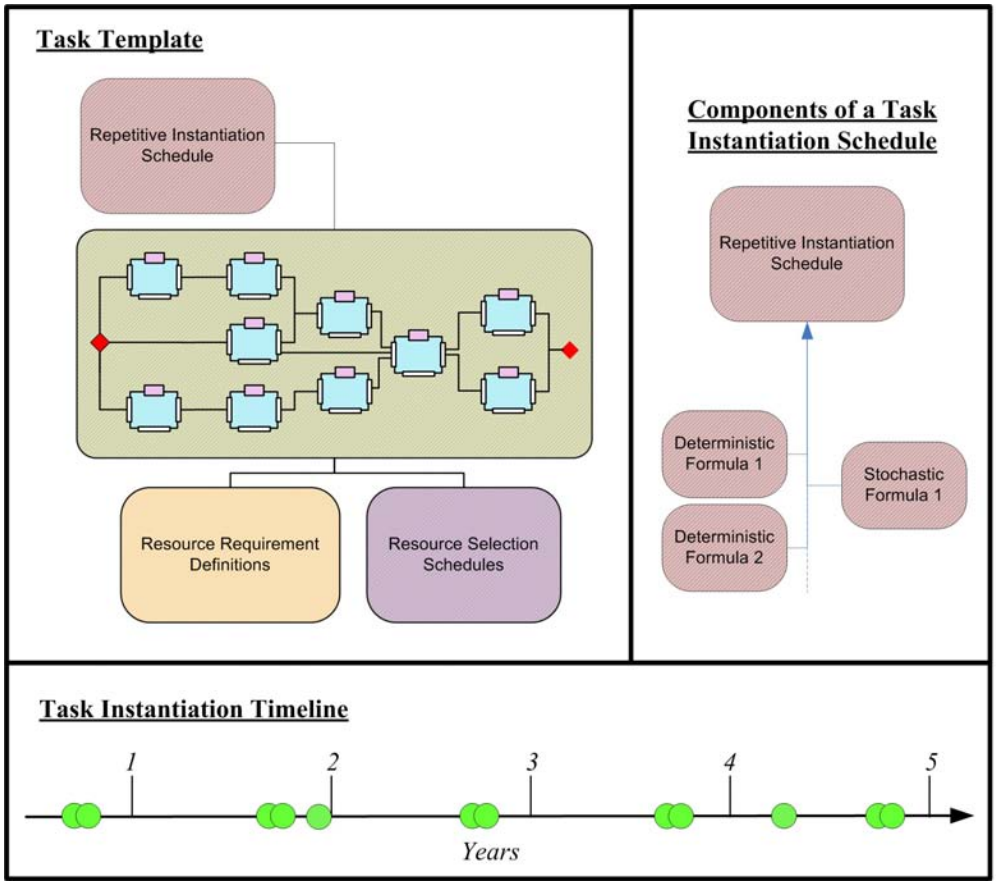


Figure 17 : *A pictorial of a Task-Template, Task Instantiation Schedule, and a sample Instantiation Timeline. Note that Resource Requirement Definitions and Resource Selection Schedules are actually connected to individual Activities however, for the sake of clarity, they are collectively shown to be connected to the Task-Template.*

prerequisite criteria[13]. These criteria are used to validate whether individuals can become part of a Resource.

Also encapsulated within each Activity are a set of "Resource Selection Schedules". A single Resource Selection Schedule is linked to each Resource requirement and each selection schedule prescribes the location(s) within the establishment, equipment fleet and/or infrastructure pool where resources are to be selected to fulfill a Resource requirement[14]. These Resource Selection Schedules play an important role within MARS. In a practical sense, the combination of Resource Selection Schedules and Task instantiation schedules determine "who" is required to do "what" at a particular time within a simulation.

Once an Activity is complete, participating Resources may be routed to dependent Activities in the network. From an Activity viewpoint, the routing of Resources occurs on two conceptual levels: internal routing and external routing. The routing that occurs "internal to" an Activity serves to effectively group Resources that are to proceed together to the same subsequent Activity in the network. The routing that occurs "external to" an Activity serves to demonstrate the path of each grouped Resource from the termination of one Activity to the commencement of the next. It is this external routing that indicates how Activities are "connected" to form the network that constitutes a Task-Template. The collective success with which Resources are created, participate, and flow between Activities, is the basic measure for determining the viability of a scenario.

From the perspective of a user, it is convenient to think of MARS as a framework by which the user creates Task-Templates to identify the requirement, scheduling, and processing of resources (*i.e.*, personnel, materiel, and facilities) to produce or exercise a capability. The Task-Template concept is fundamental to the creation of a simulation scenario within MARS. With few exceptions, the rest of discussions in this section elaborate upon topics that are important for the creation of a Task-Template. These include the Task instantiation schedule, the Activity network hierarchy, Element-Option schedules, Activity processing attributes, resource pools, Resource requirements, and Resource Selection Schedules. Included in these discussions are references to section 3 where the relational database structure, used to store Task-Template information, is described.

2.2 The Task Instantiation Schedule

Task-Templates may be instantiated repeatedly and concurrently, as is the case in the "real world", in order to simulate the ability of the armed forces to fulfil a broad range of demands over a predefined time horizon. To accomplish this within a simulation requires the definition of what are called "Task instantiation schedules". These schedules may be deterministic or stochastic in nature. To define a schedule, the user is allowed to select from a predefined collection of instantiation formulae and then set associated parameters. As an example, the instantiation schedule for an HR training Task might take the form of a deter-

[13]It is equally plausible that a Resource requirement may represent the need to fill a single position with a person that has a particular set of rank, occupation, and status attributes.

[14]Individuals already assigned to other Activities may also be reallocated to fulfill Resource requirements.

ministic (*i.e.* arithmetic) sequence. This sequence could be designed to mimic the periodic regeneration of readiness capability, as per the current MRS. In another example, a Non-Combat Evacuation Operation (NEO) could be set to occur on random intervals according to any one of several probability distributions.

In most cases it is expected that an instantiation schedule will be composed of a single formula, however in cases where this is not adequate, MARS can accommodate the creation of a schedule from a composite of several instantiation formulae. For example, in the case where an MRP such as that shown in Figure 3 specifies the need for several “lines of operation”, a single Task-Template may be employed with two arithmetic instantiation schedules, each out of phase with each other, according to a prescribed delay. This concept is illustrated in Figure 17.

The ability to create Task-Templates that may be repetitively instantiated within a simulation is one aspect that sets MARS apart from many static project management tools. Different Task-Template and Instantiation schedule combinations may be created to compare and assess various managed readiness concepts. The relational data structure used to encode Task instantiation schedules is briefly described in section 3.3.

2.3 Elements, Element-Options & Element-Option Schedules

The ability to define a Task-Template and use it repeatedly is of great utility and convenience because the user does not need to redefine a Task each time it is to occur. However, in practice, the likelihood that two instances of a Task have identical Resource requirements or Activity networks is small. As a case in point, the HR training Task which serves to prepare a Mission Specific Task Force for operations may take on a slightly different form depending on the number and makeup of infantry, artillery, armor, and engineering units that need to be employed. Because of these differences, some Activities may need to be added, removed, or altered between instantiations. To account for the potential alteration of Activities and Resource requirements it becomes necessary to define different "variations" of a Task-Template and instantiate these according to a particular sequence in a MARS scenario. This functionality is implicitly accommodated by the way a Task-Template is defined within MARS.

To accommodate variations in the definition of a Task-Template a tri-level hierarchy is adopted to define the constituent parts of a Task-Template within MARS. The use of a hierarchy to break down a Task into smaller parts is consistent with the way that a Work Breakdown Structure (WBS) is used to define a project within a project management milieu [9]. On the bottom level of the WBS hierarchy are the individual Activities already mentioned above. Individual Activities are grouped together to form "Elements" which in-turn are grouped to represent the entire Task. This concept is illustrated in Figure 18.

The "Element" level in the hierarchy provides an intermediate degree of Activity aggregation that provides a degree of generic flexibility when defining a Task-Template. In practice, as noted above, one may wish to create several variations of a Task by making changes

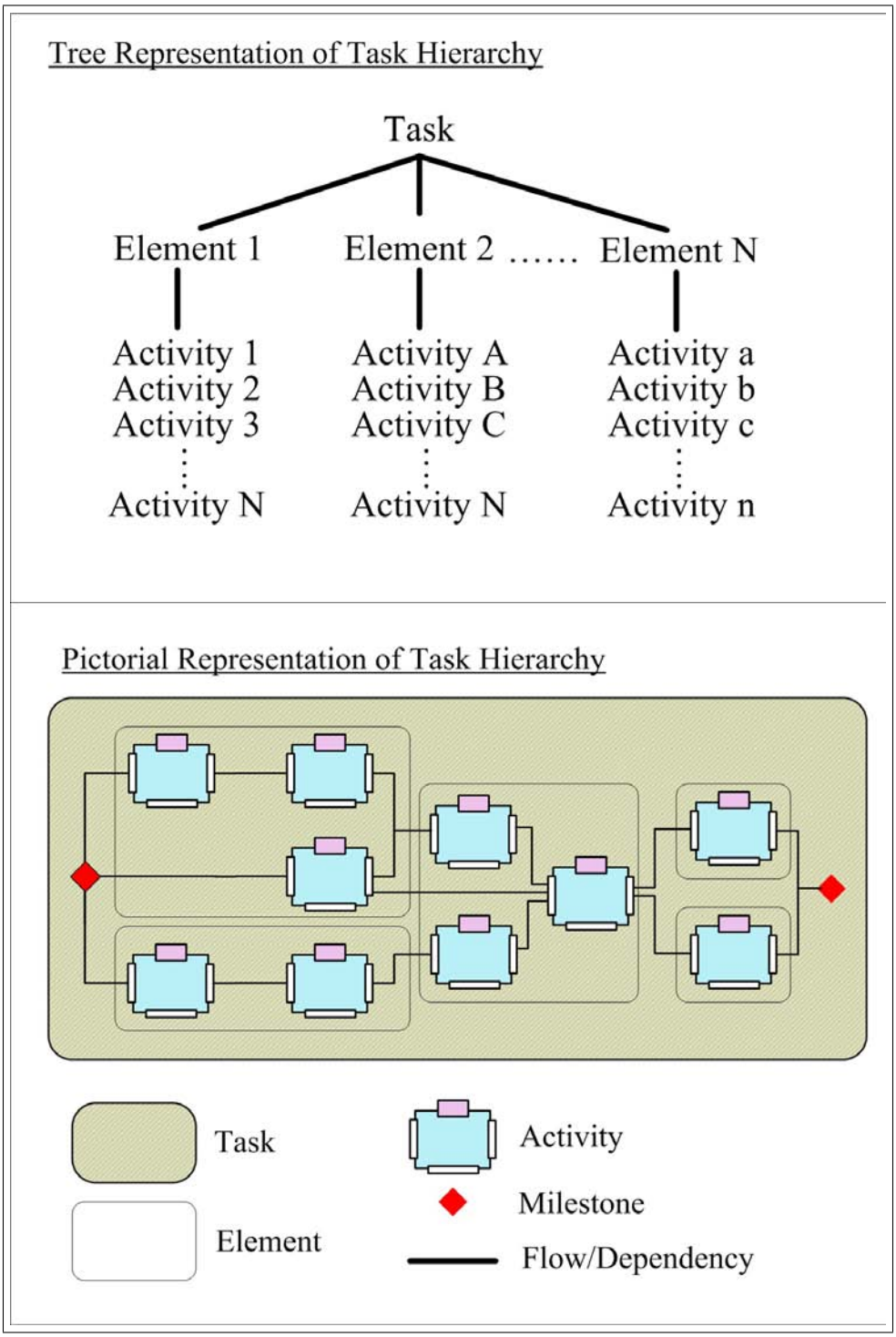


Figure 18 : *The Hierarchical Breakdown of a Task into Elements and Activities.*

to a particular Activity or set of Activities without having to create a new Task-Template with its own instantiation schedule for each variation. Variability is accommodated within MARS by allowing an Element to be interchanged with one of several alternatives called "Element-Options". This concept is illustrated in Figure 19. In effect, each Element is simply a placeholder for one of many potential Element-Options.

Mindful of the Element and Element-Option concepts, it is instructive to view the Task-Template hierarchy from several perspectives. From a "top-down" perspective, a Task-Template contains a set of Elements. Each Element refers to a set of possible Element-Options, and each Element-Option contains a group of Activities[15]. From a "bottom-up" perspective, Activities can be part of many Element-Options, an Element-Option may be referenced by multiple Elements, and each Element may be part of more than one Task. The data structure that encapsulates these relationships is briefly described in section 3.3. From the data structure perspective, the creation of Element-Options creates a fourth level within the Task-Template hierarchy; however each Element makes use of only one Element-Option upon Task instantiation. It is for this reason that the Task hierarchy is best viewed as tri-level.

Given that each Element makes use of only one Element-Option when a Task is instantiated, an important part of an Element definition is the Element-Option schedule. An Element-Option is defined by the user for every Element. An Element-Option schedule consists of a sequential list of Element-Options. When a Task is instantiated, the simulation consults the Element-Option schedule which in-turn dictates the next Element-Option in the sequence. The associated Activities inside that option are then generated. A pointer attached to the schedule is then incremented in preparation for the next instance that the schedule is accessed. For example for the case where MARS is used to analyze a particular MRP, an Element may be created to represent the training of an "Infantry Combat Team". Element-Options consistent with this may include mechanized and dismounted formations. An Element-Option schedule would then determine the sequence by which those options would be required to match a specific MRP.

2.4 Activity Definition

The breakdown of a large Task into Elements, Element-Options, and Activities is easy to conceptualize. It is expected that defining this hierarchy would be the first step taken by the user when developing a Task-Template for testing within MARS. The essential steps for the definition of a Task-Template are enumerated below.

1. Define the hierarchy of Activities, Elements, and Element-Options.
2. Define Task instantiation and Element-Option schedules.
3. Define attributes for each of the individual Activities.
4. Define Resource requirements and Resource Selection Schedules.

[15] It should be noted that in the first MARS prototype an Element-Option need not contain any Activities. These represent “Do-Nothing” options

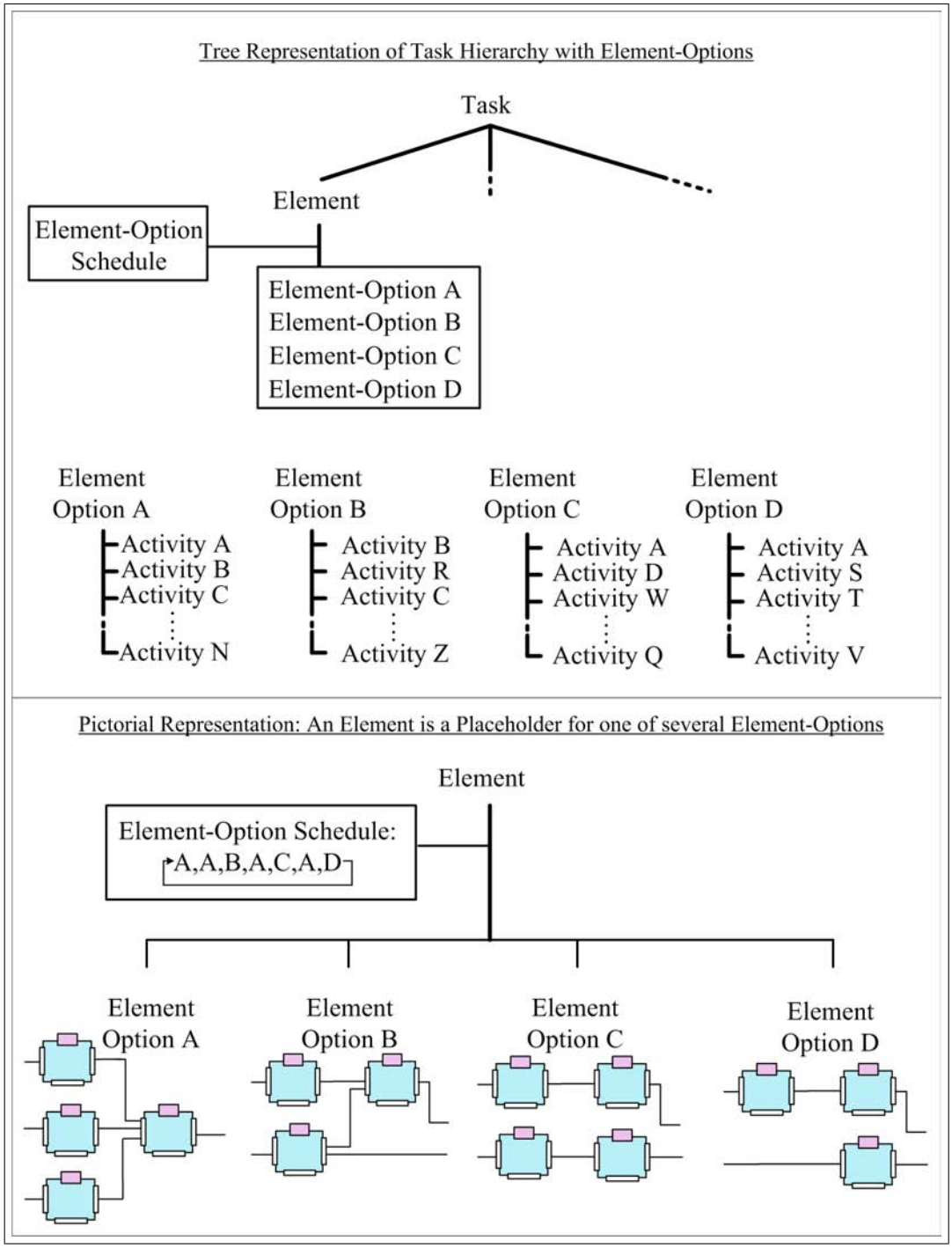


Figure 19 : *Elements in relation to Element-Options. The Element-Options concept provides a degree of flexibility when designing a Task-Template. Each Element represents a placeholder for any one of several Element-Options. The Element-Option schedule dictates which Element-Option is instantiated upon the instantiation of a Task.*

The basic ideas behind the first two steps, template hierarchy and instantiation schedules are described in the previous section. In a sense, when these are defined they constitute the skeletal structure of a Task. A practical example of how these are implemented in the subject of reference [10]. The third step, which focuses on the definition of individual Activity attributes is the focus of this section. The last step, the definition of Resource requirements and acquisition schedules, is the subject of section 2.5.

2.4.1 Activity Attribute Categories

An Activity is defined by specifying a series of parameters called Activity attributes. In broad terms, these attributes are used within a simulation to determine:

- How Activities are linked to each other within the Activity network,
- How an Activity is processed through the Activity lifecycle (introduced in section 1.5.2), and
- How resources are acquired and processed.

In effect, these are the three main functions of Activity attributes, however to facilitate a clearer explanation of individual attributes, they are categorized as follows:

1. General Attributes,
2. Temporal Attributes,
3. An Attribute to Alter the Status of the Parent Task,
4. Internal Routing Attributes,
5. External Routing Attributes,
6. Attributes to Set Resource Status, and
7. Resource Selection Attributes.

The Activity attributes in each of these categories may be defined explicitly by the user or may be set to take on default values. Data requirements for each of the first six categories are described next. Item 7, Resource Selection Attributes, is discussed in section 2.5.

2.4.2 General Attributes

As the name suggests, general attributes characterize an Activity as a whole. Attributes in this category are shown in Figure 20.

Activity Description: This attribute provides the name of the Activity.

Activity Waivers Eligible Status: This binary attribute indicates if personnel assigned to an Activity are eligible to be put on "waivers" upon Activity termination.

Activity RUL: The Resource Utilization Level (RUL) attribute determines the utilization level attached to participating resources. It indicates the proportion of time that resources assigned to an Activity are occupied with that Activity and/or are otherwise

unavailable for participation in other simultaneously occurring Activities. This attribute provides a default that is inherited by all resources participating in an Activity. In most circumstances it is expected that the RUL will be set to 1 in order to indicate complete dedication to an Activity. However the functionality provided by setting an RUL that is less than 1 enables individuals to participate in concurrent Activities. The RUL attribute also allows for the creation of special "Null-Activities" with RUL values of 0. As a practical example, a Null-Activity would be used to set the waivers status of personnel within a Resource. Such an Activity does not require the "participation" of resources in the regular sense; instead the Activity would serve as a vehicle to set or reset the statuses of attached resources. A Null-Activity might be used to set or reset the LOB status of personnel. The RUL value remains constant over the course of an Activity. In cases where the RUL is expected to change significantly over the course of an Activity, that Activity should be split into 2 or more separate Activities for modelling in MARS.

Activity Importance Level: The importance of an Activity relative to others is defined via the Activity Importance level. It is measured using an absolute scale from 0 to 10 with a value of 10 indicating highest importance (*i.e.* priority). As will be discussed later, the priority level attribute is used when selecting and/or allocating resources being sought after by other Activities that are to occur concurrently. In the first MARS prototype the Importance Level attribute is set as input by the user and remains fixed for an Activity over the duration of a simulation. As the sophistication of MARS grows, the opportunity exists to assign a hierarchy of different priority levels across Elements and Activities and provide algorithms that would dynamically alter priority levels (and thereby affect resource allocation) based on some dynamic scheduling methodology. An example could be the critical path methodology where Activity priority levels might be set by analyzing temporal quantities such as slack-time [11, 9]. In any case, the current utility of the Activity priority level attribute within MARS provides a basic ability to approximate the competition for resources.

Activity Probability of Occurrence: This attribute takes on a value between 0 and 1 and enables the instantiation of an Activity on a stochastic rather than a deterministic basis. It is expected that most Activities will be prescribed a value of 1, however in cases where there is some uncertainty about whether a certain Activity will be required,

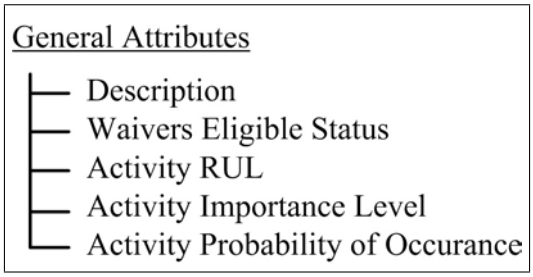


Figure 20 : *General attributes of an Activity.*

values less than 1 may be specified. For example, this attribute may be of use in relation to certain deployment Activities that do not occur on a set schedule. The probability of occurrence attribute, together with the previously described stochastic Task instantiation schedules, are features that enable the simulation of uncertainty and randomness in the occurrence of events within MARS.

2.4.3 Temporal Attributes

The temporal attributes of an Activity, when combined with attributes to be described later in section 2.4.6 concerning the "flow" of Resources between Activities, implicitly define when a particular Activity is to occur in relation to others within the Task. In the spirit of using "best practices" for project management [9], MARS provides the facility to specify temporal constraints associated with a particular Activity. These are shown in Figure 21 and explained below.

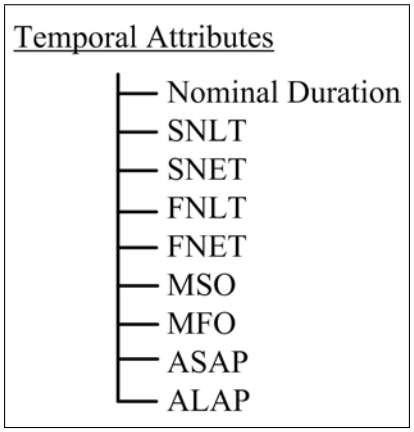


Figure 21 : *Temporal attributes of an Activity. Note that only a subset of these are implemented in the first MARS prototype.*

As the name indicates, the Nominal Duration attribute of an Activity indicates its duration in time. The acronyms ASAP and ALAP mean "As Soon As Possible" and "As Late As Possible", respectively. Each is assigned as an exclusive alternative to the other to determine when an Activity will be scheduled in cases where the window of availability for carrying out the Activity is longer than its nominal duration.

The acronyms MSO and MFO stand for "Must Start On" and "Must Finish On". These are hard constraint attributes that indicate the need for a specific Activity to take place on or by a specific day. Softer constraints akin to MSO are SNLT and SNET which mean "Start No Later Than" and "Start No Earlier Than", respectively. Softer constraints akin to MFO are FNLT and FNET which mean "Finish No Later Than" and "Finish No Earlier Than". Evidently the specification of timing attributes can not be done on an independent basis since each relates to the other. Proper care must be exercised in the definition of these to ensure that Activity timing does not become over-constrained.

As scheduling algorithms within MARS are developed, temporal attributes may be used in combination with other attributes to enhance resource selection algorithms. In the first prototype of MARS, scheduling of Activities is assumed to be ASAP and requires the specification of specific dates for SNET, SNLT and nominal duration.

As with all temporal inputs, the default time increment within MARS is a single day. SNET is specified relative to a specific date called the "Level 1 Anchor time" which is an attribute that is specified as part of the Element-Option that includes the Activity. As the name implies, the "Level 1 Anchor Time" is used as an anchor against which other time attributes are relative. The reader is referred to section 3.3 and Figure 37 for a more complete discussion of anchor times and their relation to SNET, SNLT and nominal duration.

2.4.4 An Attribute to Alter the Status of the Parent Task

Within the context of the first MARS prototype, the personnel, materiel, and/or facilities that make up a Resource may be selected and assigned to Activities based on:

1. Where they are located within the establishment, equipment fleet or infrastructure, or

2. the Tasks in which they are currently participating.

The attributes of an Activity used to alter the status of a Task are used in reference to the second of these. By setting these attributes appropriately, the status of an entire Task can be altered upon the start or completion of a particular Activity. This action could be used to signify that a particular subset of resources attached to any or all of the Activities within the Task are available (or unavailable) to be transferred to supply Activities in other Tasks. To understand how Task-setting attributes can be of use, consider an example where one Task-Template is defined to represent "training for deployment" and another is defined to represent the "deployment" itself. This example is illustrated in Figure 22. From this figure, note that within the "training for deployment" Task there is a specific "milestone" Activity. This “milestone Activity” qualifies the attached resources as available to be transferred to a "deployment" Task. Although not the last Activity in the training Task, the milestone Activity can be used to signify that attached resources are ready for deployment. In such a case, this “milestone” Activity would alter the status of the HR Training Task. The Deployment Task would be defined to search within the simulation for an HR Training Task with an appropriate status in order to extract individuals and fill Resource requirements.

