Defence Research and Development Canada Recherche et développement pour la défense Canada

DRDC | RDDC
technologysciencetechnologie

LIGHT ARMOURED VEHICLE CREW MODEL DEVELOPMENT

Brad Cain, CAE

Agata Lawrynczyk, CAE

Prepared by:
CAE Inc