2.4.5 Internal Routing Attributes

The Activity attributes related to internal and external routing explicitly define how resources flow through and between Activities. In fact, this routing, along with the timing attributes assigned to individual Activities, implicitly determines how Activities form the network that composes a Task. To better understand this concept, a graphical representation of an Activity is presented in Figure 23. In this figure categorical placeholders for general, temporal, and Task-alternating attributes are represented. Also shown are three collections

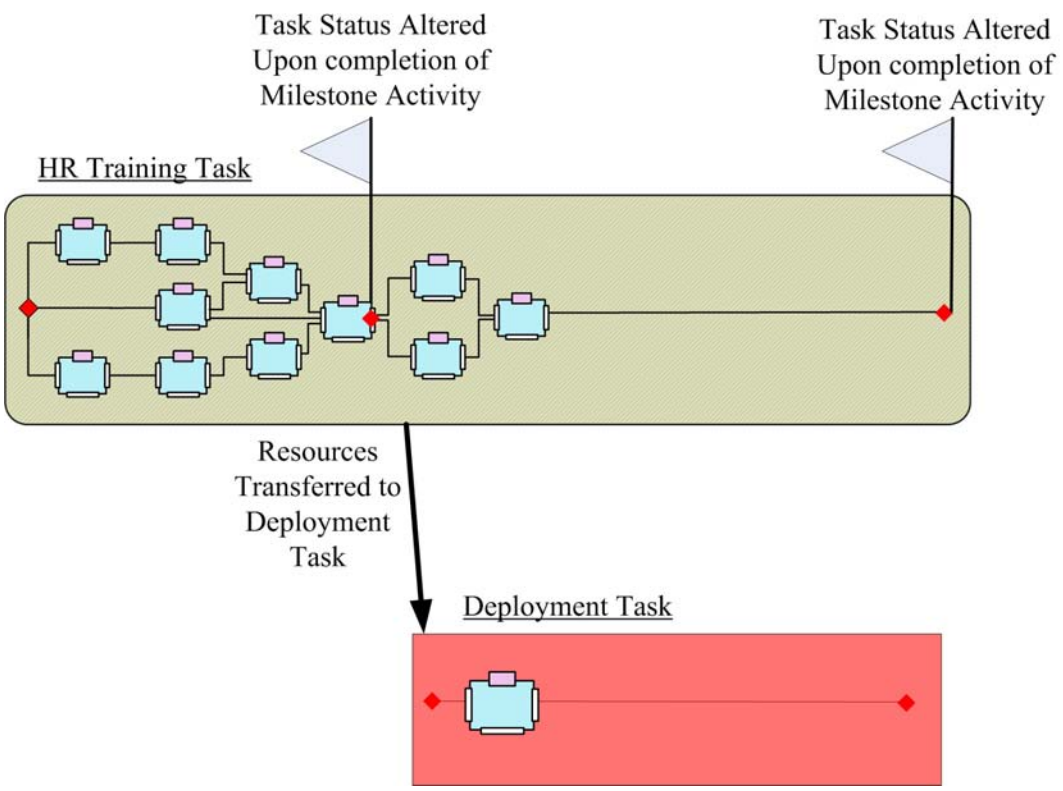


Figure 22 : *An Activity can be designed to alter the status of an entire Task in order to signal the availability of Resources for participation in other Tasks. Consult sections 2.4.4 and 2.5.5 for details.*

of nodes on the periphery of the Activity. Each node represents a conduit where Resources can be considered to either enter or leave the Activity.

- Nodes along the left of the Activity are input conduits for Resources that are passively accepted by an Activity upon completion of a prerequisite Activity. For lack of a better term, these nodes are called “Feeder Nodes”.
- Nodes along the bottom of the Activity are input conduits for Resources that are purposely selected, assigned and seized from either the establishment, vehicle fleet, or another Task. These are called “Finder Nodes”. Each Finder Node serves to acquire a single Resource.
- Nodes along the right hand side of an Activity are output conduits for sending Resources to a subsequent Activity. These nodes are called “Next-Activity Nodes”.

As illustrated in Figure 23, the internal routing within an Activity dictates which input nodes are “connected” to which output nodes. In the simplest case each Next-Activity Node would be connected to a single Feeder or Finder Node so that each Resource would enter and leave via a separate route. However, a more common case is when one or more

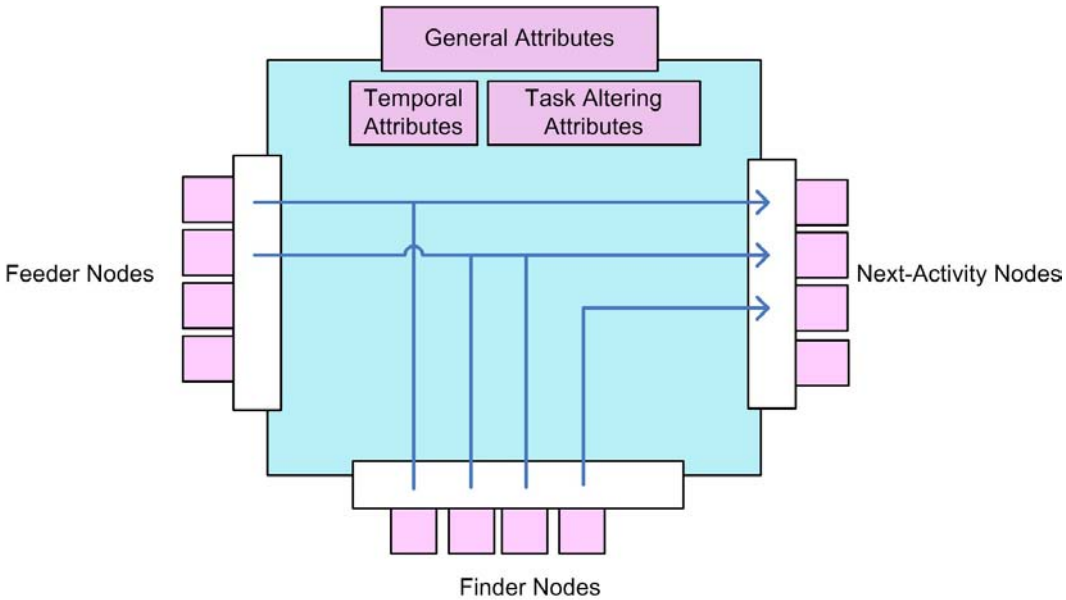


Figure 23 : *Pictorial illustration of an Activity. Attributes attached to Feeder, Finder and Next-Activity nodes control the acquisition and the routing of Resources.*

Resources that participate in a particular Activity are to leave together to participate in a particular subsequent Activity. To accommodate this case, the Activity representation allows each Next-Activity Node to be connected to more than one Feeder and/or Finder node. Based on this representation, Resources can be viewed as being *effectively* grouped before being sent to a subsequent Activity[16].

The ability to effectively group Resources before sending them to a subsequent Activity greatly simplifies the design of Activity networks where, as is common in practice, groups of Resources are batched together to form larger and larger units as they proceed through a training regimen. Once Resources are effectively batched, they may be considered to become a new and larger Resource. Note however that in the first MARS prototype there is no mechanism to un-group or split apart Resources[17]. Future developments in this area may be considered as they become required.

Each node in the Feeder, Finder and Next-Activity categories is numbered sequentially and this number is assigned to an attribute called the "Internal Number". This attribute is of primary interest when defining internal routing. Specifically, the internal number of each

[16]It is important to note however that although internal routing may be viewed to "effectively" group resources, these resources remain separate within MARS algorithms. Grouping or "batching", as it is more commonly referred to in simulation terminology, is not actually performed by MARS.

[17]In the first MARS prototype there does not exist a mechanism to split a Resource and send some individuals to one Next-Activity node and some individuals to another. As an example, this might be required in cases where an Activity represents an individual training course. Some students entering the Activity through a particular Finder node would pass the course and other students would fail. Depending on their pass/fail status, each would be send to a different Next-Activity node.

Next-Activity node may be linked to a set of internal numbers for the finder nodes and a set of internal numbers for the feeder nodes. These linkages embody the internal routing. The relational data structure used to store these linkages is illustrated in section 3.3.

2.4.6 External Routing Attributes

Whereas the term internal routing refers to how Resources flow through an Activity, the term "external routing" indicates how Resources flow between Activities. An illustration of external routing is presented in Figure 24. The attributes that control the external routing between Activities are connected to Next-Activity and Feeder Nodes. With respect to the former, the attribute of relevant interest is the ID number of the subsequent Activity. Likewise, with respect to Feeder nodes, the attribute of relevant interest is the ID number of the preceding Activity.

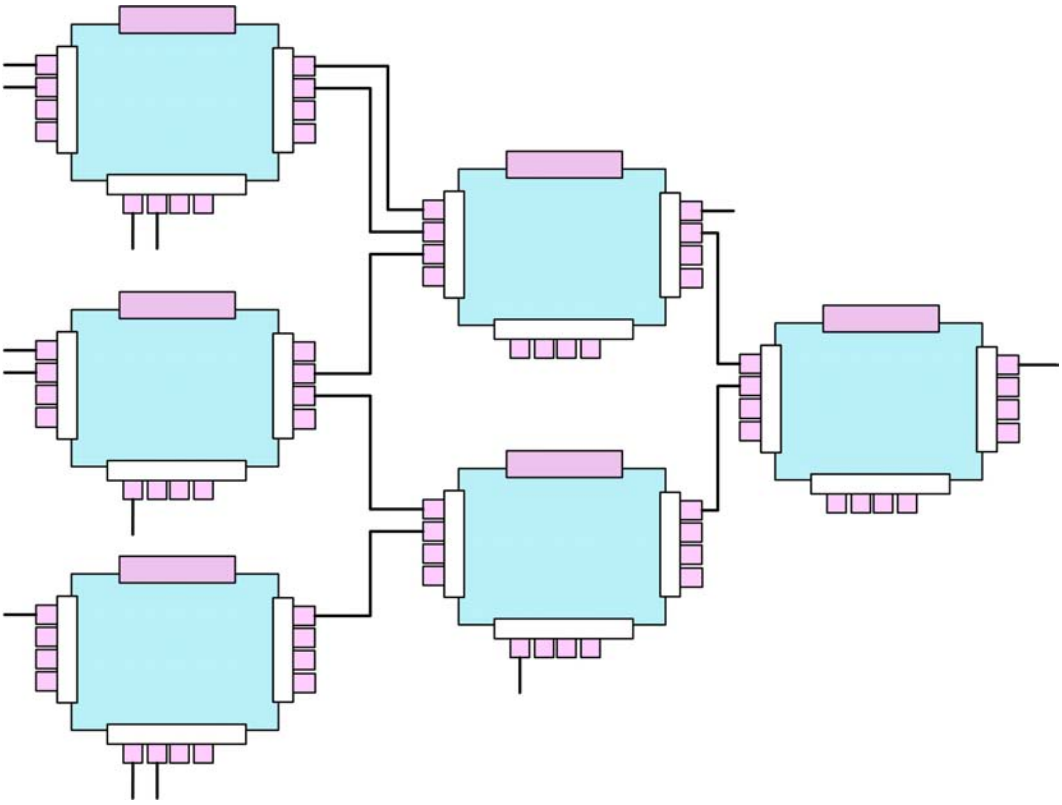


Figure 24 : *External Routing. Although Resources are tracked individually inside MARS, it is conceptually convenient to view routing as a means to batch several Resources together. In this example several individual Resources are effectively batched into a single Resource in the last Activity.*

To pass Resources, each Next-Activity Node is visited in turn. The ID number of the

subsequent Activity is identified and that Activity is found within the simulation[18]. Once found, each of the feeder nodes is checked in-turn. The first Feeder node with a matching prerequisite Activity ID that has not already received a Resource is chosen to accept the Resource. In the first MARS prototype the algorithms allow a Resource to be passed to only one subsequent Activity. Considering that Activities can have different Resource utilization levels (RUL), it is expected that this limitation will be removed in future versions. The relational data structure used to store the attributes that control external routing is illustrated in section 3.3.

In most cases the external routing constructs within MARS facilitate the movement of Resources from one Activity to the next. However, the function of external routing has a broader scope. It can also be used to enforce the sequential dependence of Activities in a Task network that do not pass Resources between one another. This idea is illustrated in Figure 25.

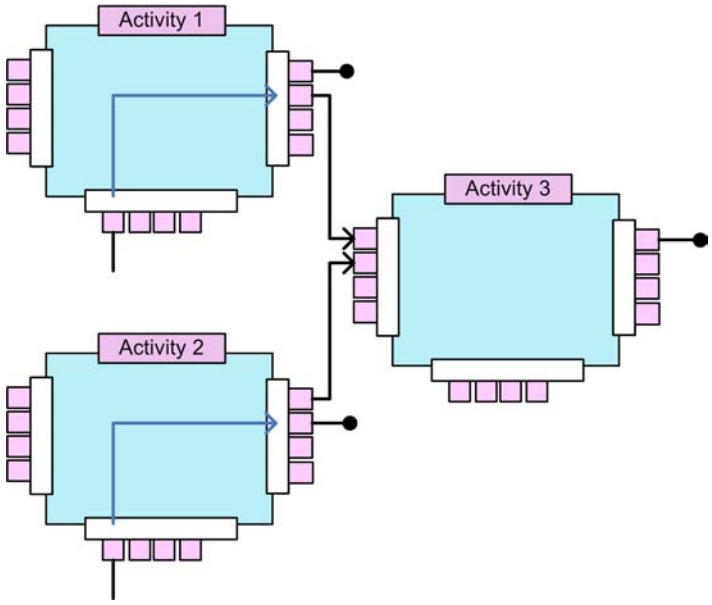


Figure 25 : *External Routing to indicate Temporal Dependence. It may the case that one Activity is not supposed to start until another is finished but neither share the same Resources. This is the case for Activities 2 and 3 above. This scenario is accommodated in MARS by linking a Next-Activity Node on Activity 2 to a Feeder node on Activity 3. No Resources would be passed between the Activities but the sequential dependence between Activities is maintained.*

2.4.7 Attributes to Set Resource Status

Not only are the Feeder, Finder and Next-Activity nodes of pivotal importance for specifying Resource routing, they also provide a conceptual basis on which to prescribe sep-

[18] As is discussed in sections 3.1 and 3.7, the Global Wait Queue is searched to find dependent Activities.

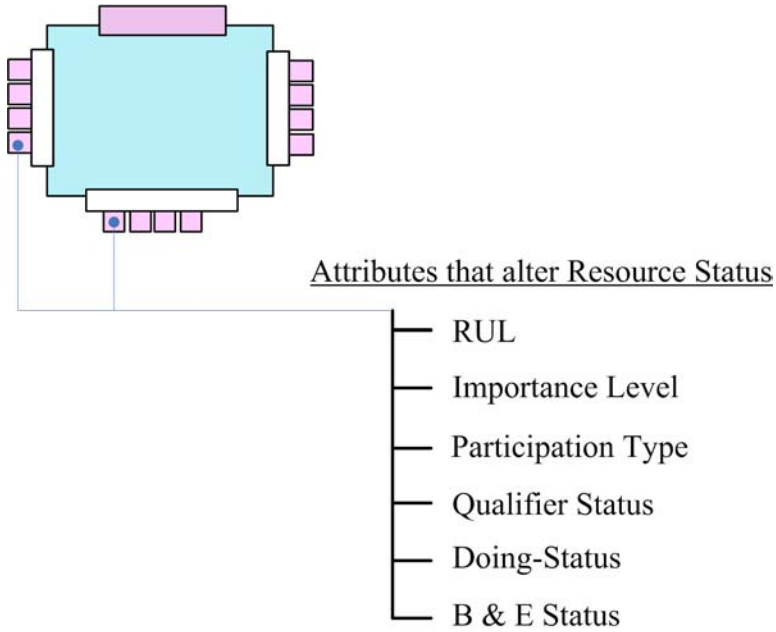


Figure 26 : *Attributes that serve to alter the status of individuals in a Resource are attached to Feeder and Finder nodes.*

arate attributes to individuals in Resources that take part in an Activity. For example, by identifying the Resource that enters a particular node, it is conceptually straightforward to understand how a separate RUL can be assigned to that Resource by attributing an RUL to the node itself[19]. This concept may be extended to other attributes including the Importance Level, the Participation Type, Qualifier Status, Doing-Status and B&E Status. Each of these is shown is Figure 26 and described below. The relational data structure used to store these attributes is described in section 3.5.1.

Importance Level: In any system where the demand for resources is larger than the supply there is a requirement to prioritize resource allocation. There are a variety of strategies to ascribe priority. In the first MARS prototype a very simple strategy is used. This strategy employs the Importance Level attribute. Each Feeder and Finder Node is apportioned an Importance Level which is then automatically assigned to the associated Resource. This attribute may or may not be deliberately set by the user when defining the node. If it is not set, it takes on the value of the Activity Importance Level (section 2.4.2). The Importance Level for both the Activity and the Resource can be defined to fall between 1 and 10. Individuals that are part of a Resource with a high Importance Level can not be removed or reallocated to become part of a Resource with a lower Importance Level. Sections 2.5.3 and 2.5.5 contain further details about the selection, assignment, and seizure of individuals to become part of a Resource.

[19] An RUL specified for any Feeder or Finder node will override the default RUL specified for the entire Activity.

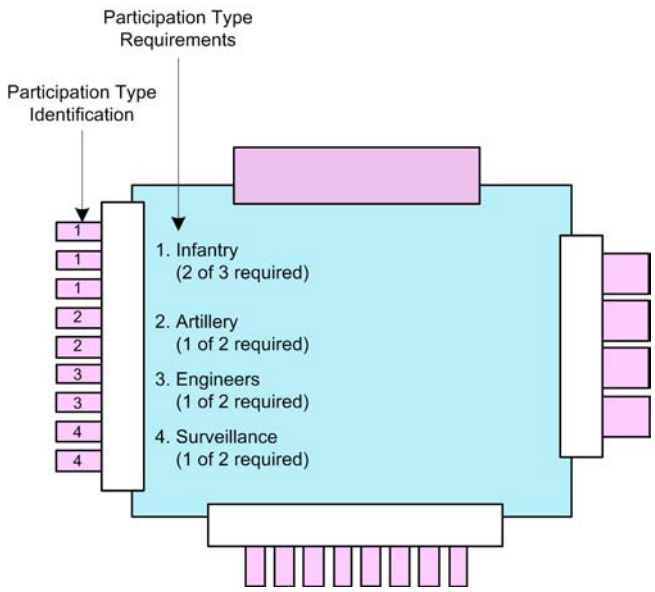


Figure 27 : *Participation Type Example. Refer to section 2.4.7 for a details.*

Participation Type: As is the case in practice, the way that some Resources participate in an Activity is different than others. An Activity that represents a training course is one example where this is the case. Resources representing instructors, students, supporting personnel, and equipment, will play different roles when participating in a training Activity. Accordingly, there may be different criteria on the minimum number of Resources that are required to fill each role before the Activity can begin. The input of this type of information is made possible by first assigning a Participation Type attribute to each Feeder and Finder node and then specifying minimum requirements for each Participation Type as part of the Activity definition. A practical example that illustrates the use of the Participation Type attribute is an Activity to represent collective training at the CMTC. Depending on the intention of the modeler, this Activity may require a certain number of resources in each of several different types (*e.g.* infantry, artillery, engineering, surveillance). This idea is illustrated in Figure 27. For this example the group of nodes accepting each of these Resource types would be assigned a different participation type. The minimum number required for participation type would become part of the Activity definition. As shown, the Activity can be defined so that three Feeder nodes accept infantry Resources but only two such Resources would be required for the CMTC Activity to commence.

Qualifier Status: This optional attribute is used to set the Qualification Status of a Resource to a specific level while participating in an Activity. The Qualification Status attribute is used to prepare Resources for removal from one Activity to another. Different levels of Qualification Status provide a ranking to prioritize those Resources that are most available for removal. An example where qualification levels can be useful is where two resources are to be “on-call” on an alternating basis over a prescribed

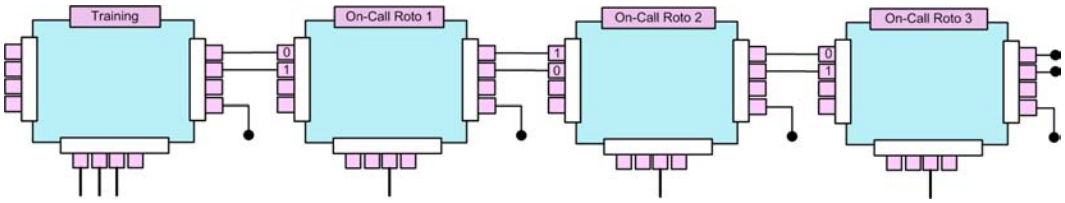


Figure 28 : *A simple On-Call Example. Qualifier Status (i.e. 0 or 1) is used to demarcate the Resource that is "On-Call" and the Resource that is available as backup.*

period of time (see Figure 28). In this case, several Activities would be aligned in a sequence. The same two Resources would participate in each Activity but upon entry into each Activity the Qualification Level of each would alternate to signify that one is "on-call" whereas the other is not. Further discussion of the purposeful removal of Resources already participating in an Activity is presented in section 2.5.4.

Doing-Status: In the first MARS prototype the user can define any number of status types where each has a number of status levels. This idea is illustrated in Figure 29. When an Activity is processed within a simulation, the individuals in the participating Resources may be assigned multiple statuses. Status levels are assigned upon commencement and/or termination of the Activity. Status types fit into two families. The first family of statuses corresponds to those which are set at the start of an Activity and automatically returned to a null level upon the completion of the Activity (see Figure 29). The status types in this family are called "Doing-Statuses" because when they are linked to a Feeder or Finder Node they indicate that individuals within the corresponding resources are "Doing" some particular job or action. With respect to the current MRS, a "Collective Training" status is one example of a Doing-Status that could be defined by the user. Levels within this "Collective Training" status could be defined to match the Army's standardized collective training levels (1-7). By defining collective training levels in this way, the individuals in resources that participate in a particular training Activity could be appropriately demarcated.

B&E Status: The second of the two families of status are called B&E[20] statuses. Whereas statuses in the Doing-Status family are only assigned while an individual within a Resource is actually participating in an Activity, the statuses in the B&E family may be used to indicate that an individual has successfully started or completed a certain Activity. In other words, in contrast to Doing-Statuses which are automatically reassigned to a null level upon the completion of an Activity, statuses in the B&E family are not automatically reset. This idea is illustrated in Figure 29. A B&E status that is set at the beginning or end of an Activity is maintained until purposely reset in either the same Activity or another Activity. Like Doing-Statuses, B&E statuses can be used to qualify or disqualify individuals for potential reassignment to other Activities.

[20]Note that name "B&E" is coined because statuses in this family can be assigned at the *beginning* or *end* of an Activity.

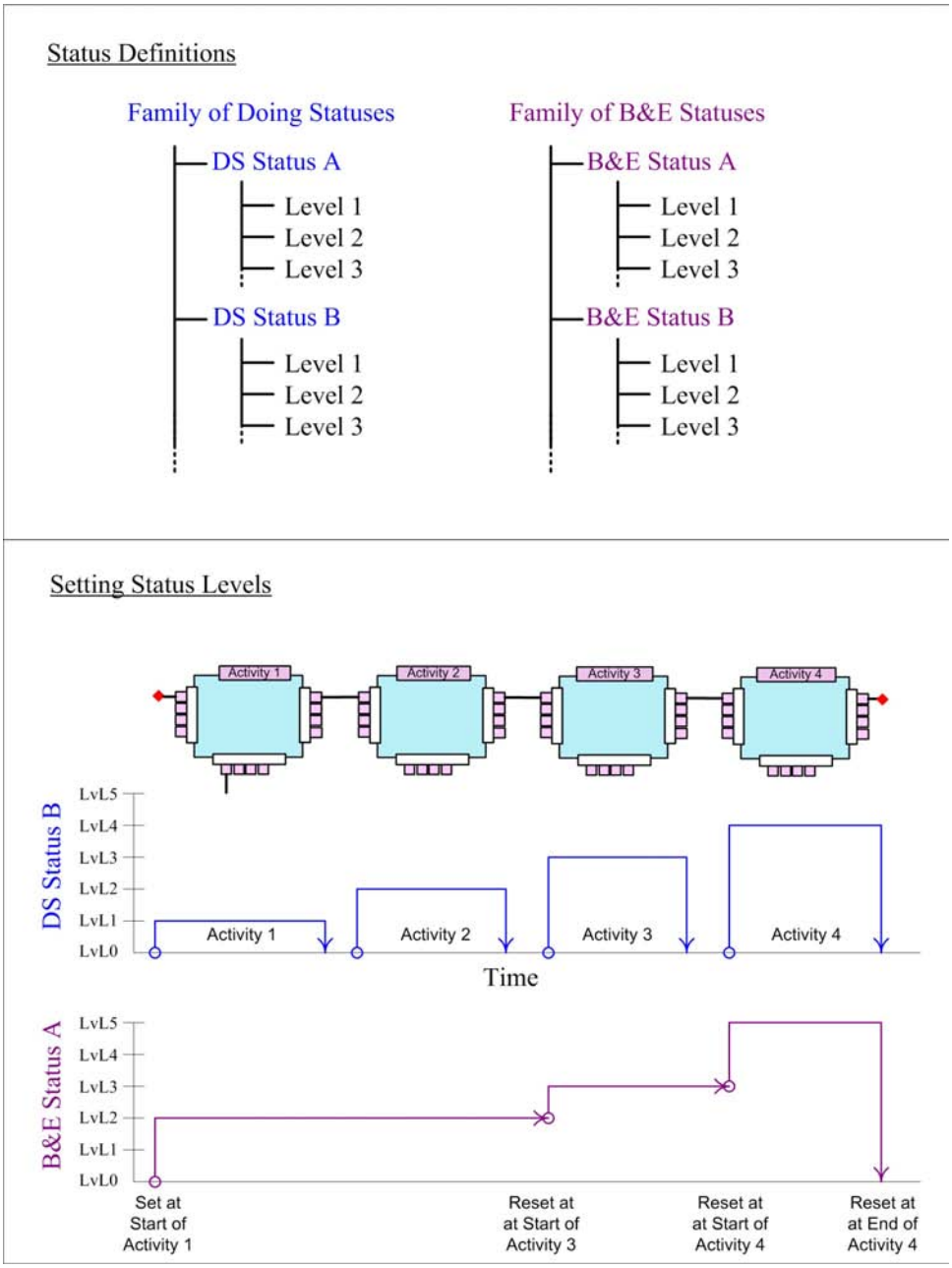


Figure 29 : *Status Definitions and Timing. Defining the family of Doing Statuses and B&E Statuses is illustrated at the top of the figure. Each status in a family may have several levels. A Resource may be assigned several statuses at one time. As shown at the bottom of the figure Doing Statuses are assigned to individuals in a Resource only while participating in an Activity. After and Activity is finished Doing statuses are reset to the null level (LvL 0). On the other hand, B&E Statuses can be maintained after an Activity is complete. As the name suggests, B&E statuses are set at the beginning or end of an Activity. (Note: For purposes of simplicity, all Activities in this example have the same duration.)*

For example, statuses in the B&E family could be used to tag individuals that have attained a skill level that has an associated expiration date. A particular status level could be set upon completion of one Activity and then reset upon commencement of a subsequent Activity. The time lapse between the Activities would correspond to the time period that the skill level is deemed valid.[21]

2.5 Resource Selection Attributes

Resource selection attributes facilitate the definition of supply and demand as well as resource acquisition schedules. Evaluating plans to allocate Resources (personnel, materiel, and facilities) is one of the main functions of MARS. To this end, several attributes are assigned to an Activity to characterize the requirement for Resources. Like the other Activity attributes, Resource requirement attributes must be input by the user[22]. In broad terms, the user needs to build three constructs to define the following attributes in MARS.

1. The total supply of resources.
2. The demand for resources.
3. Acquisition schedules (These allow the user to schedule which parts of the supply shall be assigned to fill a particular demand.).

These constructs are illustrated in Figure 30 and described in sections 2.5.1, 2.5.2, and 2.5.3. It is intended that all three of these constructs be created in a step that is separate from the initial creation of a Task-Template. Only at the final stage of template design should pointers to these constructs be linked to Finder Nodes within each Activity. This aspect is the subject of section 2.5.6.

2.5.1 Defining a Resource Supply

A critical input in the creation of a scenario within MARS is the definition of Resource supplies. In military terminology this means representing the organization of personnel in the establishment, the equipment pools, and the network of facilities that make up the infrastructure. In the first MARS prototype only utilities for the creation of the establishment have been implemented. This follows directly from the fact that personnel issues within the Army have been the main focus in the development of the first prototype. With this in mind, it should be noted that the principles used to represent the establishment within MARS are expected to be applicable to those used for equipment and facilities.

Within the Canadian Military, the term establishment is used to define the hierarchy of formations, units and subunits that are used to organize personnel in garrison. Within this

[21] In the first prototype of MARS Statuses can be set only at the start or end of an Activity. In the future it is expected that the ability to alter statuses will be extended to other attributes like Rank, military occupation, and qualification level so that these may be altered over the course of a simulation. Among other things, this will facilitate the creation of Activity constructs to model recruitment, promotion, posting, and attrition.

[22] In future versions of MARS it may be desirable to incorporate algorithms to automatically deduce selected Resource requirement attributes based on superseding inputs and/or scheduling algorithms; however it is expected that there will always remain a need to directly input Resource requirements into MARS.

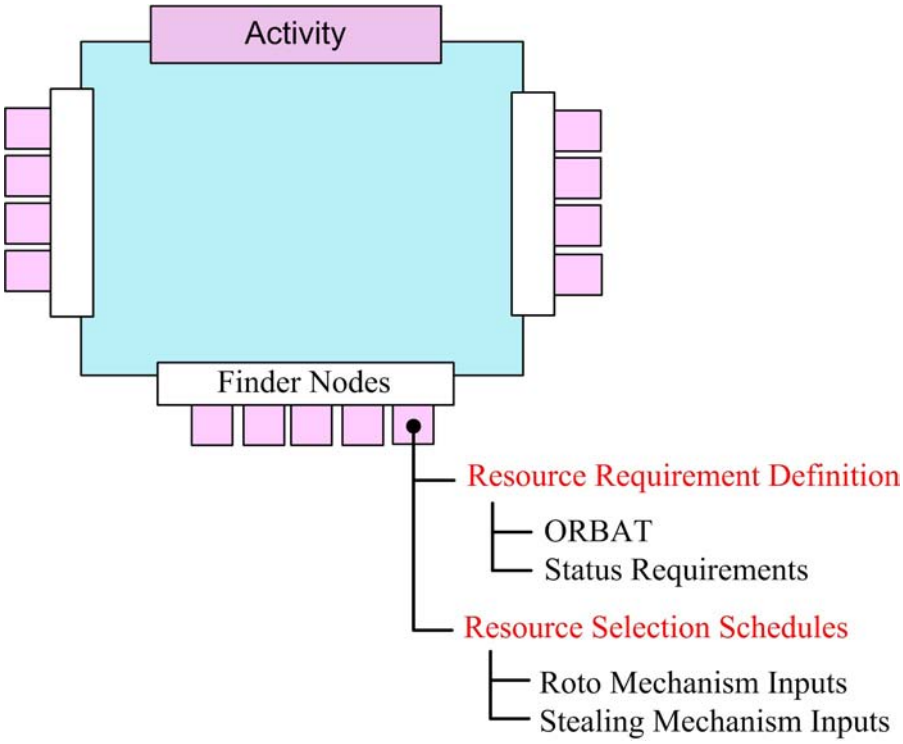


Figure 30 : *Attached to each Finder node is information that characterizes the demand for resources (i.e. what is required) and scheduling information that characterizes the parts of the establishment, equipment and/or facilities pools that are responsible for filling demands at any given point in time.*

hierarchy a variety of categorizations, such as that between the institutional & field force or the regular & reserve force, can be made. However, at its most primitive level, the establishment is a tree structure where each branch may have several children branches and no child can have more than one immediate parent branch. A particular example is that of the Army field force which is currently composed of Area formations that contain battalion or regimental units. These units may be further broken down into companies, troops, platoons, and so on, until finally the terminal branches in the tree contain individual positions. Each position represents a requirement that can be defined in terms of a Rank, Military Occupation Code (MOC)[23] and in some cases a set of Qualifications. At any point in time, these individual positions may be empty or they may have one or more personnel assigned to them. Each of these personnel also has a particular combination of Rank, MOC and Qualifications.

Utilities for defining the establishment, position requirements, and the personnel assigned

[23]Military Occupation Codes have now been officially replaced by Military Occupation Specialty IDs (MOSIDs) however legacy systems still exist. MARS can accommodate both through an internal conversion process.

to each position are incorporated into the MARS interface[24]. This information is then stored within the MARS database[25]. MARS is designed so that a pointer to any branch within the establishment tree can be used to produce a list of personnel whose candidacy may be tested for filling Resource requirements. In the first prototype of MARS, the value of the pointer is the name of the branch within the establishment. As such, a list of candidate individuals to fill a certain Resource requirement can be created by specifying the name of any formation, unit, subunit, *etc*. As will be described in section 2.5.4, this concept is of importance for the definition of Resource Selection Schedules.

One important aspect not directly related to the design of MARS but nonetheless important when considering its practical utility is how data about the establishment is collected. Within the Canadian Forces there are several places from which information about the establishment and manning levels can be obtained. One source is the On-Line Establishment Browser (OLEB); another is the Human Resources Management System (HRMS). Scrubbing and importing these data into MARS is expected to be one of the first steps in the construction of a scenario.

There are two basic options for incorporating personnel into a simulation. For the first option MARS provides a utility to automatically create one individual for every person in the establishment. This option mimics a “best case supply” where all positions in the establishment are filled. For the second option an ensemble of individual personnel may be assigned to a single position in each unit. This option may be chosen because current CF manning databases may not typically contain information to assign individuals to particular positions within a unit. The main issue when considering the choice between these two options is the sensitivity of simulation results to the "completeness" and "accuracy" of data that characterize the supply of personnel. A related issue concerns individual qualifications. In many cases the indication of an individual’s Rank and Military Occupation is inadequate to determine his or her candidacy to fill a particular position in a theatre of operation. If these position and personnel qualifications are not available to be input to the simulation, the capacity of the military to fill operational demands may be overestimated. These and other aspects must be considered when applying MARS to analyse real problems.

2.5.2 Defining A Resource Requirement/Demand

The method to define a Resource requirement as the demand for a collection of positions, each with a range of Rank, MOC, qualification, and status prerequisites, is similiar to that described for the establishment (see section 2.5.1). It begins with the definition of an order of battle (ORBAT). As indicated in section 1.4.2, different ORBATs may be defined and/or imported for use within MARS. An ORBAT consists of a tree hierarchy of Unit(s) and Subunit(s) where the terminal branches contain a collection of individual position requirements. However unlike the establishment, these positions are not directly linked to personnel. Instead, individuals are linked to position requirements when they become part

[24] A screen-shot of the graphical interface to this utility is shown in Figure 7.

[25] The tabular structure used to store this information in the relational database is depicted in Figure 48. The establishment hierarchy is stored within a single table. Other tables hold position and personnel information.

of a Resource during the course of a simulation[26].

The algorithms within MARS are designed such that a single pointer to a branch within any ORBAT can produce a collection of individual positional requirements. Pointers such as these are assigned to each Feeder Node to become part of the Resource requirement.

An attribute of MARS that sets it apart from many of the Force Generation simulators developed in other countries is that it explicitly recognizes that the ORBAT for units to be deployed to a theatre of operation can be different from those in garrison. This is especially true for the Canadian Army where, unlike its American and Australian allies, attempts have only just begun to transform the regimental structure of the establishment to something which is more consistent with the structure of deployable units. With this in mind it is to be expected that in the current Canadian system, an infantry unit in garrison can be different from an infantry unit sent out for deployment. In reality, deployed units are often larger than garrison units and so augmentees[27] are often required. This can be accommodated in MARS.

2.5.3 Constructing a Resource Selection Schedule

In a basic scenario, an Activity could be designed so that each time it is instantiated it would call upon a particular part of the establishment to fill each Resource. However, in as much as this is possible, MARS is designed to allow the user much more flexibility when determining where to select individuals to fill the positions within a Resource. This flexibility is enabled via two mechanisms: the Roto Mechanism and the Stealing Mechanism. Each allows the user to prescribe a set of search rules that ultimately indicate the most appropriate location from which to select individuals to fill a Resource based on the current condition of the simulation.

The Roto Mechanism: The first mechanism allows the user to encode search rules in terms of periodic cycles called rotations or “Rotos”. As such, it is called the Roto Mechanism. Each time that a Task-Template is instantiated, a new location within the establishment is selected based on a periodic pattern that is input by the user. Once the end of the pattern is reached, it cycles to the beginning. This mechanism approximately conforms to how the Army currently assigns units to MSTFs in the current MRP.

The Stealing Mechanism: The second mechanism allows for the direct selection of individuals to fill Resource requirements from those who are already assigned to other Activities. This mechanism is called the “Stealing Mechanism”. Particular Tasks, Activities, and Resources can be pinpointed for the purpose of "Stealing" individuals.

[26] With the exception of the linkage to personnel, the tabular structure used to store ORBAT info in the relational database is very similiar to the establishment (refer to section 3, figures 48 and 46). All ORBATs are stored in a single table and other tables hold the positional requirement information.

[27] The need for augmentees underlies the "borrowing" and "stealing" that often occurs when attempting to fill Tasking requirements. A common metaphor is that the Army is required to "Steal from Peter to pay Paul". This statement is an implicit acknowledgement of the fact that by borrowing from a unit to fulfill an immediate demand there can be repercussions when that unit is required to fill subsequent Tasks.

The Roto Mechanism and the Stealing Mechanism are described in-turn in the discussions that follow (sections 2.5.4 and 2.5.5).

2.5.4 The Roto Mechanism

In the first MARS prototype, the Roto Mechanism is designed to operate based on the current structure of the Canadian Army's field forces. However, it is important to note that it may be extended to the other services or other structures.

The Roto mechanism allows for the creation of pointers into the establishment tree. As indicated in section 2.5.1, these pointers take on the name of the branch within the tree. For example, a pointer to the first platoon of the first company in the first Battalion of the Princess Patricia's Canadian Light Infantry would be `LFWA\1 PPCLI\A COY\1 PL`. This example makes clear the more general observation that every pointer can be decomposed into three parts: the Area formation (`LFWA`), the Unit (`1 PPCLI`), and the SubUnit listing (`A COY\1PL`). Note that in the last case the subunit listing is two branches deep. In MARS the term SubUnit is used to refer to any branch beyond the Unit designation. That is, any aggregate below the Unit level is termed a SubUnit.

In MARS an establishment pointer can be constructed by concatenating an Area, Unit and Subunit. Using the Roto mechanism each of these can be determined independently based on a Rotation in a way that loosely mirrors how a series of schedulers might assign establishment resources to recurring Tasks in practice. For example, a recurring Task might be assigned to each Area on a rotating basis or in terms of some predefined sequence. Upon assignment of the Task to a particular Area, responsibility for the Task would be delegated to a particular Unit. If multiple units have resources to fulfill the Task, Unit selection would likely occur on a rotating basis. Likewise, the Task may be delegated on a rotary basis to particular SubUnits. In essence, the Roto Mechanism allows for the creation of Rotary sequences on each of the Area, Unit and SubUnit levels. It also allows for the specification of alternate locations which serve as a reservoir to draw augmentees for those cases where the supply of individuals in the designated primary location is inadequate. The Roto mechanism is composed of three constructs: Selection Rules, Rotos and Roto Collections. These are described below and illustrated in Figure 31.

Selection Rules: A selection rule is an ordered sequence where each item in the sequence has two attributes: the Name of a location and an attribute to specify whether this location is designated as "primary" or "alternate". For example, a selection rule might be: `1 PPCLI`, `2 PPCLI` and `3 PPCLI`, where `2 PPCLI` and `3 PPCLI` would be alternate units that would be searched if `1 PPCLI` could not accommodate the demand. A fictional example of a SubUnit Rotation might be: `A COY\1 PL\1 SECT`, `A COY\1 PL\2 SECT`, `A COY\1 PL\3 SECT`, `A COY\1 PL\PL HQ` and `A COY\2 PL\`, where the 1*st* and 2*nd* section would be the primary locations and the remainder would be alternate locations.

Roto: A Roto is simply a sequence of Selection Rules. Area, Unit and SubUnit Rotos are given individual names. In Figure 31 a Roto is depicted as a wheel and individual

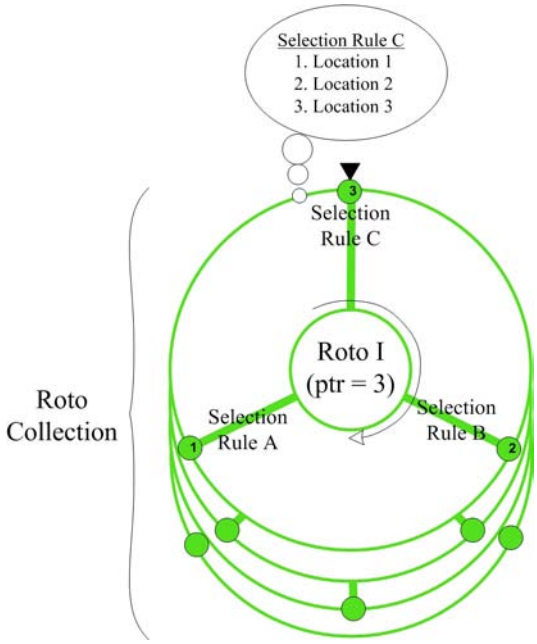


Figure 31 : *The Roto Concept. The Roto Mechanism is based on 3 main constructs: Selection Rules, Rotos, and Roto Collections. A Selection Rule contains an ordered list of locations within the establishment. Several selection rules may be contained in a single Roto. As the Roto pointer is incremented a different Selection Rule is used. Rotos can be grouped to form Roto Collections.*

Selection Rules are ordered along the perimeter the wheel. As the wheel is turned a new Selection Rule is selected. A pointer indicates the current Selection Rule. With the exception of Area Rotos, these pointers are stored as part of the Roto. Several Resource demands can be ultimately linked to the same Roto. Where each one will increment and/or use the current value of the pointer.

Roto Collection: Simply stated, a Roto Collection is a collection of Rotos where each Roto has an associated identifier. In a collection of Unit Rotos, each Roto has an Area identifier associated with it. Similarly, each SubUnit Roto in a Collection of SubUnit Rotos has an associated Unit identifier. Each Roto Collection has a name and, in general, the Rotos within a collection share some commonality. A Roto may be part of one or more collections and for each case it can be associated with a different identifier[28]. The utility of Roto Collections is implicit in how the Roto Mechanism works. The relational structure used to store Roto collections in relation to Rotos and Selection Rules is briefly described in section 3.5.3.

An illustration of how the Roto Mechanism operates is depicted in Figure 32 where from left to right is shown a single Area Roto, a Collection of Unit Rotos, and a collection of SubUnit Rotos. The Roto Mechanism requires four attribute inputs to demarcate the Area

[28] In some cases this may not make practical sense however it is allowable.

Roto, Area Roto Pointer, Unit Roto Collection, and SubUnit Roto Collection that are to be used together. Combinations of these four attributes, which are outlined in red in Figure 32, are represented by a single ID called the Roto Schedule ID[29]. To initiate the use of the Roto Mechanism, a Roto Schedule ID is linked to a particular Finder Node. Based on this specification, the Roto Mechanism cycles through four basic steps. These are explained below and illustrated in Figure 32.

1. In the first step, the Area Roto and the Area Roto Pointer are used to access a particular Area Rule. This Area Rule is used to prescribe the primary Area as well as a list of alternate or “backup” areas responsible for filling the Resource demand.

2. In step two each of the Areas obtained in step one are used together with the demarcated Unit Roto Collection to determine an appropriate Unit Roto. This is accomplished by matching each Area with one of the Area identifiers in the Unit Roto Collection. When a match is found, a link is made to a specific Unit Roto. Each Unit Roto then accesses a Unit Selection Rule which is then used to compile a list of primary and secondary Units.

3. Step three mirrors step two except that it concerns SubUnits. The SubUnit Roto Collection demarcated via the Roto Selection ID is used together with each of the Units compiled in step 2 to link to specific SubUnit Rotos that access an appropriate SubUnit Selection Rule and an associated list of primary and secondary SubUnits.

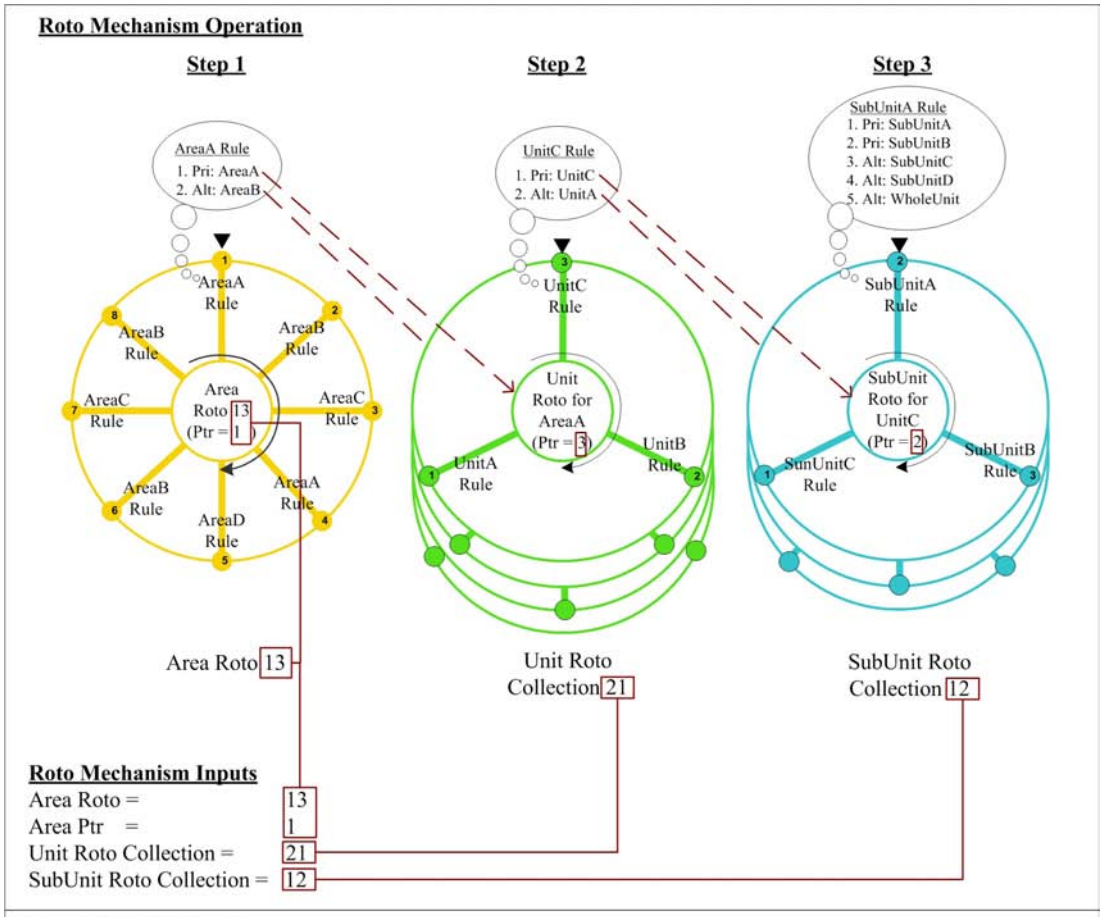
4. In the fourth step the Area, Unit and SubUnit lists are compiled to create a master list of aggregated names which are used as pointers into the establishment to find individuals. During the compilation of this list, individual locations are considered to be primary locations if each of the Area, Unit and SubUnit parts are all designated as "primary" otherwise the location is designated as an "alternate". A listing of pointers to specific branches of the establishment is illustrated at the bottom of Figure 32. From this list, two sublists of candidate individuals to fill the Resource demand can be compiled. The first sublist is called the primary list and the second is called the alternate list. Individuals in the alternate list are only solicited if individuals in the primary list can not meet a specific positional requirement. For scenarios where it is desired to test the capacity to fill Resource demands without delving into augmentee stocks, MARS can be set so that it does not make use of the alternate list[30].

Outside of these four steps, the only outstanding issue with the Roto Mechanism is how it utilizes and increments the pointer associated with each Roto. In this regard, recall the following from previous discussions.

- Task-Templates are instantiated repeatedly over the course of a simulation.

[29] Note that in the first MARS prototype the Roto Schedule ID is called the Hybrid ID.

[30] In future versions of MARS it is expected that various “levels” in the alternate category may be required. This kind of enhancement is not expected to be onerous. Further to this, it is interesting to note that metrics might be envisioned to measure the “degree” or “cost” of assigning individuals from the alternate list.



Step 4: Roto Mechanism Output

PRI or ALT	AREA	UNIT	SUBUNIT
1 Primary	AreaA	UnitC	SubUnitA
2 Primary			SubUnitB
3 Alternate			SubUnitC
4 Alternate			SubUnitD
5 Alternate			Whole Unit
6 Alternate		UnitA	SubUnitC
7 Alternate			SubUnitB
8 Alternate			SubUnitA
9 Alternate			SubUnitD
10 Alternate			WholeUnit
11 Alternate	AreaB	UnitB	SubUnitB
12 Alternate			SubUnitD
13 Alternate			SubUnitC
14 Alternate			SubUnitA
15 Alternate		UnitA	SubUnitA
16 Alternate			SubUnitD
17 Alternate			SubUnitB
18 Alternate			SubUnitC

Figure 32 : *The Roto Mechanism.*

- Each time a Task is instantiated all of the incumbent Element-Options, Activities, and Resource requirements are generated.

- More than one instance of an Element-Option may exist within a Task.

- Multiple Activities may have Feeder nodes that essentially have the same Resource requirement.

The first prototype of MARS has been designed so that pointer incrementation is dependent on the instance of the Element from which the Roto Mechanism is accessed. An explanation of how MARS is designed to access and increment the pointer attached to each Roto is provided in the remainder of this section.

In a conceptual sense, the algorithms that control the incrementation and use of Roto Pointers within MARS create and maintain a new array each time a new Element is instantiated. The field headings for each new array are: Task ID, Element ID, Area Roto, Area Roto Pointer, Area Identifier, Unit Roto, Unit Roto Pointer, Unit Identifier, SubUnit Roto, and SubUnit Roto Pointer[31]. The list of records in each array is unique (*i.e.* no duplicate records are allowed).

The first two attributes, Task ID and Element ID, are unique identification numbers automatically created during a simulation to identify each individual instantiation of a Task or Element within a scenario. The second set of attributes, the Area Roto and the Area Roto Pointer, are related to the selection of an appropriate Area Rule. The next set of attributes, the Area identifier, the Unit Roto, and Unit Roto Pointer, are related to the selection of an appropriate Unit Rule. Finally, the last set of attributes, the Unit Identifier, the SubUnit Roto and the SubUnit Roto Pointer, concern selection of an appropriate SubUnit Roto Rule.

Generally speaking, each time the Roto Mechanism is prompted from an Activity in a particular Element, MARS algorithms determine whether the combination of Rotos and identifiers already exists in one of the records in the array. If the combination exists in a particular record, then the pointers in that record are used. If the combination does not exist then the associated pointers[32] are incremented and a new record is added to the list. More specifically, the construction and inherent use of this array is explained below in conjunction with Figure 32.

1. Finder nodes in each Activity are visited in sequence for the purpose of acquiring personnel to fill the corresponding positional requirements. The Roto Schedule ID on the Finder node is used to initiate use of the Roto Mechanism.

2. The unique IDs assigned to the Task and Element to which the Activity belongs are used to access the appropriate array.

[31] Note that in the first MARS prototype a new array is not created for each Element instantiation, instead new records are appended to a single array. Hence the reason for the Task and Element fields.

[32] Recall that a pointer is stored as part of each Unit and SubUnit Roto.

3. The array is searched to determine if there are any records with the current Area Roto. If one exists, the associated pointer in the array is used directly to find the appropriate Area Rule. If the Area Roto is not in the array however, then the Area Roto Pointer is incremented, and the combination of Task ID, Element ID, Area Rule, and pointer are added to the beginning of a new record in the array. The remainder of the record is filled in steps 4 and 5. The new pointer is used to determine the proper Area Rule from which the primary Area(s) is obtained.

4. The primary Area obtained in the previous step is used together with the prescribed Unit Roto Collection to find a particular Unit Roto. Once the Unit Roto is obtained, the array is once again checked. This time however, the purpose of checking the array is to determine if any record in the array holds the current combination of: Task ID, Element ID, Area Roto, Area, and Unit Roto. As in the previous step, if the combination already exists in the array, then the associated Unit Roto pointer is used. If the combination does not exist, then just as in the previous step, the Unit Roto pointer is incremented, and a new record is created that contains the combination and the new pointer and these are used to determine the proper Unit Rule. From the Unit Rule, the primary Unit(s) is obtained[33].

5. Each primary Unit obtained in the previous step is used together with the SubUnit Roto Collection to find a particular SubUnit Roto(s). Once the SubUnit Roto(s) is obtained, the array is once again checked. In this last step, the purpose of checking the array is to determine if the array holds the current combinations of: Task ID, Element ID, Area Roto, Area, and Unit Roto, Unit and SubUnit Roto. As in the previous two steps, if the combination exists in the array, then the associated SubUnit Roto Pointer is used. If the combination does not exist, the SubUnit Roto Pointer is incremented and a new record is created in the array. The new SubUnit Pointer is then used to find the appropriate SubUnit Rule. The SubUnit Rule is then used to obtain a list of primary and alternate subunits.

These steps are relatively straightforward. In each of steps 3, 4 and 5 the array is searched to find a specific combination of field attributes. In each subsequent step the number of field attributes in the combination grows. Step 3 concerns Area Roto pointers, step 4 concerns Unit Roto pointers, and step 5 concerns SubUnit Roto pointers. Each time a new and unique combination is added to the array one or more of the pointers on the associated rotos is incremented. The only caveat to this discussion is that the Area and Unit Identifier fields in the array are designed only to hold information pertaining to primary Areas and primary Units. The manifestation of this is that when an Area or Unit is identified as an alternate, any subsequent linkage to a Unit Roto or SubUnit Roto respectively does not cause pointer incrementation (*i.e.* Roto pointers are not incremented when the Roto is accessed to provide a rule to find alternate Units or SubUnits). This is a desired feature – the use of a Unit or SubUnit Roto to obtain alternates is not to impinge upon the selection of primary Units or SubUnits.

[33] It is important to note that the array only saves information about primary Units.

2.5.5 The Stealing Mechanism

The second mechanism used to select individuals to fulfill Resource requirements is called the Stealing Mechanism. As previously indicated, this mechanism allows individuals to be purposely acquired from an Activity that is currently in progress in order to participate in a concurrent Activity. As an example of how this mechanism may be used, recall that the MRS dictates that Army units are to engage in HR training in preparation for deployment. Currently the operational tempo within the Canadian Forces is very high and so once HR training is completed, immediate deployment is expected. However, even if an operational deployment is not scheduled, the MRS is designed to produce HR units in preparation for potential deployment. The MARS stealing mechanism is expected to be of great utility for modeling “just in case” or “on-call” scenarios like these.

An example that makes use of the Stealing Mechanism is illustrated in Figure 33 where a network of Activities which collectively represent HR training are built into the Task-Template. Most of these Activities make use of the Roto Mechanism to fill Resource requirements. As expected, these Resources flow through the Activity network in the usual way. One of the most prominent Activities in this example is the collective training exercise at CMTC. From this Activity, resources could be modeled to flow directly to a deployment Activity. However in this case, instead of being passed immediately to a deployment Activity, they are sent to what is known as a "Reservoir Activity". A Reservoir Activity is an Activity with an RUL of 0 that is intended to serve as a holding queue for Resources that are earmarked for potential reassignment to other Activities. Based on the current MRS, this Reservoir Activity could be defined to have a duration of 6 months, after which the individuals filling these resources would be released back into the establishment. By sending Resources to a Reservoir Activity, HR training and deployment can be modeled as separate Tasks each with their own Elements, Activities and instantiation schedules. With this separation, new HR training Tasks could be instantiated every 6 months and deployment Tasks instantiated on a random basis.

In the most basic case, a deployment Task could be composed of just a single Activity with several Finder Nodes. Each Finder Node would be linked to a Resource requirement and a series of attributes that would enable it to select individuals to fill these Resource requirements from an appropriate Reservoir Activity. The methodology used to select the Task, Activity, and Resources from which to acquire individuals, is the Stealing Mechanism.

Like the Roto mechanism, the Stealing mechanism permits the prescription of a series of primary and back up locations from which to draw potential individuals to fill a Resource requirement. In this context a location may refer to a Task, Activity, or Resource. In general, if the location is a Task (or set of Tasks), then any Resource attached to any Activity within the Task may be selected to supply individuals to fill a demand. If the location is an individual Activity or set of Activities, then only the Resources within those Activities become available to transfer individuals. Likewise, if the location refers to a particular Resource or set of Resources, then only individuals from those Resources are selectable. The concept behind the design of the Stealing Mechanism is to allow the algorithms to select the appropriate location from those which are currently active within a simulation.

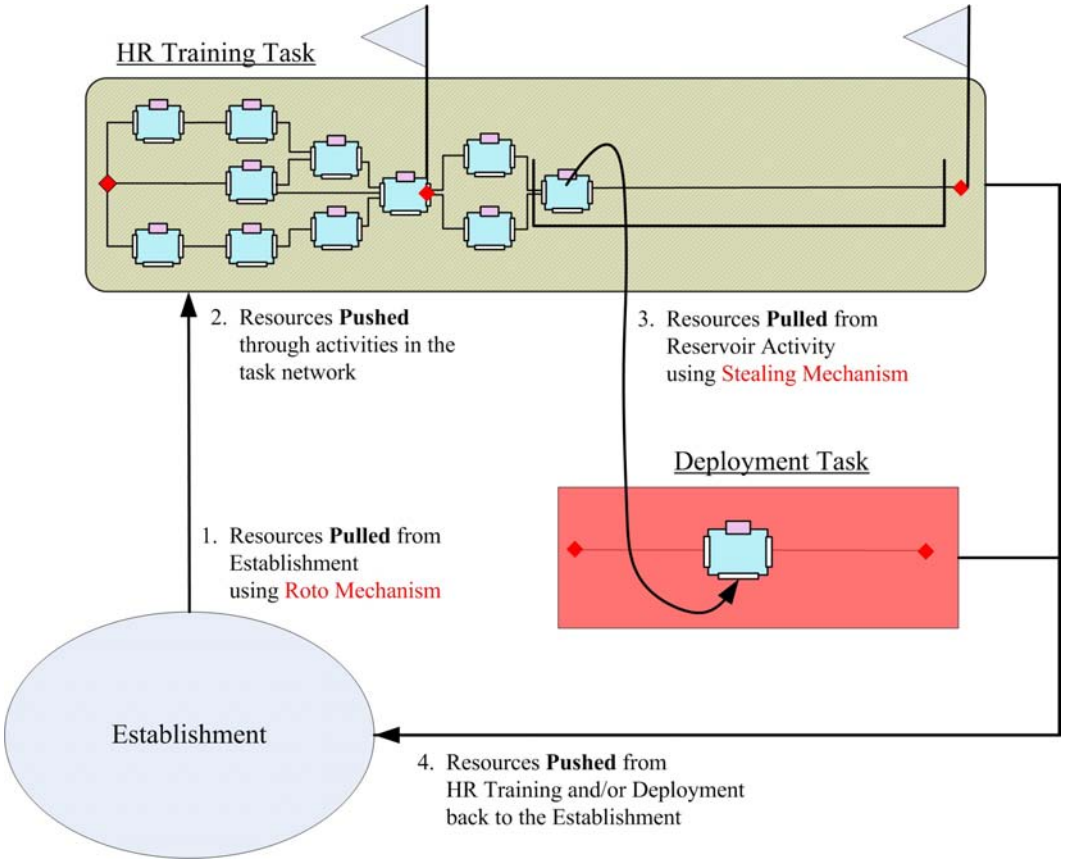


Figure 33 : *A Stealing Mechanism example. Resources are first "pulled" from the establishment to participate in HR Training using the Roto Mechanism. Resources are then "pushed" through the Activity network until they are deposited into the reservoir Activity. A selection of Resources are then "pulled" from the reservoir Activity to the Deployment Task using the Stealing Mechanism. Finally they are pushed back to the establishment.*

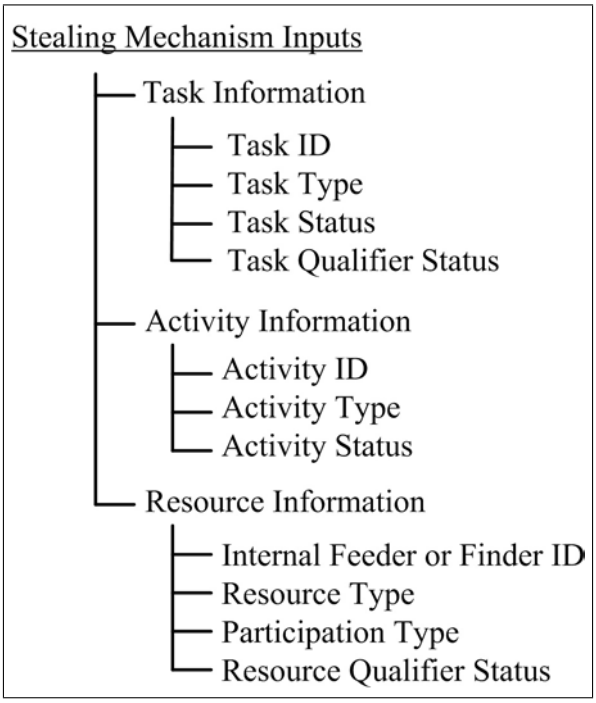


Figure 34 : *Stealing Mechanism Inputs. Note that not all of these need be specified. When there is no input for a particular attribute category it is assumed that any attribute value in that category is acceptable.*

The set of attributes that may be defined to demarcate the specific Task(s), Activity(s), or Resource(s) are shown in Figure 34 and described separately below. When the Stealing Mechanism is used, each of these attributes can be linked to a specific Finder Node to enable the algorithms within MARS to search out the proper location(s).

Task Template ID: Each Task-Template is prescribed a unique ID when it is defined and every Task instantiated from that Task-Template also records this ID. As part of the Stealing Mechanism, the Task-Template ID can be matched to find all current Tasks within a simulation that are generated from the same Task-Template.

Task Type: The Task Type attribute is also defined as part of the definition of a Task-Template. MARS enables the user to define Task types and optionally assign these to specific Task-Templates. Tasks instantiated from these templates take on the Task Type attribute. Many Task-Templates can be prescribed the same Task Type. Prescribing a Task Type as part of the Stealing Mechanism enables specific types of Tasks to be selected.

Task Status: During the lifetime of a Task it can be concurrently assigned status levels. Each level is associated with a different status type in a family of Generic Task Statuses. One status type in the family of Generic Task statuses is the Task Progress Status. Levels of Progress Status are automatically assigned by MARS algorithms as

a Task progresses. For example, one Progress Level indicates the successful instantiation of a Task while other levels correspond to successful completion, premature termination, *etc*. Any of these Progress Status Levels can be provided as an input to the Stealing Mechanism to aid in the identification of a particular Task or set of Tasks in a simulation. However, in as much as these are expected to be useful, it is expected that Status Types of more common use will be those that are alterable by Activities (section 2.4.4).

Recall that an Activity can be defined to alter the status of a Task upon its commencement or termination. These Activity-alterable Task statuses can be particularly useful for demarcating Tasks which contain reservoir Activities that are available to act as a supply of Resources for other Activities. To make use of this feature, a Reservoir Activity could be defined to alter a particular Task Status to a predefined level. This level would then become a useful input to the Stealing mechanism to identify Tasks with available Resources. At the termination of the reservoir Activity the status level of the Task could be reset to signify that the Resources are no longer available.

Task Qualifier Status: The Task Qualifier Status is an attribute assigned to a Task as part of the schedule used to instantiate the Task. If the Task instantiation schedule is an additive composite of several scheduling formulae, each of the formula within a single schedule can impart a different level of Qualifier status to the Tasks that it instantiates. Unique Qualifier Statuses can be defined by the user to achieve different effects. In essence, the Task Qualifier Status is a "user-special" status that enables additional flexibility for identifying a particular Task from all of those that might be current within a simulation. A practical use of the Qualifier status would be to demarcate Tasks that are instantiated from the same Task-Template, but which are instantiated from two formula that correspond to different "lines of operation". The use of each formula would impart a different Task Qualifier level to the Task upon its instantiation. The two different Task Qualifier Status levels would differentiate between two different lines of operation.

The set of attributes associated with the identification of an Activity or set of Activities to provide individuals to fill a demand are: the Activity ID, Activity Type, and the Activity Status. These attributes are similiar in nature to those defined for a Task.

Activity ID: As with each Task-Template, every Activity is prescribed a unique Activity ID when it is created as part of a Task-Template. This Activity ID can be specified as one of the input attributes for the Stealing Mechanism to identify any Activity created using the same Activity definition.

Activity Type: The Activity Type attribute is defined as part of the definition of an Activity. Many Different Activity Types can be defined and optionally assigned to Activities within a Task-Template. Several Activities can be assigned to the same Activity Type. When used as an input to the Stealing Mechanism, the Activity Type facilitates the preferential selection of a particular kind of Activity from all others.

Activity Status: Just as for a Task, an Activity may be assigned to one of several Activity

Status levels; each level belonging to a particular status type in a family of Activity Statuses. One of these Status types is the Progress Status. Successive levels of progress status are assigned to a Task as it progresses through its lifecycle. The lifecycle of an Activity is depicted in Figure 16 in section 1.5.2 and is the subject of further discussion in section 3. Levels of Progress Status coincide with the successful passage through each station[34] in the lifecycle. Progress status levels are available to be used within the Stealing Mechanism to select Activities that are at the same stage in their lifecycle. Activities may also be selected based on other status types in the family of Activity Statuses. In general, these statuses would be defined as required by the user, prescribed to an Activity as part of its definition, and then used as input to the Stealing Mechanism to facilitate identification of a particular Activity or set of Activities during a simulation.

The set of attributes associated with the selection of an active Resource or set of Resources in order to provide individuals to fill a demand are: Resource Type, Participation Type, Resource Qualifier ID, and Feeder/Finder ID.

Internal Feeder or Finder ID: Each Resource within an Activity can be associated to the Feeder or Finder node through which it enters an Activity. As mentioned previously, these Feeder and Finder nodes are prescribed a unique ID number when an Activity is defined. This unique ID number can be used within the Stealing mechanism to identify a particular Resource within an Activity.

Resource Type: The Feeder and Finder nodes within an Activity serve to accept three basic types of resources: personnel, materiel, and facilities. Within each of these basic categories MARS allows for the definition of subtypes. These Resource Type and Subtype attributes may be used to identify particular resources to the Stealing Mechanism. The type and subtype of a Resource can be reassigned as the Resource "flows" between different Activities. The initial assignment of a subtype to a particular Resource occurs as it enters a Finder Node of an Activity. Subsequent assignments may occur as it exits the Activity and enters another Activity through a Feeder Node[35].

Participation Level: As described in section 2.4.7, each Resource can be assigned a participation type attribute when it takes part in an Activity. Within the Stealing mechanism, the Participation Level can be used to identify a subset of Resources participating in a particular Activity.

Resource Qualifier Status: As with the Qualifier attribute, described above for a Task, the Resource Qualifier Status is a "user-special" status that enables additional flexibility in the selection of a particular Resource. However, unlike the Task Qualifier Status, it is not associated with a particular instantiation schedule. As with many of the other

[34]Each level of Progress Status indicates a degree of success towards the completion of an Activity. Specific progress levels correspond to: Activity instantiation, removal of the Activity from the global hold queue, successful acquisition of resources, completion of Resource processing, Resource release, and Activity termination.

[35]The design of the first MARS prototype does not allow for the reassignment of Resource Type as part of a Next-Activity Node. This feature may be added as required to future versions.

attributes, it is assigned using the Feeder or Finder node through which a Resource enters an Activity. The Resource Qualifier status is to be used to tag resources taking part in particular Activities in a way that would make them particularly identifiable to the Stealing Mechanism. As an example, it might be used to demarcate a Resource in an Activity as "on-call" for a NEO. A special Resource Qualification Level could be defined for this purpose. Upon application of the Stealing Mechanism, the Qualification level would allow for the identification of the on-call Resource for assignment of individuals to a NEO deployment event.

In the first MARS prototype the Resource-related attributes intended to identify particular Resources to the Stealing Mechanism are limited to those described above. Many other attributes are assigned to Resources as they flow through a Task. Future versions of MARS may incorporate design enhancements to use a larger collection of Resource related attributes depending on the perceived need for this functionality. However, in light of this, it should be noted that the Stealing mechanism has not been fully implemented within the first MARS prototype.

As a final note regarding the Stealing Mechanism, this mechanism is not the only means by which individuals can be removed from a Resource that is participating in an Activity. The Roto Mechanism can also serve to acquire personnel already participating in other Activities, albeit in an indirect way. Generally speaking, the main difference between the Roto Mechanism and the Stealing Mechanism is that the Roto Mechanism serves to directly target the establishment, equipment fleets, or facilities infrastructure to acquire resources, whereas the Stealing Mechanism directly targets concurrent Tasks, Activities and Resources to acquire individuals. A concept important to both of these is that when individual personnel belong to a Resource that is participating in an Activity, those individuals still retain their position within the establishment[36] [37].

2.5.6 Setting Resource Requirements & Selection Schedules/Rules within an Activity

The discussions in the previous two sections elaborate on issues regarding the determination of Resource Requirements and the definition of Resource Selection Schedules. To recap, a Resource Requirement is an indication of "what" is required to compose a Resource. A Resource Selection Schedule is a set of rules for determining "where" to find individual persons, equipment, or facilities to fill a Resource. The amalgamation of a Resource Requirement with a Selection Schedule/Rule occurs as part of the definition of each Finder Node in an Activity.

[36] Note that in order to remove an individual that is already part of a Resource participating in another Activity(s), a series of attributes are checked to ensure that certain requirements are satisfied. For example, an individual participating in a high importance Activity and can not be removed to participate in an Activity of lower importance.

[37] An issue of interest concerns the "re-filling" of resources where individuals have been removed for participation in other Activities. In the first MARS prototype resources cannot be refilled, however this issue may need to be revisited in the future when more complete scenarios need to be modelled.

- A Resource Requirement is specified in terms of an ORBAT and a series of status requirements.
- A Resource Selection Schedule can incorporate both the Roto Mechanism and the Stealing Mechanism in any prescribed order. For example, a selection rule may indicate that individuals should first be acquired from a particular set of Activities and then, if positions still remain unfilled, candidates from particular locations within the establishment should be solicited.

Ultimately, it is the amalgamation of Resource Requirements and Resource Selection Schedules on each Finder node that determine the utilization of resources across all Tasks in a scenario. It is for this reason that special care needs to be taken when defining these rules in complex scenarios. Over-prescribed Resource requirements will never be filled and incorrect selection schedules will not mirror how resource stocks are accessed in practice. With this in mind, an important characterization of the first MARS prototype is that it is completely descriptive. Among other things, it does not contain any special prescriptive algorithms to automatically define and/or manage the matching of demands with resource supplies. This is an intentional aspect of the first prototype. As military partners obtain a degree of comfort and trust in MARS, the development of sophisticated routines to automatically produce different demand options and/or scheduling alternatives at the expense of application transparency may be merited.

2.6 Closure

Discussions in this section are intended to explain MARS inputs, facilitate an understanding of MARS design concepts, and enable a user to define scenarios[38]. It is recommended that the construction of a new scenario begin with a definition of each Task-Template in terms of its hierarchical breakdown into Elements, Element-Options, and a network of Activities. After the hierarchy is defined, attributes that relate specifically to Tasks, Elements, and Element-Options can be prescribed. The third step is the definition of Task Instantiation Schedules and Element-Option Schedules. The fourth step is the definition of certain Activity attributes including the temporal and routing attributes. This step is linked with defining the resource supply, demand requirements, and resource acquisition schedules. The Final step is to assign and link Resource requirements and selection schedules to individual Finder nodes on each Activity.

Evidently the process of prescribing MARS input can be as simple or as involved as is dictated by the complexity of the scenario to be modeled. Moreover, it should be noted that the same scenario may be modeled in different ways. This follows from the fact that a major thrust in the development of MARS has been to provide the user with a series of constructs to model scenarios that cover a full spectrum of aggregation and sophistication. The subordinate parts of a Task-Template embody these constructs. These constructs are intended so that the user does not need to have direct access to the simulation kernel in order to make intricate changes to a simulation. To the degree that this goal can not be

[38]These concepts provide the foundation upon which the MARS graphical user interface is designed.

realized to model scenarios required by military partners, the first MARS prototype will be upgraded. Several potential avenues for upgrade have been noted in the discussions in this section. Many of these correspond to the extension of concepts that already have a seed in the first MARS prototype. A brief list of other issues on the horizon for future consideration are listed in section 4. In the next section an outline of the algorithms that instantiate Task-Templates and manipulate Activities is presented.

3 The Activity Lifecycle

In section 1.5 it was explained that MARS is designed based on two levels of abstraction. The first level of abstraction, detailed in the previous section, embodies the idea that Tasks are actually networks of Activities that set the demand for Resources. These Tasks are instantiated through the use of Task-Templates. The first level of abstraction may be considered to be the framework for organizing user input within MARS.

The second level of abstraction, which is the topic of this section, embodies the internal processes by which Task-Templates are instantiated and manipulated to forecast resource utilization and assess the sustainability of resource pools based on a particular set of scheduling, allocation and policy concepts. As introduced in section 1.5.2, at the heart of the second level of abstraction is an Activity-Station-Lifecycle paradigm. Within this paradigm the relationship of an individual Activity to others in the network of a Task is reflected in the way that it moves from station to station in a lifecycle. A representation of this Activity lifecycle is depicted in Figure 35. It is composed of 5 stations. A brief explanation of each station follows in the section below.

3.1 The Lifecycle of an Activity – An Overview

Station 1: Task Generation At the first station, denoted as “Task Generation”, a Task is created through the instantiation of a Task-Template. In this step all of the Activities that make up the Task are instantiated. Task instantiation, otherwise known as Task generation, is effected through the use of Task instantiation schedules and Element-Option schedules over the course of a simulation.

Station 2: The Global Wait Queue Upon the instantiation of a Task, each Activity is placed inside a queue to await processing. This queue is called the "Global Wait Queue". The Global Wait Queue is the second station visited by an Activity in its lifecycle. Depending on the number of Activities that define each Task and the frequency at which tasks are instantiated, a multitude of Activities belonging to different Tasks may be present in the Global Wait Queue at any one time. Each Activity contains a myriad of attributes that serve to control its behaviour in relation to other Activities. Specifically, Activity timing and routing attributes are used to control the release of Activities from the queue so that the logical dependence between Activities is maintained.

Station 3: Resource Acquisition Once released from the Global Wait Queue, the selection and assignment of Resources for participation in an Activity is carried out at station 3 in the lifecycle. When compared to the rest of the lifecycle, the processes encompassed at this station are the most comprehensive.

Each Activity can contain multiple Resource requirements and each Resource requirement can contain a collection of positional requirements. Individual Resource requirements are processed according to a user-prescribed ordering scheme. As each Resource requirement is processed, the selection of individuals occurs in three steps.

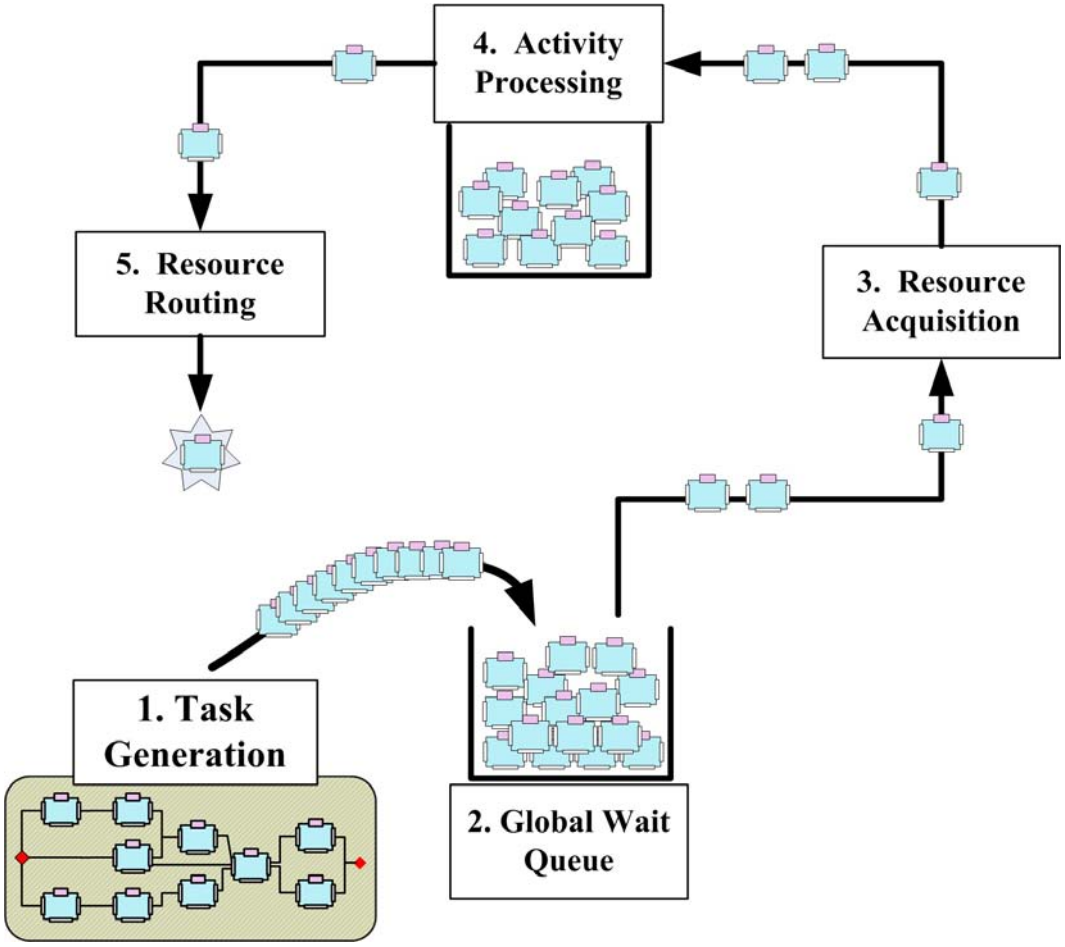


Figure 35 : *A Pictorial View of the Activity Lifecycle. Activities "flow" from station to station in the Lifecycle. An overview of this lifecycle is presented in section 3.1.*

First, positions in the Resource requirement are sorted so that those that are most important and/or more difficult to fill are at the top of the list. Then, the list of candidate personnel identified to fill the positions in the Resource are ordered so that those types of individuals who are most plentiful, most common, or have non-specialist attributes, are at the top of the list. Finally, positions are filled in sequential order. For each position, the list of candidate individuals is traversed and assignments are made based on a comparison between the particulars of the position requirement and the attributes of the candidate.

In the first MARS prototype individuals selected to fill Resource requirements are assigned according to the order that Activities enter this station. However, it is important to note that the assignment of an individual to a position within a Resource may preempt a previous assignment based on a comparison of Activity Importance Levels. In other words, an individual assigned to a position in a Resource participating in a low priority Activity can be reassigned to a position in a Resource in an Activity with a higher priority if required.

Station 4: Activity Processing After station 3, each Activity flows to the Activity processing station where individuals in resources that participate in the Activity are tagged with one or more statuses. The Activity itself remains in a Processing Queue for a duration that is dependent on its timing attributes. After the Activity is complete it is released from the Processing Queue and statuses assigned to the Activity and participating resources may be set (or reset).

Station 5: Resource Routing & Activity Dissolution The last station in the lifecycle of an Activity provides the means to feed Resources to subsequent Activities are residing in the Global Wait Queue. As indicated previously, the specification of Activity dependence is explicitly defined via the Task network and attributes internal to each Activity. All of the resources participating in an Activity may be earmarked to be fed to a single Activity or alternately different resources can be forwarded to different Activities. After an Activity forwards resources to dependent Activities, it is finally dissolved.

When a Resource is passed to another Activity it is assigned to take part in that Activity once that Activity is released from the global wait queue. The release of the dependent Activity from the Global Wait Queue occurs based on its Task timing attributes and a determination about whether it has been fed an adequate supply of resources from its prerequisite Activities.

3.2 Viewing the Activity Lifecycle from the Perspective of a Resource

From the perspective of an Activity, the lifecycle in Figure 35 is self evident; however it is also instructive to view the Activity lifecycle from the viewpoint of a individual resource.

Station 1: Task Generation From the viewpoint of the personnel, equipment, and facili-

ties that represent the resource supply, the generation of Activities that occurs during Task instantiation at the first station in the Activity lifecycle represents the creation of a set of Resource demands (*i.e.* requirements).

Station 2: The Global Wait Queue Activities in the Global Wait Queue at the second station may or may not have Resources assigned to them. For example, Activities dependent on other Activities that have already been completed will usually have Resources assigned to them. Activities in the Global Wait Queue that are not dependent on prerequisite Activities will not usually have Resources already assigned to them.

Station 3: Resource Selection At the Resource selection station, Resource requirements for which individuals have not yet been allocated are identified, and together with Resource allocation schedules, individuals to fill Resource requirements are selected and assigned. In the current implementation, individuals may be selected to fill Resource requirements based on their position within the establishment or another Task[39]. It is at this station where the "borrowing" of resources occurs between establishment units according to the rules set out in the Resource Selection Schedule.

Station 4: Activity Processing At this station individuals in a Resource that have been previously assigned to an Activity are seized. After they are seized they are delayed and their status is altered at the processing station.

Station 5: Resource Routing & Activity Dissolution Finally, at the last station individuals in a Resource are either assigned to a dependent Task already resident in the Global Wait Queue or they are released back to the establishment, equipment fleet, or facilities pool.

Viewing the Activity lifecycle from the perspective of either an Activity or a Resource is somewhat akin to viewing the MRP from the perspective of a Force Employer and Force Generator respectively. From the Force Employer's perspective the major focus is the completion of tasks (akin to Activities). From the Force Generators perspective the major focus is on the effective management and utilization of resources. Understanding issues from both perspectives can be equally illuminating. With this in mind, it is noted that the depiction of the simulation methodology presented in Figure 35 is distinctly biased toward a single viewpoint – that of Activities. It has been presented as such because this is the viewpoint upon which the design of MARS has been anchored for implementation within a discrete event simulation environment.

More details pertaining to each of the stations in the Activity lifecycle are provided in the remainder of this section. In keeping with the aim of this report, the scope is limited to the conceptual ideas surrounding the design methodology intended to govern processing at each station. Brief descriptions of the relational data structure used to store scenario inputs and record scenario outputs are also included. These should be accessible to a more

[39]The ability to set the location to select individuals to fill a Resource requirement and mimic a kind of purposeful "stealing" from Activities already in progress (*i.e.* the Stealing Mechanism of section 3.5.4) has been designed but is not implemented in the first MARS prototype.

advanced MARS user wishing to gain further insight into the simulation kernel and the general algorithmic rules behind its operation without delving into programmatical details[40].

3.3 Station 1: Task Generation/Instantiation

The ability of MARS to forecast forward in time based on the repeated instantiation of similiar Tasks is one of the attributes that sets it apart from other project management software. The length of a scenario is an implicit function of the Task Instantiation schedule associated with each Task-Template in a scenario.

The relational data structure used to store Task Instantiation Schedules is shown in Figure 36. In this figure, as with all subsequent figures in this section that describe data structure, the arrangement of the tables is based on standard relational database symbology. The field names for each table are listed. The linkages between tables, symbolized via the connecting lines, take on "one-to-many" or "many-to-one" attributes. The small circle at the end of each linking line indicates the "many" part of the linkage. One aspect of note with respect to the use of the relational structure within MARS is that special allowance has been made to account for the fact that MARS kernel algorithms are encoded using the ARENA simulation language, and while this language is robust, it can not currently use alphanumeric information. Only integer data are allowed. As such, there are some cases where alphanumeric objects are dually defined with integer codes to facilitate use inside ARENA.

A single Task-Template is represented by a record within the *Generic Tasks* table and is linked to an instantiation schedule in the *Task Generation (Instantiation) Schedules* table which can be an aggregate of several instantiation formulae[41]. Particular formula parameters for each instantiation formula are stored in the *Generation (Instantiation) Formula* table. Alphanumeric parameter definitions are contained in the *Formula Definitions Table*. Some parameters of particular interest in Figure 36 are as follows.

First Instantiation Time: This parameter prescribes the first simulation date that a Task-Template is instantiated. The process of instantiation is that of taking all Activities contained within a Task-Template and placing them into the Global Wait Queue. A First instantiation Time of 0 for example, would indicate that the first instantiation would occur at the start of the simulation.

Level 0 Anchor Time: The base anchor time of a Task, called the Level 0 Anchor time, is the difference between a particular milestone in the Task-Template and the time when it is instantiated. For example, in the case of a HR Training Task which contains a network of many incumbent Activities, an appropriate milestone might be the conclusion of collective training at CMTC. A Level 0 Anchor Time of 500 days would indicate that the expected completion of training at CMTC is 500 days after Task in-

[40] For programmatical details, the reader is referred to the third companion report in this series [5].
[41] Recall that instantiation formulas can be either deterministic or stochastic.

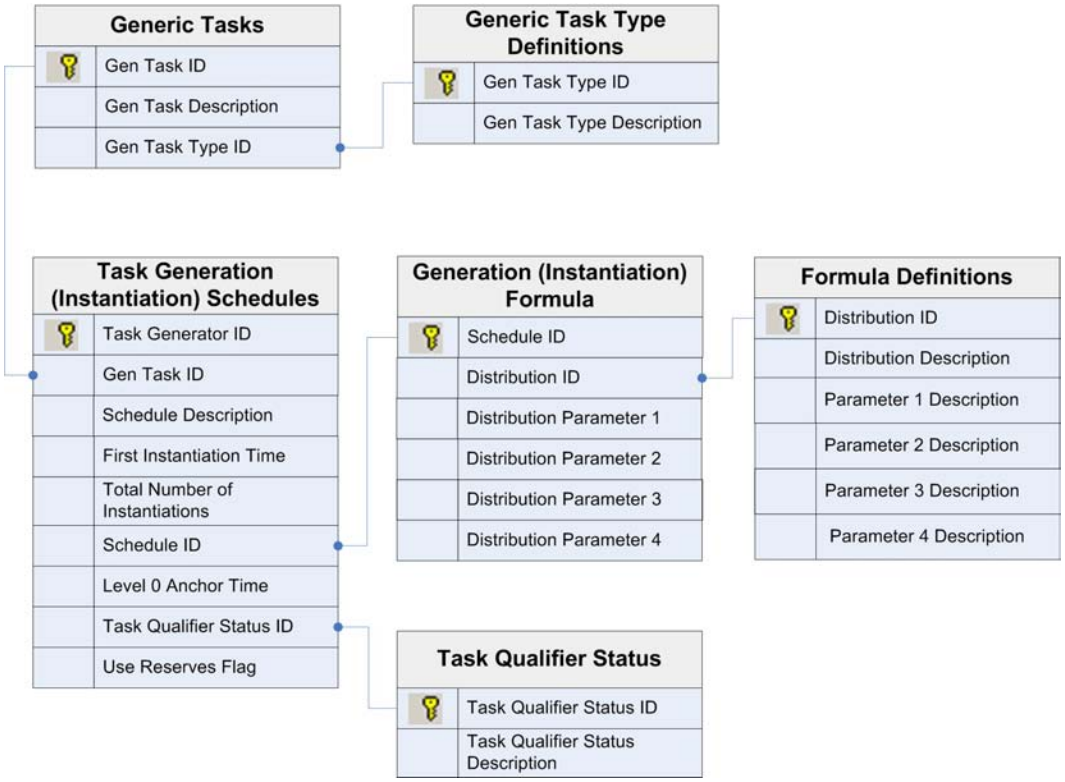


Figure 36 : *The relational data structure used to store Task Instantiation Schedules in relation to Task-Templates (see section 3.3.)*

stantiation[42]. In the first MARS prototype, the nominal release date of an Activity (with reference to the start date of the simulation) is prescribed upon instantiation of the Activity itself. The relationship between the Level 0 Anchor Time and other time related parameters discussed in section 2.4.3 is illustrated in Figure 37. In essence, two absolute dates are assigned to each Activity as they enter the Global Wait Queue: the Start No Earlier Than date (SNET) and the Start No Later Than date (SNLT) introduced in section 2.4.3.

Total Number of Instantiations: The Total Number of instantiations determines the total number of times that the Task-Template is instantiated within a simulation.

Use Reserves Flag: In the first MARS prototype there exists an option to automatically generate reservists to fill Resource requirements. This flag turns this option on or off.

Task Type: In the first implementation of MARS the user is provided the ability to create

[42]In future versions of MARS where the allocation of resources for all Activities might be computed many months in advance of the start of a Task, the LvL0 anchor time is expected to be of importance for helping to prescribe Task priority. That is, tasks that are instantiated early, may be afforded greater latitude in how resources are assigned.

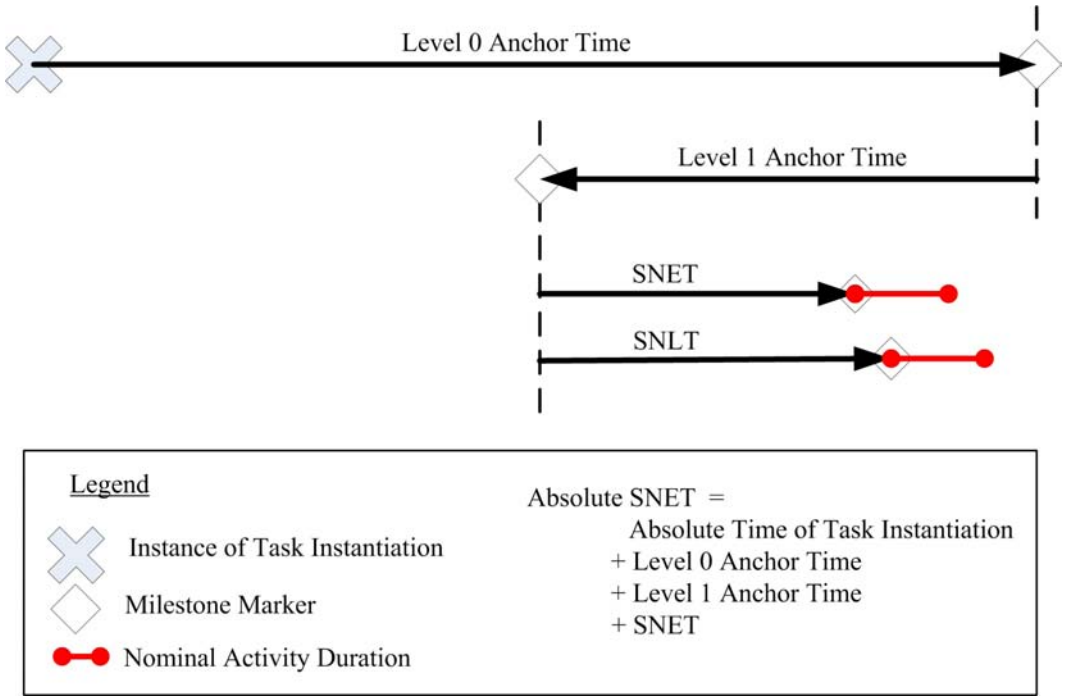


Figure 37 : *The relationship between different timing parameters within MARS. Note that the Time of Task Instantiation is defined as part of the Task Instantiation Schedule (section 2.2). It is specified in terms of the actual simulation time measured in days. As shown, other timing attributes (section 2.4.6 are defined relative to one another. They are added to the Time of Task Instantiation to compute the actual "Start No Earlier Than" (SNET) and "Start No Later Than" (SNLT) times for an Activity.*

new Task-Types and assign one of these to each Task-Template. Task Type designations are expected to be of particular utility when using the Stealing Mechanism. They are also useful during the post processing of simulation results.

Task Qualifier Status: As described in section 2.5.5, a Task Qualifier status may be assigned to a Task depending on the formula used for its instantiation. As with the Task Type, the Task Qualification Status is expected to be of particular utility when applying the Stealing Mechanism and/or analysing simulation results.

The creation of individual Activities upon the instantiation of a Task-Template is accomplished by accessing a series of tables that store the Task-Template hierarchy. These tables are illustrated in Figure 38. From the figure it is evident that the hierarchy of Elements, Element-Options and Activities that compose a Task-Template is spread across seven tables. Four of these are “definition” tables and three of these are “junction” tables[43]. The field parameters in these tables are described in section 2. The relational structure between

[43] A junction table is used whenever there exists the need to represent a many-to-many relationship between data fields in a relational database.

the tables is designed to allow a Task-Template to consist of more than one Element while, at the same time, an Element can exist as part of more than one Task-Template. The same kind of "many-to-many" relationship exists for each of the other layers in the hierarchy.

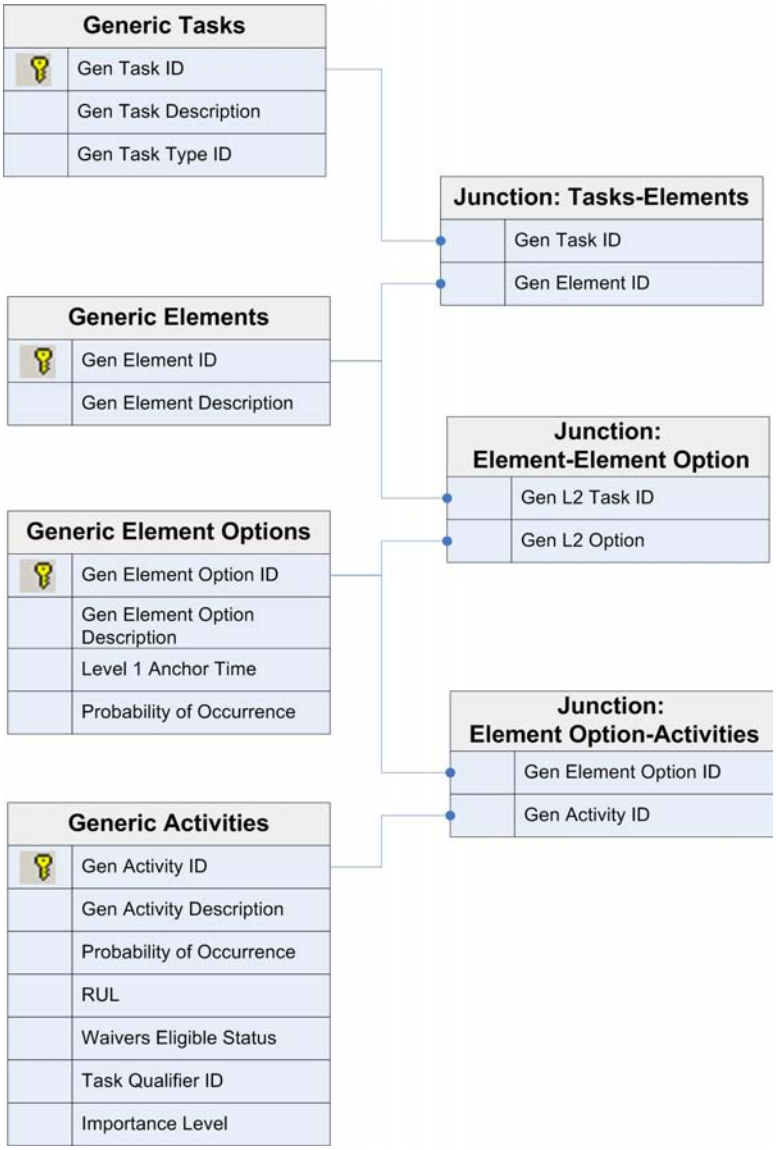


Figure 38 : *The relational data structure used to store the Task-Template hierarchy (*i.e. *Tasks, Elements, Element-Options & Activities).*

A point of interest in the tables of Figure 38 is the term "Generic". This term is used to differentiate between tables used to store information about Task-Templates and tables to be described later in this section that contain information about specific instances of a Task, Element, or Activity. In contrast to the term "Generic", information about a particular instance of a Task, Element or Activity is prefixed by the term "Specific" within the MARS

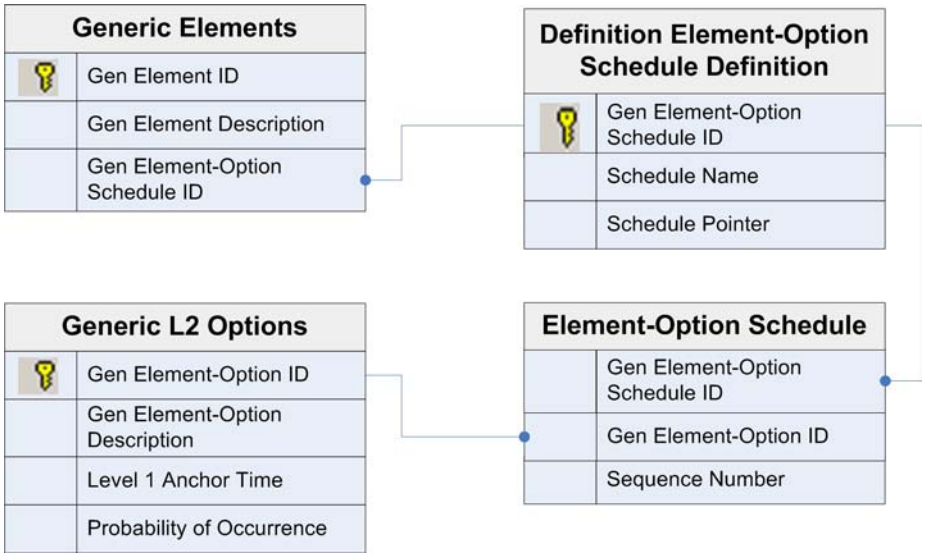


Figure 39 : *The Relational Data Structure used to store Element-Option Schedules in relation to Elements.*

data storage structure.

The relational structure used to store Element-Option Schedules for the purpose of determining the order that Elements employ Element-Options to instantiate particular Activities is illustrated in Figure 39. Recall that an Element-Option Schedule is a simple sequence of Element-Options where the sequence order determines the order that Element-Options are instantiated. Within the relational structure in the figure, the Element-Option Schedule table is a junction table that serves to link an Element-Option Schedule defined in the Element-Option Schedule Definition to each Element-Option in the Generic L2 Options table.

An important field parameter in the Element-Option Schedule Definition table is the schedule pointer. This pointer prescribes the starting point for instantiating Element-Options in an Element-Option Schedule. This pointer is read into the MARS simulation kernel at the start of a simulation and incremented accordingly as it proceeds[44].

An important part of Task Instantiation is the generation of individual Activities themselves. As is evident from discussions in section 2, the data requirements to define an Activity can be quite extensive. The relational layout of these requirements are illustrated in Figures 40 through 42. With few exceptions, these requirements are assigned upon instantiation.

Illustrated in Figure 40 is the data structure for Activity attributes that are categorized in section 2.4 as either General, Temporal, or related to the alteration of Task Status. Activity

[44]It should be noted that the Element-Option Schedule Pointer is not rewritten to the data structure upon completion of a simulation. This option may need to be incorporated in future versions of MARS if there is a need to stop a simulation, write out intermediate data, and then restart the simulation again.

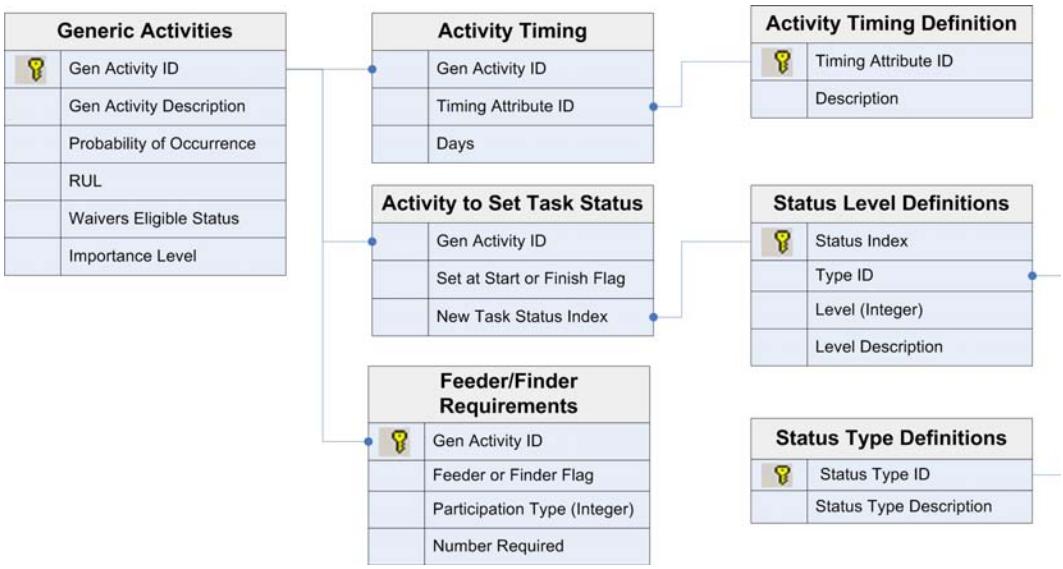


Figure 40 : *The relational data storage structure for Activities. Part 1: General & Temporal Attributes.*

Timing and Status Type and Level Definition tables are also included in this figure. With regard to the latter, it is noted that the family of progress statuses for both Tasks and Activities are defined and stored in the same tables.

Figure 41 illustrates how internal and external routing information is shared between attributes attached to Feeder, Finder and Next-Activity nodes.

Figure 42 shows how particular status requirements are attached to individual finder nodes. The relational layout of tables used to store definitions of Doing-Statuses and B&E Statuses is also shown in Figure 42.

To close this discussion about the instantiation of Tasks and the generation of individual Activities at the first stage in the Activity Lifecycle, it is important to note that the algorithms within MARS are designed to track the progress of these Activities as they move from stage to stage. Not only is Activity tracking of use when debugging a scenario but it is also provides a mechanism by which MARS is able to find Activities that are in a particular state. As discussed in section 2.5.5, this can be of particular relevance to the Stealing Mechanism but it has other applications as well. For example, Activities that remain incomplete for any of a variety of reasons may be counted as a measure of performance. The relational data structure used to store information about the instantiation of individual Activities, as well as the Elements and Tasks of which they are a part, is presented in Figure 43. As an Activity moves from station to station in the Lifecycle new entries are recorded in the corresponding tables. The tables in this figure are also linked to tables (to be described later in section 3.5.5) that hold information about the movement of Resources from Activity to Activity.

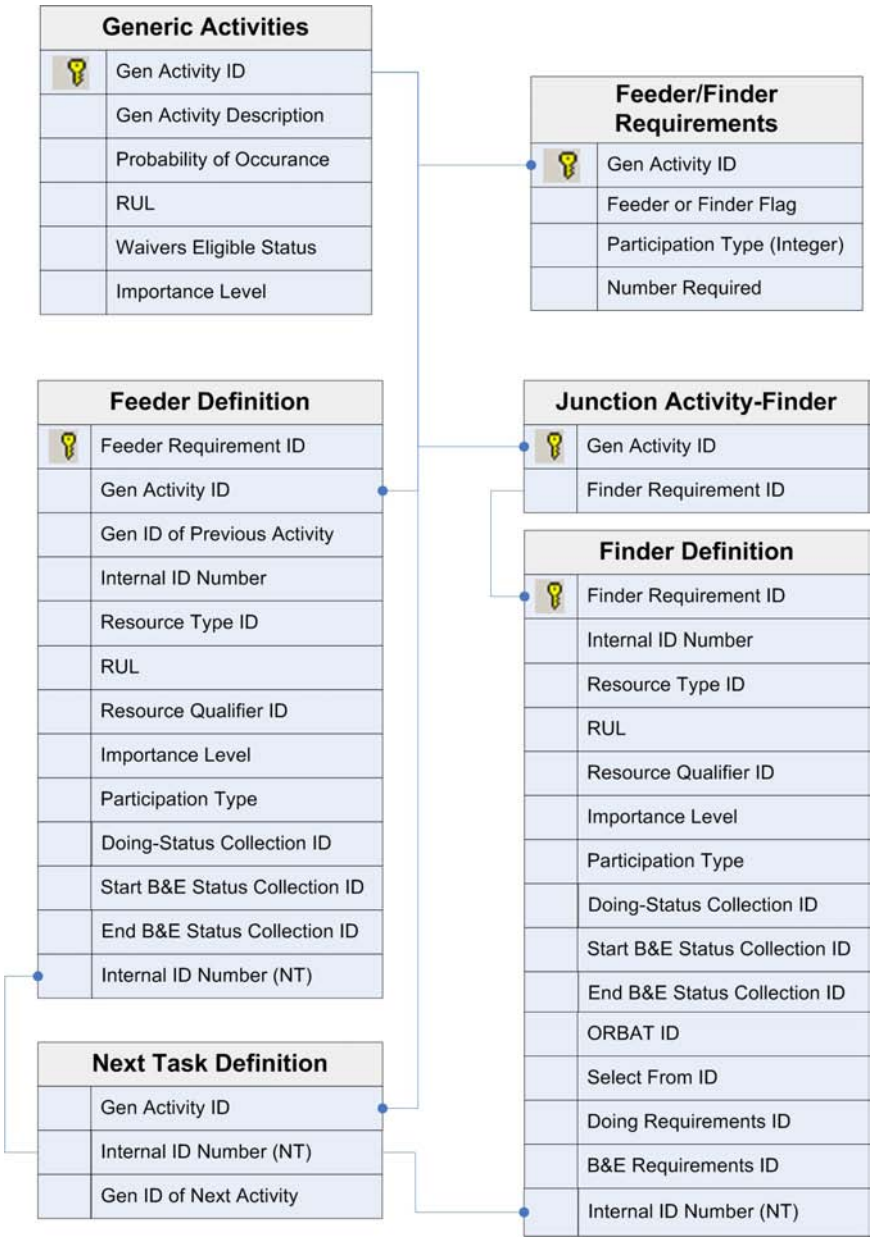


Figure 41 : *The relational data storage structure for Activities. Part 2: Internal & External Routing Attributes.*

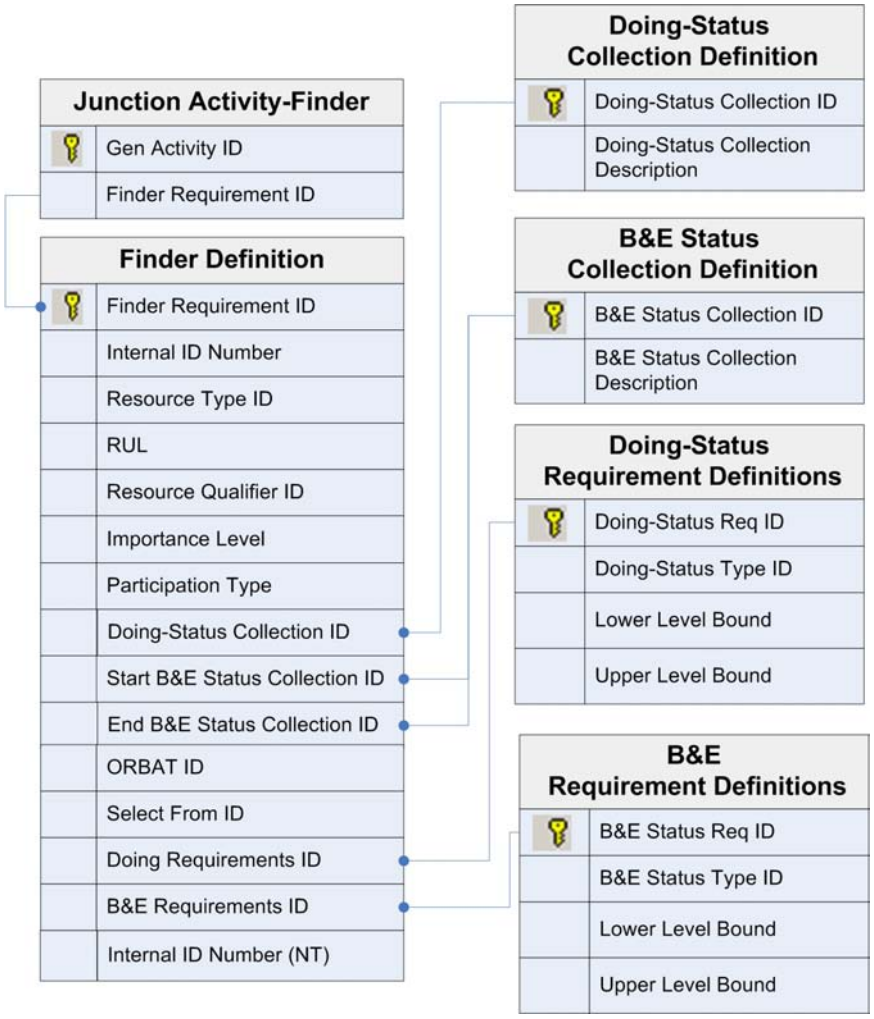


Figure 42 : *The relational data storage structure for Activities. Part 3: Status requirements are part of the Resource requirement attached to each finder node.*

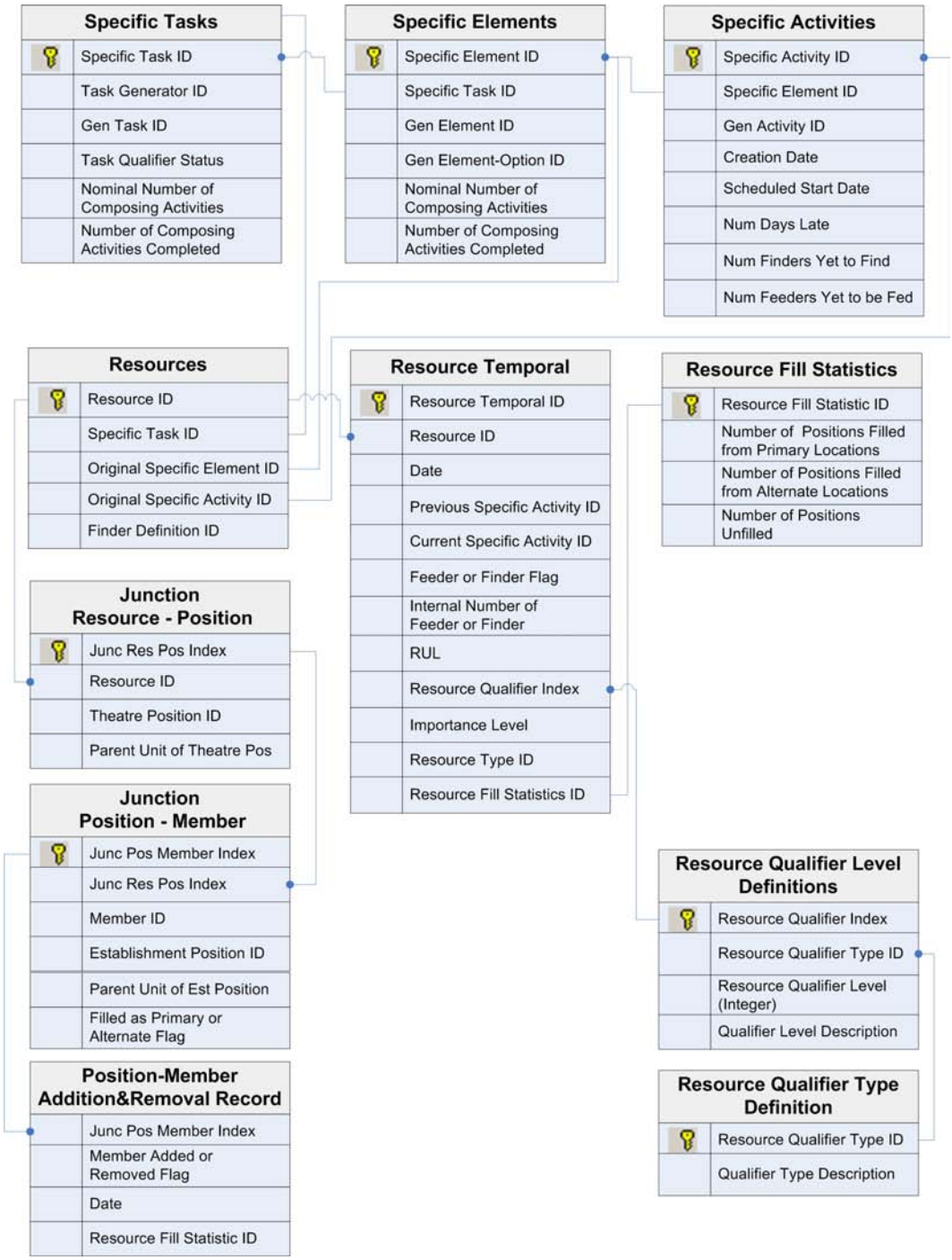


Figure 43 : *The relational data storage structure for specific instances of a Task, Element and Activity. As an Activity moves from station to station in its lifecycle, progress is tracked by recording new entries in each of the associated tables. Tables in this figure are also linked to those presented in Figure 52 which are used to track Resources.*

3.4 Station 2: The Global Wait Queue

Recall that upon the instantiation of a Task, each Activity is placed inside the Global Wait Queue to await processing. Task timing and routing attributes are used to control the release of Activities from the queue so that logical dependence between Activities is correctly maintained. In this regard, two timing attributes play a major role in the release of each Activity: SNET and SNLT. Activities are held in the queue until their SNET date. During this holding period Activities remain poised to accept Resources from other prerequisite Activities. Upon the SNET date, MARS algorithms are designed to determine if, for each participation type, a sufficient number of Resources have been accepted via the Feeder nodes. If sufficient Resources have been fed, the Activity is released from the queue. If sufficient Resources have not been received, then the Activity continues to be held inside the queue. As soon as there are enough Resources to satisfy minimal Resource requirements, the Activity is released from the queue. Activities that remain in the queue after their SNLT date become defunct; their Resources are released and the Activity itself is deleted from the simulation[45].

3.5 Station 3: Resource Acquisition

After their release from the global wait queue, Activities flow directly to station 3 where the algorithms within the MARS kernel use Finder Node attributes to direct the acquisition of Resources. Each finder node within an Activity is processed one at a time in ascending order of internal Finder ID number. Individuals to fill Resource requirements on each Finder Node are acquired before the next finder node is assessed.

3.5.1 Resource Requirements

As indicated in section 2, the acquisition of a Resource through a finder node is dependent on two basic pieces of information: a definition of "what" is required (*i.e.* the demand), and a rule for determining "where" to select candidates from the establishment, equipment fleet, or facilities pool. The data structure used to store information about "what" is required is depicted in Figures 42, 44 and 45. Figure 42 contains information about required status levels. In general, a Resource may require individuals with several status types where, for each status type, the status level of a potential candidate must fall between an upper and lower limit. Fields holding limit requirements are indicated in the Doing-Status Requirement Definition table and the B&E Status Requirement Definition table. Also indicated in Figure 42 are the Doing-Status Collection Definition table and the B&E Status Collection Definition table. These tables, when linked to the Finder Definition table, indicate how the statuses of individual personnel, equipment, or facilities within a Resource are to be altered upon the commencement or termination of an Activity. The complete definitions of Doing

[45] With this in mind, an issue of relevance to the prescription of more advanced timing mechanisms is the prescription of the minimal Resource requirements for an Activity. If proper care is not taken, the default or cancelation of a single Activity within a Task may have an unwanted "domino effect" and thereby preempt the completion of an entire Task.

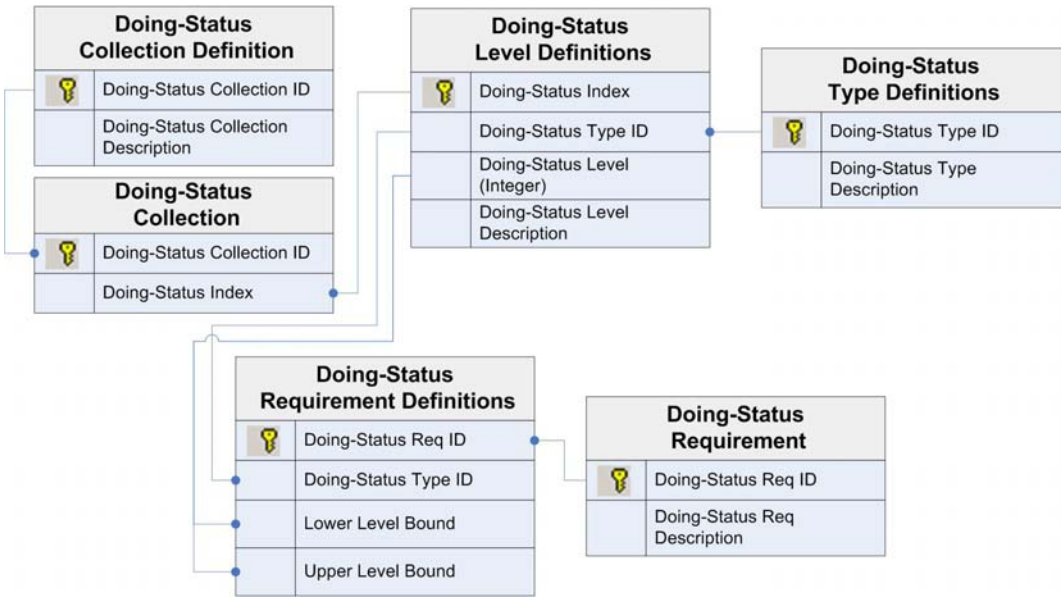


Figure 44 *: Relational data structure used to store status & status requirements definitions. Part 1: Doing Status.*

and B&E status collections are illustrated in Figures 44 and 45. More than one status type can be assigned to individuals as they enter a Finder Node[46].

Doing-Statuses and B&E statuses are not the only criteria used for the selection of individuals within a Resource requirement. Specifically, each individual in the collection that makes up a Resource may be required to fulfill a different combination of Rank, MOSID and Qualification requirements[47]. These individual requirements are called position requirements. As described in section 2.5.2, each finder node is assigned a particular pointer. This pointer takes on the name of a branch within a predefined ORBAT. As shown in Figure 46, ORBATs are saved in the Theatre Units table and related data about individual positions are stored in the Theatre Position table and Position Requirement Definition tables. To the Position Requirement Definition table is linked a separate set of requirements for Rank, Occupation and Qualifications.

Rank: The first prototype of MARS allows for the specification of rank requirements in terms of a range. Every Rank is prescribed an integer parity rating where low ratings correspond to low ranks and high ratings represent high ranks. A Rank range is defined

[46]Note that in the first MARS prototype, status level assignments in the B&E family are defined as part of the definition of each Feeder and Finder Node. Enhancements in future versions of MARS will likely allow for the assignment of B&E status levels as part of each Next-Activity node. This alteration may serve to enable the consolidation of certain input data.

[47]Note that in the current version of MARS only personnel resources are accommodated. In future versions, the capabilities to handle both equipment and facilities are to be included. In so doing it is expected that individual attributes akin to Rank and MOSID will be included for these latter categories.

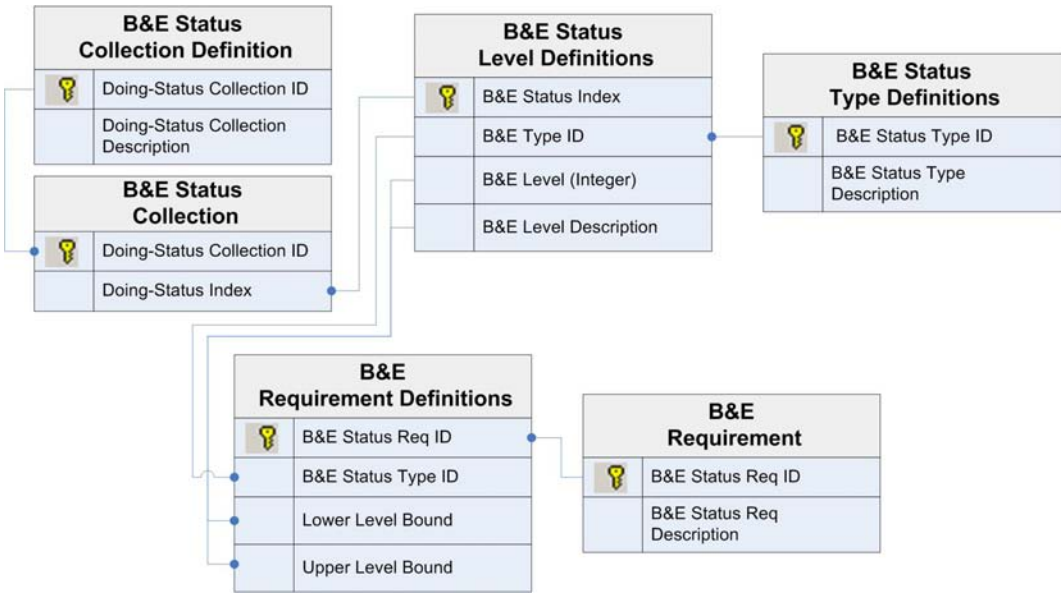


Figure 45 : *Relational data structure used to store status & status requirements definitions. Part 2: B&E Status.*

by specifying upper and lower parity ratings.

MOSID: The acronym MOSID means Military Occupations Specialty Identification. It refers to a new naming convention that is slowly replacing the use of MOC codes in the Canadian Military. A collection of acceptable MOSIDs may be specified for each position requirement. The order of appearance of each MOSID in a collection demarcates the desired selection preference.

Qualification: Collections of qualifications are defined similiar to MOSID collections however, whereas an individual can typically have only one MOSID, he/she may have attained several Qualifications. The potential of any individual to satisfy only a subset of qualification requirements is accounted for by using a Priority attribute. Several qualifications within a single requirement may be assigned a priority rating of 1 to indicate that they are essential while others might be assigned a rating of 2, 3, 4... to indicate they are not as important. Before running a scenario, the user can specify a global variable to indicate the desired level with which to fill qualifications requirements when selecting personnel. A global priority level of 2 for example would signify that qualifications with priority ratings of two or less must be satisfied for an individual to become a candidate for potential selection.

Since each position within a Resource can have a unique combination of Rank, MOSID, and qualification requirements, and several personnel may be candidates for more than one position, the way that personnel are selected to fill particular positions can impact upon the ultimate ability of the establishment to fill a set of Resource demands. An attribute of particular importance in this regard is the Search Rating. Search ratings are used within

MARS algorithms to set the order that individual positions within a Resource are filled. Higher ratings are assigned to positions that are more important and/or harder to fill and lower ratings are assigned to positions that are easier to fill. Currently the individual position requirements within a Resource are filled in descending order of search rating.

As is illustrated in Figure 46, Individual search ratings are stored in the definition tables for Rank, MOSID, and Qualification and collective search ratings are stored in the tables labeled Position Rank Combo Definition, Position MOSID Combo Definition, and Position Qual Combo Definition. As is described below, the way that individual search ratings combine to form a collective search rating is different for each of the Rank, MOSID and Qualification categories. With respect to this it is noted that future versions of MARS will incorporate more sophisticated methods for setting and/or automatically computing search ratings – procedures that in the first MARS prototype are intentionally simple.

Collective Rank Search Rating: Each Rank Definition is assigned an individual search rating which is stored in the Rank Definition Table. In general, the individual search ratings in this table are intended to be inversely proportional to the number of personnel available at that rank level. With this in mind, the method to compute the collective search rating for any range of ranks is to extract the minimum of the individual search ratings in the range.

Collective MOSID Search Rating: Similiar to the method used for Ranks, an individual search rating is part of the definition for each MOSID definition. The minimum of these is used as the collective Search rating for a set of MOSIDs.

Collective Qualification Search Rating: The search rating for individual qualifications is stored in the Qualification Definition table. The collective search rating for a set of qualifications is obtained by computing the average of the individual qualification search ratings[48].

After the computation of collective search ratings for each of the Rank, MOSID and Qualification categories, a total cumulative search rating is computed for each position requirement. This cumulative search rating is computed internally within MARS kernel using a weighted sum and then used to sort position requirements. As noted above, the sorted list sets the order with which position requirements are filled.

3.5.2 The Resource Selection Mechanisms

Aside from the specification of positional requirements, the determination of where to find potential candidates is the most important part of a Resource specification. As discussed in section 2.5.3, the design of the first MARS prototype incorporates two mechanisms to find individual personnel: the Roto Mechanism and the Stealing Mechanism.

[48]Parity ratings associated with each Qualification in a collection are not included in the computation of cumulative search ratings in the first MARS prototype.

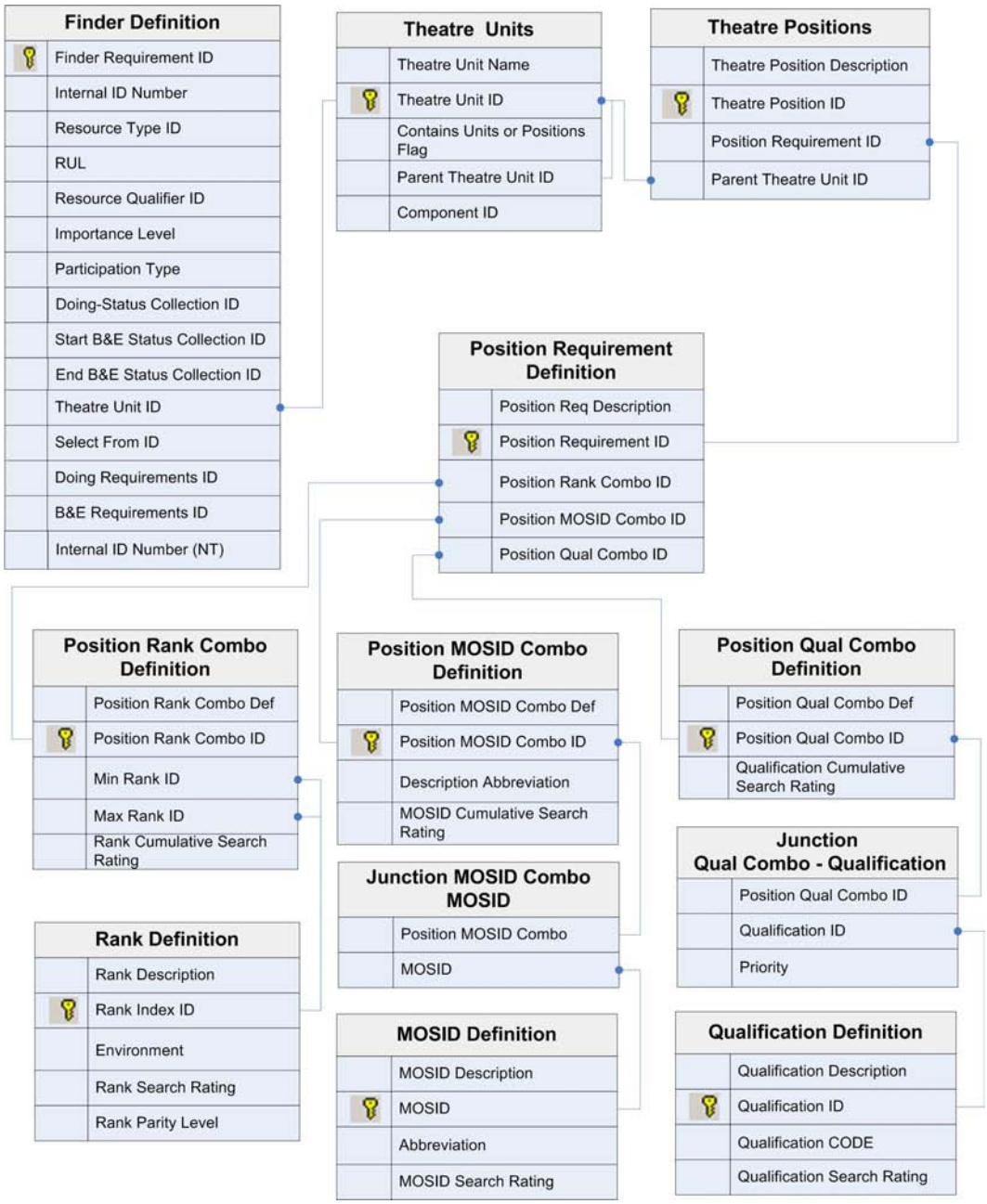


Figure 46 : *Relational data structure used to store information about particular position requirements for a Resource. Each Finder is linked to a theatre unit which is in-turn linked to a group of individual positions. Each position can contain Rank, Occupation, and Qualification requirements.*

3.5.3 The Roto Mechanism

The data layout for the Roto Mechanism is depicted in Figure 47. This figure is best interpreted in conjunction with Figure 32. Each record in the Finder Definition table represents a single Finder Node. Each Finder Node can make use of both the Roto Mechanism and the Stealing Mechanism as per a user specified sequence stored in the Select From Table. For example, the Select From Table may indicate that individuals to fill a specific Resource requirement should first be selected from a particular place in the establishment, and then only in the case where certain positions remain unfilled should individuals be solicited from those already assigned to other Tasks, Activities, or Resources.

Referring once again to Figure 47, Select From Table is connected to Hybrid Table. The term "hybrid" is used because each record is a hybrid combination of fields that correspond to either the Roto Mechanism or the Stealing Mechanism depending on the `Use Roto or Stealing Mechanism Flag`. If the flag indicates that the Roto Mechanism is to be used, the field data corresponding to the `Area Roto ID`, the `Area Roto Pointer`, the `Unit Collection ID` and the `SubUnit Collection ID` are employed by the selection algorithms. If the Flag is set to indicate use of the Stealing Mechanism, then the `Stealing Junction ID` is employed.

For the Roto Mechanism, the linkages between the Hybrid Table and tables pertaining to Area, Unit and SubUnit Rotos are clearly evident. The layout of these tables follows directly from the description of the Roto Mechanism in section 2.5.4. The Area Roto Definition Table and the Area Roto Table encode the Area Rotos and the Area selection rules are encoded in the Area with Backups Rule Definition and the Area with Backups Rule tables. The Unit and SubUnit Rotos with their respective selection rules are stored in a similiar fashion – two tables are used to define the Rotos and two tables are used to define the selection rules. The only major difference between the Area and the Unit and SubUnit constructs is that Unit Rotos and SubUnit Rotos can belong to "collections" while Area Rotos can not. Collection information for Unit Rotos is contained in the Unit Roto Collection Definition table and the Unit Roto Collection tables. Collection information for SubUnit Rotos is contained in the SubUnit Roto Collection Definition Table, and the SubUnit Roto Collection tables.

Each finder node that makes use of the Roto Mechanism uses all of the above mentioned tables to create a set of pointers into the establishment. As discussed in section 2.5.4, some pointers in the set correspond to primary locations for the selection of individuals and other pointers correspond to alternate (*i.e.* backup) locations. For each location MARS algorithms are designed to traverse through the establishment tree to obtain the collection of personnel that fill the incumbent positions. These personnel are then assessed as candidates to fill Resource requirements. The data structure that encodes the link between establishment units, establishment positions, and personnel is depicted in Figure 48. In this figure there are two items of particular note. The first item is that the Position Requirement Definition table is the same table as in Figure 46 which links to the subset of Rank, MOSID and Qualification requirements. The second item is that the personnel are linked to establishment positions through an intermediate table called the Establishment Position Filled

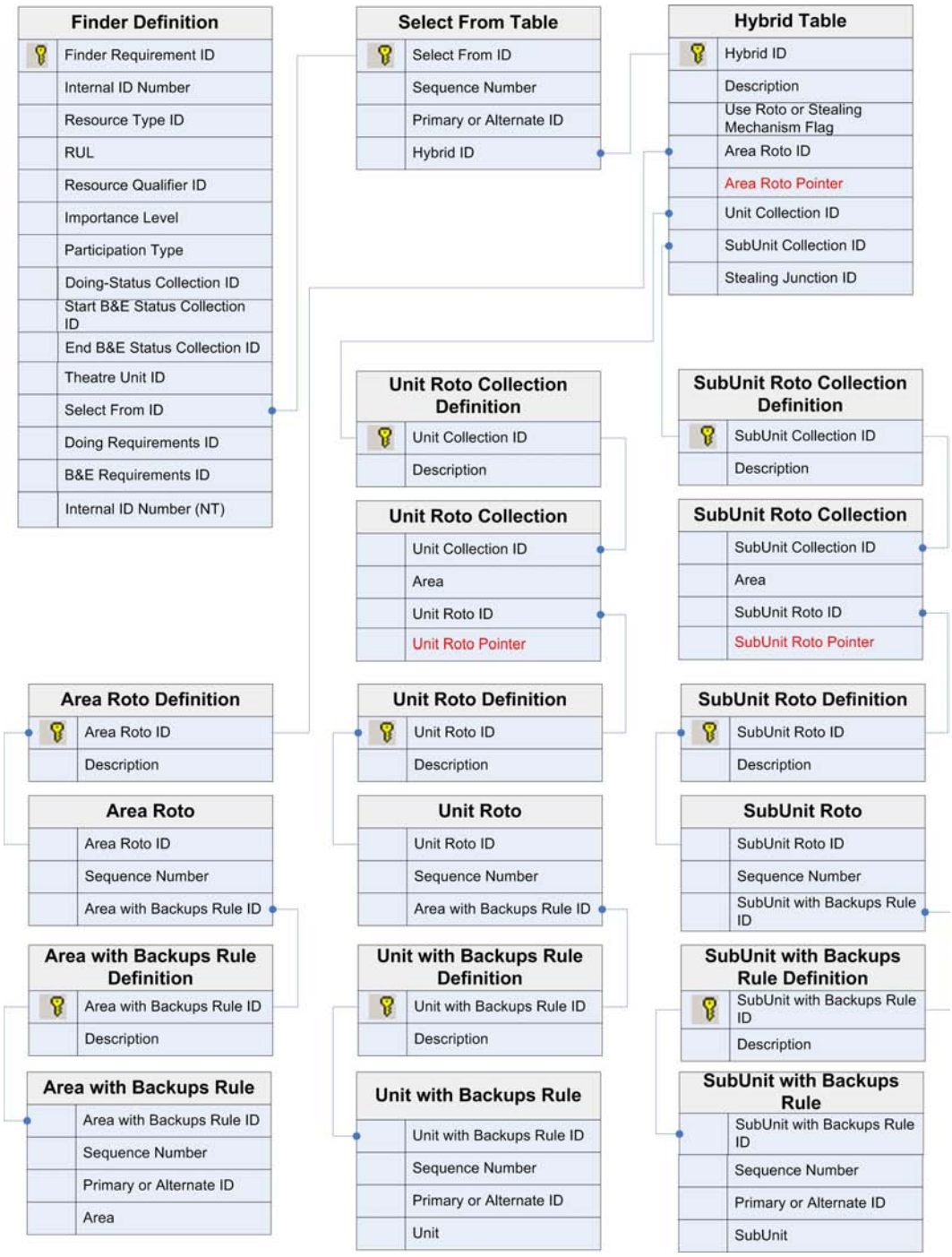


Figure 47 : *Relational data structure used to store Roto Mechanism Information. This figure is best interpreted in conjunction with Figure 32. Items in Red are initialization values that are updated over the course of a simulation.*

Record table. With regard to the latter, the establishment is considered to be "static" in the first MARS prototype. The term static refers to the fact that personnel do not traverse through a cycle of recruitment, promotion, transfer, posting, and attrition within a simulation. Instead, a person assigned to a particular position remains in that position through the entire simulation. Methods to model the movement of personnel within the establishment are to be incorporated into future versions of MARS. The Establishment Positions Filled Initial table as well as the collection of other personnel related tables in Figure 49 set the stage for this kind of enhancement.

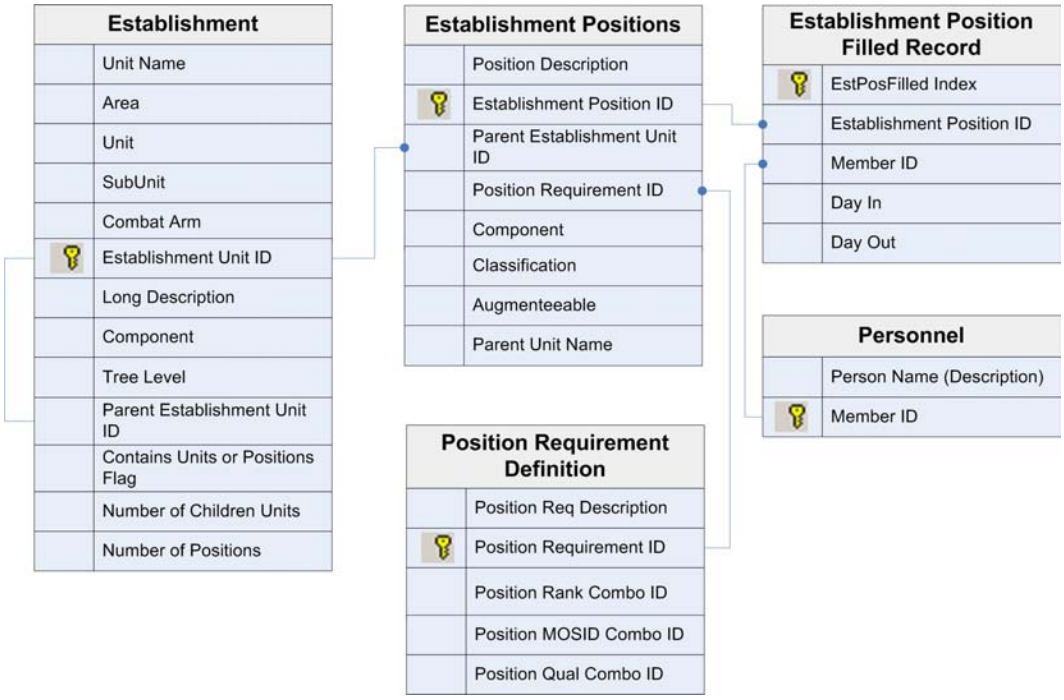


Figure 48 : *The Relational data structure used to store the establishment and personnel who fill positions in the establishment. Note that in the first MARS prototype there is no mechanism by which to directly model human resource related processes such as attrition, retirement, recruitment, and postings. Since personnel are not considered to "move" from position to position over the course of a simulation, the establishment is said to remain "static". It is expected that this limitation will be lifted in future versions of MARS.*

3.5.4 The Stealing Mechanism

The relational data layout for the Stealing Mechanism follows directly from discussions in section 2.5.5 and is depicted in Figure 50. This figure is best interpreted in conjunction with discussions in section 2.5.5. To enable the user to encode a sequence of Task, Activity, and/or Resource locations from which to draw upon individuals to fill a Resource requirement, five tables are used.

The Stealing Junction Definition table and the Stealing Junction table encode a sequence

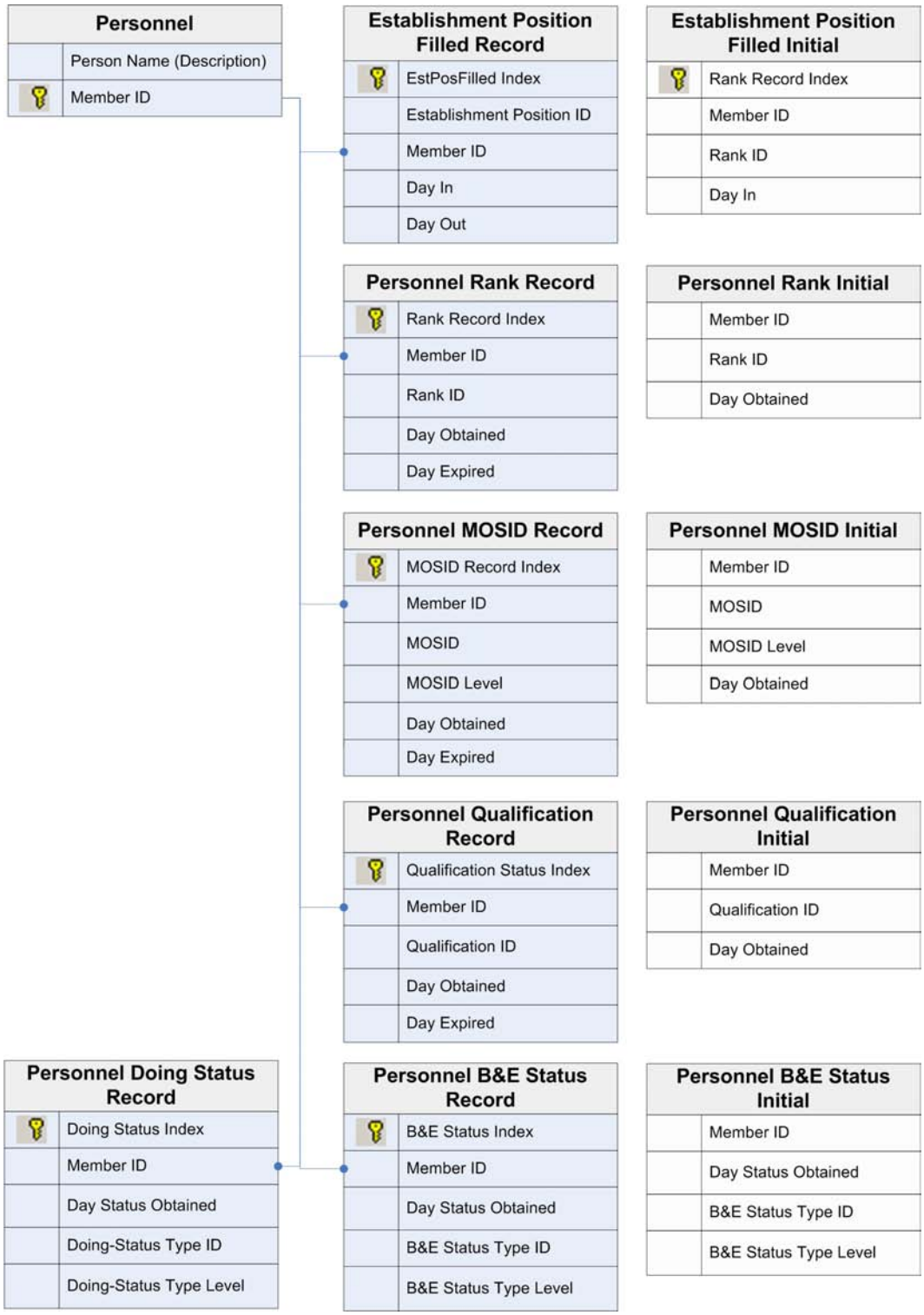


Figure 49 : *The Relational data structure used to store Information about Personnel. At the beginning of a simulation the initial status of individual personnel are imported from the unlinked tables into the tables labeled "Record".*

of rules to select primary and secondary locations. Each location rule may contain specific information about the attributes of Tasks, Activities and Resources that qualify for the potential reassignment of attached individuals. As the names imply, the Task Selection table stores information about Task attributes, the Activity Selection table stores information about Activity attributes, and the Resource Selection table stores information about Resource Attributes.

In the first prototype of MARS the design of the Stealing Mechanism allows for only one combination of Task, Activity and Resource Selection attributes per record in the Stealing Junction table. The implicit assumption is that when these are prescribed together in the same record, all associated attributes must be concurrently satisfied in order for the particular Task, Activity or Resource to be selected. For example, if each of the `Task Selection ID`, `Activity Selection ID`, and the `Resource Selection ID` contain meaningful links from within the same record of the Stealing Junction table, then for an individual to be available for reassignment to the current Resource, it must be affiliated with a Resource, Activity, and a Task with matching attributes. This functionality is representative of a "logical AND" where multiple conditions must be satisfied simultaneously for a certain action to take place. The functionality of a "logical OR" on the other hand is enabled by creating multiple records in the Stealing Junction table where each may have the same `Sequence Number` and/or `Primary or Alternate ID`[49]

3.5.5 Assigning Individuals to Positions

In the first MARS prototype, both the Roto Mechanism and the Stealing Mechanism serve to produce two lists of candidate individuals for potential selection to fill positions within a Resource requirement. The first list corresponds to those individuals in primary locations and the second list corresponds to those individuals in alternate locations. The primary list is compiled from the list of primary locations and the alternate list is compiled from the list of alternate locations as per the selection Mechanism. An example list of primary and alternate locations obtained using the Roto Mechanism is illustrated at the bottom of Figure 32.

Each of the primary and alternate lists of personnel is sorted based on an aggregate "search" rating that is intended to measure the uniqueness of each individual from all others. Those with higher Ranks, special MOSIDs and rare qualifications are assigned higher search ratings. The algorithmic specifics about how each list is sorted are as follows.

Primary List All personnel in the primary list are sorted in ascending order of search rating irrespective of the location from which they were selected.

[49] In future versions of MARS, multiple Task Selection IDs, Activity Selection IDs, and Resource Selection IDs may be linked to a single Stealing Junction ID through the construction of intermediate junction tables. Notwithstanding the increased complexity that these tables would introduce, they would serve to expand the "logical AND" functionality within the Stealing Mechanism and ultimately serve to make it more flexible and robust.

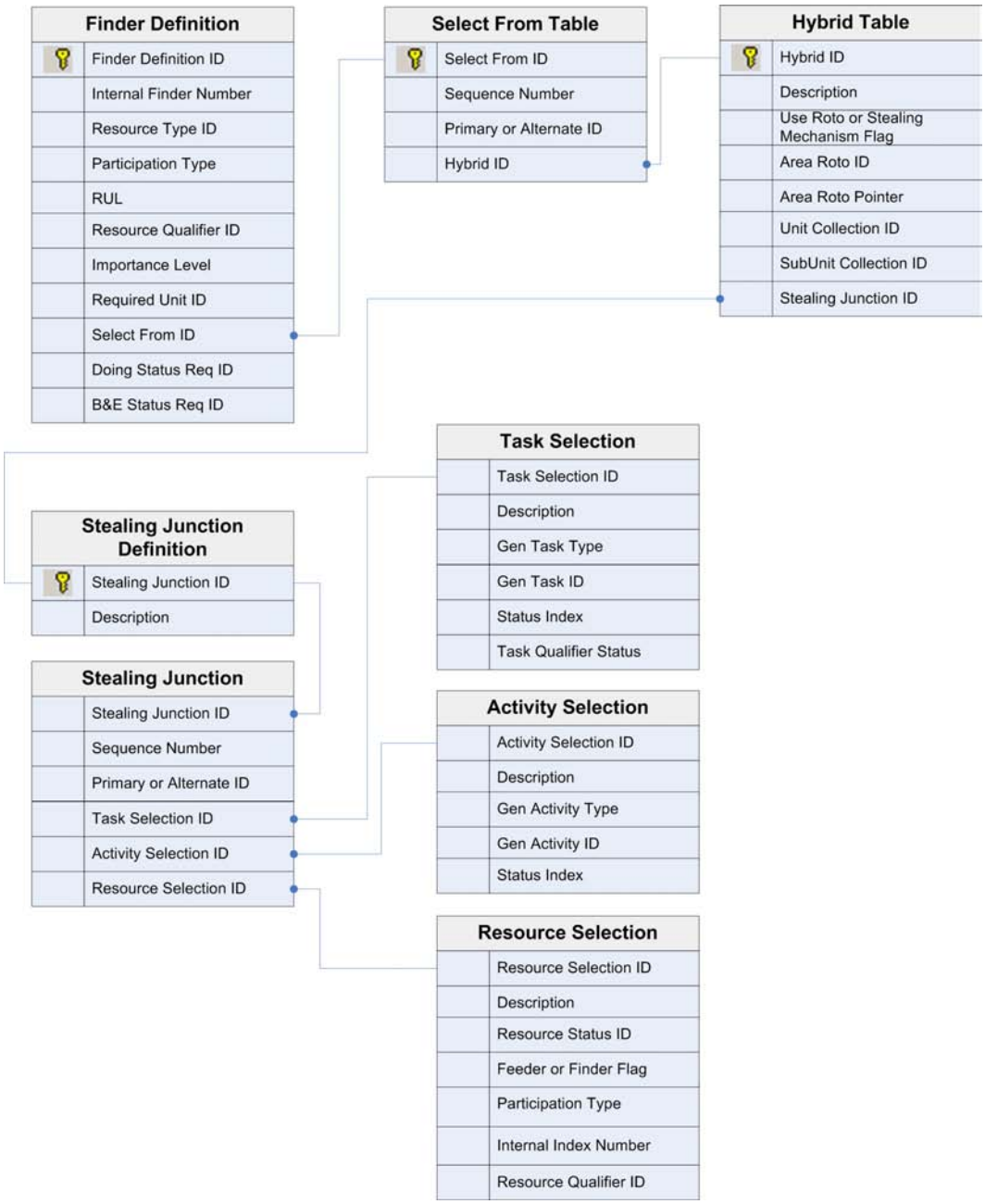


Figure 50 : *The relational data structure used to store defining information for the Stealing Mechanism. This figure is best interpreted in conjunction with discussions in sections 2.5.5 and 3.5.4.*

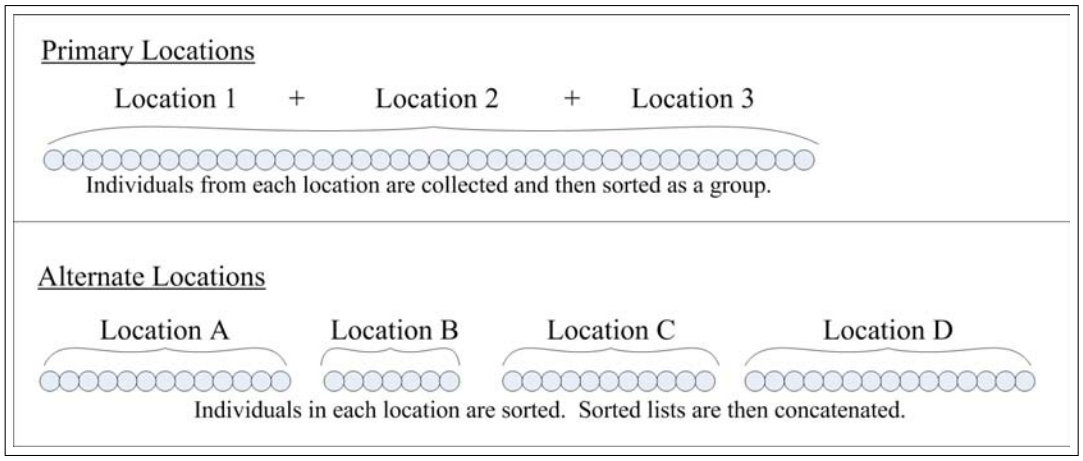


Figure 51 : *In order to draw individuals from the establishment to fill positions in a Resource, the Roto Mechanism produces a series of primary and alternate locations upon which to draw individuals. As described in section 3.5.5, the individuals in locations demarcated as primary are first collected and then sorted. Individuals in each alternate location are sorted before they are concatenated. The flexibility to use alternate sorting and concatenation strategies is envisioned for future versions of MARS.*

Alternate List The list of alternate personnel are segmented according to their residing location. With the Roto Mechanism for example, locations are represented by particular branches of the establishment. Branches are ordered according to the sequence of alternate locations produced by the Roto. Within each branch the list of personnel is sorted in ascending order of search rating. This concept is illustrated in Figure 51.

The main purpose for sorting candidate individuals according to search rating is to reduce the probability that highly valuable and/or uniquely qualified individuals are selected to fill a position when a myriad of other candidates with lesser or more common qualifications would suffice.

Mindful of the fact that position requirements within each Resource are sorted in descending order of perceived importance and/or estimated difficulty to fill, MARS algorithms are designed to fill each position in the sequence by first going through the ordered list of primary individuals. The first instance where an individual fits the minimum requirements of the position, he/she is selected and assigned to that position. One by one, each position is compared against the candidate personnel in the primary list until all positions have been assessed.

In a second step, those positions that remain unfilled are then compared against the list of personnel in the secondary list. Positions that are unfilled[50] after searching through the secondary list remain that way throughout the remainder of a simulation. This functionality

[50]Counting the number of unfilled positions in a set of resources in an Activity, Element or Task is one of the ways that the viability of a scenario is measured with MARS.

is expected to approximately mimic the majority of practical cases. However, it may be revisited in future versions of MARS to account for cases where individuals are expected to join an Activity that is already in progress.

Rank, MOSID, and Qualification attributes are of prime importance when assessing candidate individuals to fill the requirements of a position in a Resource. However, of equal importance are other attributes including Doing-Status, B&E status, Importance level, and Utilization Level. With regard to these latter attributes, the algorithms within MARS are designed to account for the fact that some candidates may already be assigned to Activities that may not have begun. Recall that as Activities are completed some resources may be passed to other Activities residing in the Global Wait Queue. When this occurs, those resources are essentially "pre-booked" to participate in that Activity. Within the MARS paradigm the term "assignment" is used to imply that pre-booking has occurred.

In stage 3 of the Activity lifecycle, individuals are selected and assigned at the same time. During the assessment of an individual to fill a position requirement, MARS algorithms project forward to determine whether there are any conflicts between the current Activity for which the individual is being assessed and Activities to which the individual is already assigned that may occur in the future. Part of this projection is an evaluation of how statuses and utilization levels are expected to change when those Activities to which candidates are already assigned actually occur. For example, MARS will not allow the utilization level of any individual to exceed 100%. If, in the process of assessing a candidate for a Resource with an RUL of 1, it is found that the individual is already assigned to an overlapping Activity with the same importance level and an RUL of 0.5, then that individual will be deemed unavailable and will not be selected. The same is true for status levels. If an overlapping Activity to which an individual is already assigned will cause a change in status level beyond the allowable bounds of the position for which the individual is currently being assessed, the assignment will not occur.

The algorithms that control the selection of candidates are designed to remove personnel from resources attached to other Activities when the current Activity has a higher importance level and the addition of the new Activity would result in a utilization level greater than 1. In these cases individuals are removed from all concurrent Activities with lower importance levels that have individual utilization levels greater than zero[51]. The term "removal" implies that the position becomes empty in the Resource from which the individual is being removed. As noted earlier, the algorithms within MARS make no further attempt to fill empty positions. To track the addition and removal of individuals to resources throughout the duration of a simulation, the kernel algorithms within MARS are designed to record this information into one or more of the tables shown in Figure 52. All of these tables are of particular importance to the MARS post processing routines (not described in this report) which are designed to consolidate and analyse scenario results.

Similiar to Figure 43, Figure 52 includes the tables Specific Tasks, Specific Elements and Specific Activities that record the temporal information about the changing conditions of

[51] Resources with utilization levels of zero are unaffected by the inadvertent removal of individuals.

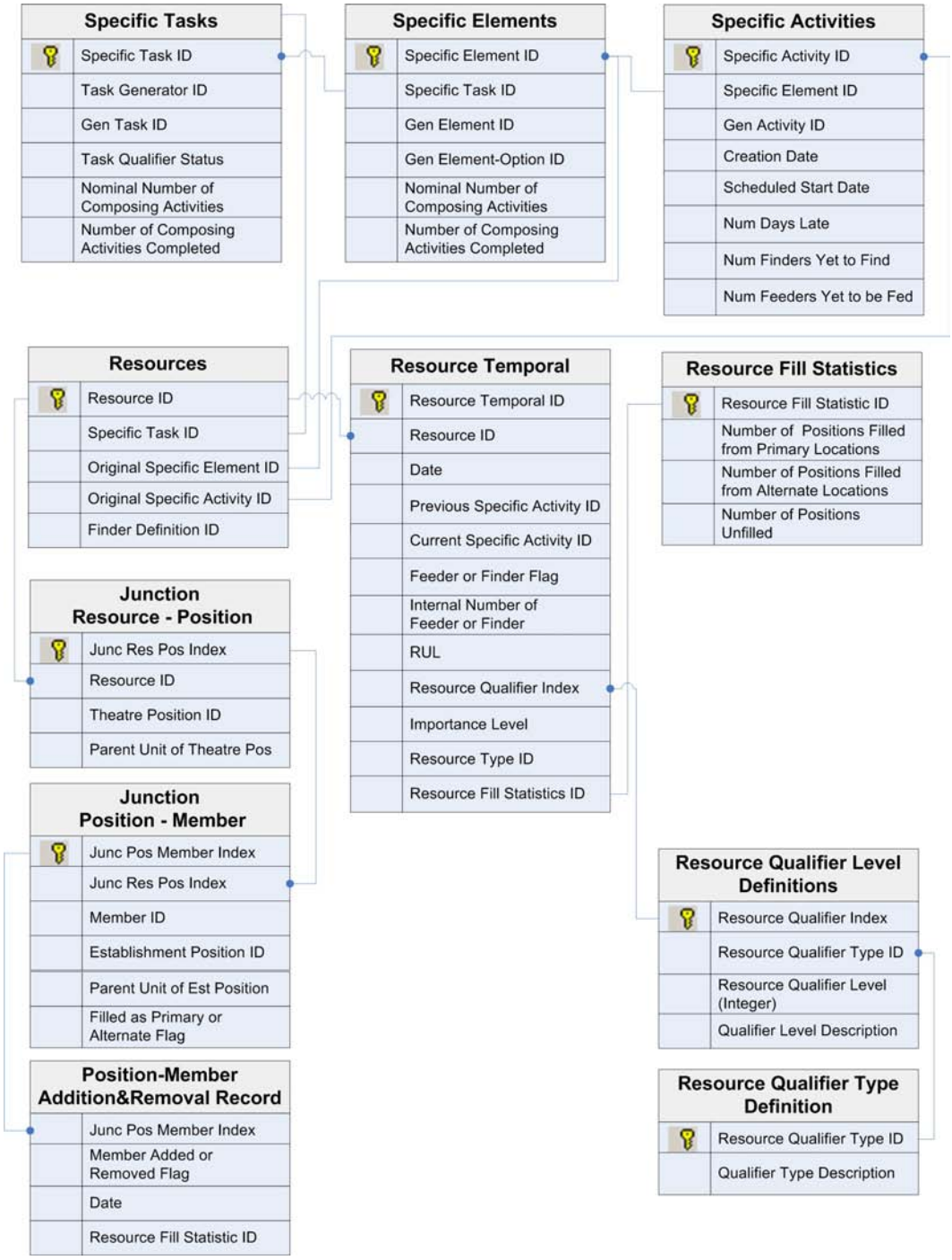


Figure 52 : *The Relational data structure used to track a Resource over the course of a simulation.*

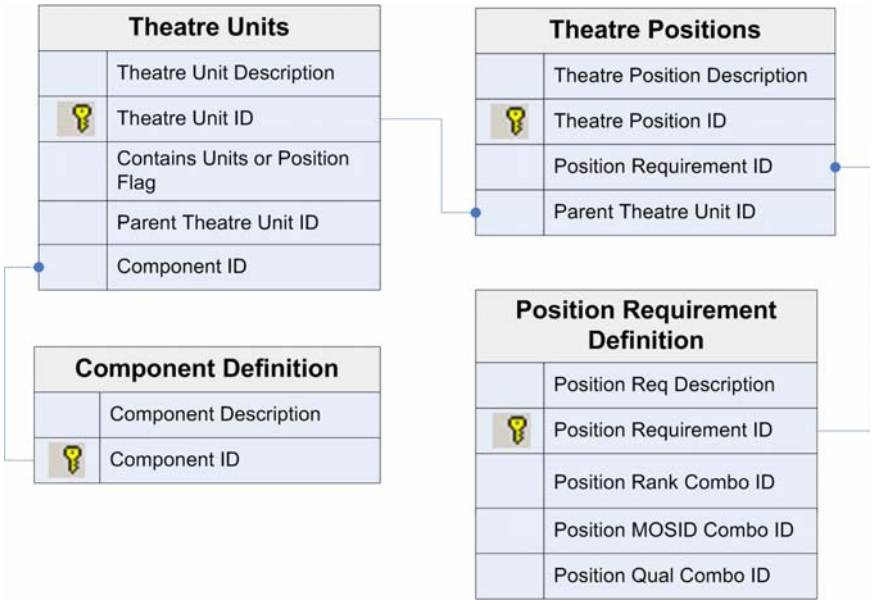


Figure 53 : *The relational data structure used to store ORBAT information.*

Tasks, Elements, and Activities as a simulation progresses. These tables are linked to a table called Resources that is in-turn linked to several other tables that store dynamic information about the condition of each Resource over the course of a simulation. Specifically, the Resource Temporal table stores information about Resources as they flow from one Activity to the next. Fields within this table include the `Previous Specific Activity ID`, the `Current Specific Activity ID`, the `RUL`, and the `Importance Level`.

Attached to the Resource Temporal table is the Resource Fill Statistics table. This table holds information about the number of positions within the Resource that are filled at any particular time. As individuals are assigned to fill positions within a Resource, new records are added to the Resource Fill Statistics table. Other tables linked to the Resources table in Figure 52 are the Junction Resource-Position table, the Junction Position Member table, and the Position-Member Addition&Removal Record table. The Junction Resource-Position table links each Resource to individual positions. The `Theatre Position ID` field in this table is directly linked to the Theatre Positions table depicted in Figure 53 which inturn links to the multitude of requirements that define the position.

Returning to Figure 52, the Junction Resource-Position table is linked to the Junction Position-Member table. This latter table records information about the individuals assigned to each position in a Resource. The `Member ID` field links to the Personnel table in Figure 49 and the `Establishment Position ID` and the `Parent Unit of Est Position` fields hold information about the establishment position held by the individual when they were assigned to the Resource. Information that indicates when each individual is assigned and removed from a particular position within a Resource is stored in the Position-Member Addition&Removal Record table.

One of the last procedures carried out by the MARS algorithms as part of station 3 is the creation of a record(s) in the Specific Activities table (see Figure 43). As Resources are fed to an Activity in stage 2 and/or found for an Activity in stage 3, internal counters within the MARS kernel keep track of the number of Resources that remain to be fed and/or found before the Activity can be effectively released from stage 3 for processing at stage 4. When Activities are finally released, these counters, as well as other information about the Activity (such as the `Instantiation Date`), are written to a single record in the Specific Activities table.

In practice, the allocation of individuals to Activities which is modelled at stage 3 is a complex and non-deterministic process. Besides the fact that it is often difficult to extract the "procedural rules" that govern the practical allocation of individuals from the Canadian Army, these rules may be different across the establishment and can be dependent on the situation. With this in mind, the processes within MARS that govern the selection and assignment of resources have been designed to be very simple and without some of the flexibility and foresight that may exist in practice. By careful specification of Activity timing attributes and importance levels, it is expected that the overall effect of inherent simplifications will not hinder the production of results that are relatively realistic. As the MARS algorithms are developed, simple improvements may enable the pooling of positional requirements before the filling process begins. This would preclude the filling of resources on an individual basis. The pooling of Resource requirements is likely to be dependent on the participation type and/or the primary and secondary locations from which individuals are to be selected. Other more substantial enhancements may include the ability to pre-book (*i.e.* assign) individuals for participation in a string of Activities all in one step. This feature would be quite different from the method implemented in the first MARS prototype where pre-booking for an Activity is only done when a prerequisite Activity is completed.

There is much potential for new algorithms to be implemented that would effectively allow individuals to be pre-booked many months (potentially years) into the future. In broad terms, this functionality would enable the inclusion of algorithms for the automatic optimization of resource allocation. However, before these kind of enhancements can be implemented, many issues regarding the inclusion of optimization algorithms will require careful consideration. Of these, the transparency and practicality of "optimal" solutions, as well as the ability of algorithms to continuously alter pre-booking schedules based on unexpected random events, will be of primary importance.

3.6 Station 4: Activity Processing

After the assignment of resources at station 3, Activities flow to the Activity Processing station. The major tenet of this stage is a large queue called the Processing Queue which holds Activities for their duration. As Activities enter the processing queue a series of statuses are assigned to the Activity, the attached resources, and the individuals participating in each Resource. Among these are the Doing and B&E Statuses described in section 2.4.7. As Doing and B&E statuses change, new records are written to the `Personnel Doing Status Record` table and the `Personnel B&E Status Record` table illustrated in Figure

49. Recall that Doing-statuses are set as Activities enter the processing queue and are automatically reset to a null state as Activities are released from the queue. Similarly, B&E statuses are set as Activities enter the processing queue; however, unlike Doing statuses they are not automatically reset as Activities are released from the queue. B&E statuses are not changed unless specifically prescribed as part of the Activity definition.

3.7 Station 5: Resource Routing & Activity Dissolution

The final stage in the Activity lifecycle involves that of passing Resources to subsequent Activities, dissolving others Resources, and removing Activities from the simulation. As discussed in section 2, the flow of Resources between Activities is directly dependent on external routing attributes contained in both the Feeder Nodes and Next-Activity Nodes of an Activity. As Resources are fed between Activities, new records are added to the Resource Temporal table. As such, many records in the Resource Temporal table can link to a single record in the Resources table or the Resource Fill Statistics table.

Resources that participate in an Activity and are not passed on to a subsequent Task are effectively dissolved. When a Resource is dissolved, participating individuals are released (*i.e.* removed) and new records are written to the Position-Member Addition&Removal table.

The final action that occurs at station 5 is the effective removal of the Activity from the simulation. With regard to the relational data structure, the dissolution of an Activity is demarcated in the `Specific Activity Progress` table in Figure 43. Other tables in this figure are also updated. Specifically, it is noted that internal counters within the MARS algorithm serve to store the number of Activities completed for each Task and Element. When the completion of a particular Activity marks the completion of all Activities in a Task or Element, new records are written to the Specific Task and Specific Element tables. The only other notable instance when records are written to the Specific Task and Specific Activity tables is upon termination of a simulation.

3.8 Closure

The functionality embedded in the second level of abstraction provides the machinery by which MARS is able to process Tasks and their incumbent Activities. The computer algorithms that encode the functional operation of each station are modular by design. As MARS is further developed these modules may be replaced, expanded or rearranged in order to improve simulation fidelity.

Each subsection herein is dedicated to a particular stage in the lifecycle of an Activity. Also described is the relational structure used to store data that are either required as input to define a scenario or produced as output from a simulation. With regard to the latter, it is important to note that although Figures 36 through 53 contain the majority of important tables in the relational framework, some of the links and a few of the tables in the MARS database have not been illustrated because they are affiliated with aspects of MARS that lie

outside the scope of this report[52]. Moreover, some of the table names and field names have been altered from how they actually appear in the MARS database in order to maintain a degree of consistency with terms used in this report.

[52]These include: the graphical interface to solicit scenario input, the analysis interface used to interrogate simulation results, and/or programmatical aspects of the implementation in ARENA.

4 Future Development

The first MARS prototype is now complete. The focus of this first phase has been the development of a "proof of concept". In its present state, MARS is intended for "off-line" analysis with direct input from military planning staff. In its present state MARS is poised to integrate data that is not currently integrated with other analysis tools. As such, it is expected to have appeal to a variety of potential stakeholders across the Canadian Forces.

With this in mind it is noted that MARS is a prototype product. There are a variety of avenues for potential development. From the viewpoint of MARS architecture, enhancements can be made to each of the four component parts: the graphical user interface, the relational data storage structure, the simulation kernel, and the analysis utility. Several items for enhancement related to the simulation kernel have been noted in the context of discussions in this report. These will not be repeated here; rather the purpose of this short section is to briefly describe some of the potential options for future developmental research. These represent additions to the fundamental capabilities of MARS rather than mere extensions of the current framework. In concert with military partners, it is expected that a concrete strategy to move forward on these and other issues will be solidified after the capability of MARS to address issues with the current MRS and MRP have been assessed.

The following is a list of potential areas for future development of the MARS kernel. It should be noted that this list is not exhaustive and is presented in no particular order.

Tracking Human Resources: Although the current MARS constructs form a baseline framework, they do not easily facilitate the modeling of individuals through the human resources cycle from recruitment, through promotion, to attrition or retirement. Extending MARS to incorporate these aspects will enable analysts to assess the impact of the ageing demographic in the Canadian Army on its ability to carry out future tasks.

Costing: MARS is designed to track and forecast the assignment and utilization of resources over a wide range of Activities. If simple models can be developed to associate real and/or normalized costs to these Activities, then it may be possible to carry out a very basic cost/benefit analysis. These kinds of analyses could add important perspective to the efficacy of different concepts tested within MARS.

Prescriptive Scheduling: Currently MARS utilizes a "first-come first-serve" methodology for assigning resources to specific Activities and Tasks. With this mechanism the user can explicitly control how the establishment is scheduled to fill demands. This ability is important because it provides a necessary degree of transparency to the program; however, as MARS is developed further it is expected that the addition of utilities to "automatically" produce a collection of viable scheduling options will provide great benefit.

In-Situ Analysis: In its present state, MARS is intended for "off-line" analysis. In the future it may be possible to make the algorithms more amenable for "on-line" analysis

at planning conferences. The aim of the latter would be to provide immediate feedback regarding pressing issues with the collective assistance and collaboration of Army representatives from the Land Force Area Commands, the Land Force Doctrine and Training System, and the Individual Training System.

Resources: At present, although the MARS framework has been developed to handle a variety of different resource types, only personnel are currently incorporated. The added ability to model equipment and facilities is expected to provide more holistic results.

Capability Matching: At present, MARS requires a user to explicitly prescribe the establishment units that are to fill certain roles in a particular Activity. As well as requiring that the user develop a schedule, this feature requires the user to match the requirements of a particular Activity to the capabilities of certain units within the establishment. Potential benefits may be realized if a database that matches each part of the establishment with a variety of inherent capabilities can be incorporated into MARS.

Generic Constructs: The generic constructs upon which MARS is based include: Resources, Activities, Elements, Element-Options, and Task-Templates. One aspect of the MARS design was to allow for the reuse of individual constructs. For example, a generic Activity could be designed by the user to be used in the construction of several Task-Templates. However, in its current state there are limitations to Activity reuse that are inherent in how these constructs are stored inside the relational database. To remove these restrictions potential benefits may be derived by incorporating another construct into the MARS paradigm that would effectively change the way that Activities are linked together.

Quality: MARS is designed to evaluate the ability of resource pools to fill demands. However, in as much as a demand can be defined as a range of minimal and maximal requirements, there is the ability to associate a quality factor to the way in which demands are met. Incorporating these kind of factors into MARS is expected to aid in the evaluation of "how well" a particular Task or capability might be performed. Analysis like these may be helpful when attempting to determine whether to perform many tasks at minimal quality levels or only a few tasks at high quality levels.

Data Collection: Although not a development issue, the collection of data to feed into MARS will remain of continual importance if MARS is to be effectively utilized in the years to come. In this regard it should be clear from discussions in this report that a wide variety of information about resource supplies, resource demands, and processes/policies needs to be incorporated. At the time of writing, the first MARS prototype incorporates only a limited amount of information about the Field Force, its expected tasks, and related processes and policies. These will need to be expanded and continually updated in order to solve real world problems with MARS.

References

1. Stoddard, S., Shearer, R., and Klimas, J. (2006), Army Force Generation: The Marathon Simulation Model, Force Generation Workshop, DND, Ottawa Canada.

2. Richmond, M.K., Nicholson, R., Filinkov, A., Williams, P., and Francis, B. (2002), Modelling Personnel Sustainability of a Rotatable Land Force, Force Generation Workshop, DND Ottawa Canada.

3. Richmond, M.K., Filinkov, A., and Nicholson, R. (2005), The Army Sustainment Model - Prototype (ASM-P): Development & Documentation, Defence Science and Technology Organization (DSTO) Australia.

4. Ormrod, M., Young, C.N., and Pall, R. (2008), A Concept for Modelling Force Generation with the Canadian Army, Managed Readiness Simulator (MARS) v1.0, DRDC CORA, TM xxxx.

5. Pall, R., Young, C.N., and Ormrod, M. (2007), Modelling Force Generation using the Managed Readiness Simulator (MARS) - Implementation of MARS v 1.0 in a Discrete Event Simulation Environment, DRDC CORA, TM 2007-52.

6. of Land Force Readiness (DLFR), Director (2004), Army Regeneration Plan, Annex A - Managed Readiness, Released by Assistant CLS, 3350-1.

7. Isbrandt, S. and Zegers, A. (2006), The Arena Career Modelling Environment Individual Training and Education (ACME IT&E) Projection Tool, DRDC CORA, TR 2006-03.

8. Law, A.M. and Kelton, W.D. (2000), Simulation Modeling and Analysis, 3rd Edition, McGraw-Hill.

9. (1996), A Guide to the Project Management Body of Knowledge (PMBOK), Project Management Institute (PMI).

10. Ormrod, M. and Young, C.N. (2007), Preliminary Analysis of Task Force Afghanistan Sustainability using the Managed Readiness Simulator (MARS) v1.0 - Phase 1: Infantry Battle Group, Provincial Reconstruction Team, Observer Mentoring Liaison Team, (TM 2007-40).

11. Brennan, Jas. (1968), Applications of Critical Path Techniques, Brussels: The English Universities Press Ltd.

Annex A
The Whole Fleet Management System (WFM)

The third tenet of Managed Readiness is the creation of a new Whole Fleet Management architecture (WFM) that effectively removes "ownership" of certain materiel from individual units in the organizational hierarchy and redistributes these into specialized stocks. Each stock has a certain priority and is meant to supply specific types of Activities. Currently there are 4 stocks in the whole fleet architecture. The priority afforded to filling and maintaining each stock is demarcated in the list below [6].

Deployment Stock The Deployment Stock is meant to supply equipment to the theatres of operation.

Collective Training Stocks The collective training stock is separated into two sub-stocks. One sub-stock is meant to supply equipment to support the validation of MSTFs at the CMTC. The other sub-stock consists of all vehicles and equipment required to conduct individual training.

Log Stock The Log stock represents a contingency reserve and is to consist of approximately 10% of the Army total vehicle asset holdings. The aim of this stock is to meet un-forecasted short-term requirements.

Unit Stock Unit stocks represent vehicles and equipment held at Area, Brigade and/or Unit levels for the purpose of conducting collective training up to Level 4. Domestic Operations requirements for vehicles, driver training courses and the Reserve Forces are also to draw from Unit stocks.

The rationale behind the consolidation of materiel into these large pools together with the installment of centralized management and release authorities is rooted in the need to ensure that equipment is available to support high-readiness and deployment Activities when needed. Coupled with the whole fleet management system are methods to synchronize the introduction of new equipment with the MRS and the MRP.

Annex B
List Of Abbreviations

ALAP As Late As Possible
ASAP As Soon As Possible
B&E Beginning & End Status
BTE Brigade Training Exercise
CF Canadian Forces
CMTC Canadian Manoeuver Training Centre
CORA Centre for Operations Research and Analysis
CT Collective Training
DART Disaster Assistance Response Team
DLFR Directorate of Land Force Readiness (Army G3)
DND Department of National Defence
DomOps Domestic Operations
DRDC Defence Research and Development Canada
FD Force Development
FE Force Employment
FG Force Generation
FNET Finish No Earlier Than
FNLT Finish No Later Than
GUI Graphical User Interface
HRMS Human Resources Management System
HQ Headquarters
HR High Readiness
IRU Immediate Response Unit
IT Individual Training
LOB Left Out of Battle
LFORT Land Forces Operations Research Team
LineOp Line of Operation
MAJAID Major Aid Disaster
MARS Managed Readiness Simulator
MSTF Mission Specific Task Force
MFO Must Finish On
MOC Military Occupation
MOSID Military Occupation Specialty Identification
MR Managed Readiness
MRP Managed Readiness Plan
MRS Managed Readiness System
MSO Must Start On

MSTF Mission Specific Task Force
NEO Non-Combat Evacuation Operation
OLEB On-Line Establishment Browser
OPP Operational Planning Process
OPS Operations
ORBAT Order of Battle
Roto Rotation
RUL Resource Utilization Level
SCF Special Contingency Force
SNET Start No Earlier Than
SNLT Start No Later Than
TEE Trained Effective Establishment
TMST Theater Mission Specific Training
WBS Work Breakdown Structure
WFM Whole Fleet Management

This page intentionally left blank.

Distribution list

(Report distributed by CD unless otherwise noted)

Internal distribution

3 Authors (Hard Copies)

1 DG DRDC CORA

1 Chief Scientist DRDC CORA

1 Section Head (Land)

2 LFORT (2 Hard Copies)

1 LCDORT

1 DRDC Valcartier OR Team

1 PORT

1 SJS OR Team

1 CANOSCOM OR Team

3 DRDC CORA Library (2 Hard Copies, CD)

Total internal copies: 16

External distribution

Department of National Defence

1 ADM(S&T) (for distribution)

1 Director S&T Land

1 DRDKIM 3

1 DG DRDC Valcartier

1 CF College Library

1 CFANS Library

1 Fort Frontenac Library

1 COS Land Ops [DGLS] (Hard Copy)

1 COS Land Strat [DGLCD] (Hard Copy)

1 LFDTS

1 DGL Res

1 DLS [DLSP]

1 G1 [DLPM]

1 G3 [DLFR] (Hard Copy)

1 G4 [DLSS]

1 DLCD

1 DLFD

1 DLR

1 DLCI

1 DAT

1 DAD

1 DLSE

2 G35 (Hard Copy, CD)

International recipients

1 Document Exchange Manager
DSTO Research Library
Defence Science & Technology Organisation
PO Box 44
Pyrmont NSW 2009
Australia

1 Dr. Neville J Curtis
Research Leader Land Operations Research
75 Labs
Land Operations Division
PO Box 1500
Edinburgh SA 5111
Australia

1 Michael Gillman (for dist'n and library)
Chief Technologist
Land Battlespace Systems
Dstl Integrated Systems
Room 31, Bldg A3, Fort Halstead
Sevenoaks, Kent, UK, TN14 7BP

1 Dr. Jason Field
Land Battlespace Systems
Dstl Integrated Systems
Fort Halstead
Sevenoaks, Kent, UK, TN147BP

1 Director, US AMSAA
ATTN: AMSRD-AMS-S)
392 Hopkins Road
APG, MD 21005-5071

1 Mr. Patrick O'Neill
Chief, Combat Support Analysis Division
USAMSAA (ATTN: AMSRD-AMS-S)
392 Hopkins Road
APG, MD 21005-5071

1 Dr. James T. Treharne
OCA Division
Center for Army Analysis
6001 Goethals Road
Fort Belvoir, VA 22060-5230

1 Mr. Robert Barrett
Chief, International Activities
Center for Army Analysis
6001 Goethals Road
Fort Belvoir, VA 22060-5230

1 Mr. John Hughes
HQ, TRADOC Analysis Center (TRAC)
Programs & Resources Directorate (PRD)
255 Sedgwick Avenue
Fort Leavenworth, Kansas 66027-2345

1 Mr. Bob Barbier
TNO Defence, Security and Safety
Information and Operations
P.O. Box 96864, 2509 JG
The Hague, The Netherlands

1 Ms. Belinda Smeenk
TNO Defence, Security and Safety
Information and Operations
P.O. Box 96864, 2509 JG
The Hague, The Netherlands

Total external copies: 35

Total copies: 51

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) Defense Research and Development Canada – Centre for Operational Research and Analysis NDHQ, 101 Col By Drive, Ottawa ON K1A 0K2	2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable). UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

A Framework & Prototype for Modelling Army Force Generation

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Young, Chad ; Pall, Raman ; Ormrod, Mike

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc).	6b. NO. OF REFS (total cited in document)
December 2007	124	11

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Technical Memorandum

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defense Research and Development Canada – Centre for Operational Research and Analysis
NDHQ, 101 Col By Drive, Ottawa ON K1A 0K2

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Specify whether project or grant). N/A	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).
10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique.) DRDC CORA TM 2007–54	10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This report provides a comprehensive introduction to the first prototype of the Managed Readiness Simulator (MARS) without delving into the programming details. MARS is designed to provide the user with a collection of modelling constructs that may be combined to track and forecast the Army's ability to generate forces that fulfill government commitments in cases where repetitive, concurrent and/or random tasks compete for personnel, equipment, and facilities resources. An underpinning thrust behind the development of MARS has been the analysis of immediate issues related to the Canadian Army's concept of Managed Readiness. In this role, MARS is to be used to evaluate both capacity and scheduling issues pertaining to the allocation of personnel through the myriad of concurrent activities that take organizational units through a regenerative cycle towards deployment. This report is the second of three companion reports. Data requirements are put into context, concepts related to algorithm design are explained, and ideas critical for applying MARS and creating scenarios are introduced.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Managed Readiness
Force Generation
Force Employment
Operations Planning Process (OPP)
Force Development
Human Resources
Personnel Modelling

DEFENCE DÉFENSE

DRDC CORA

www.drdc-rddc.gc.ca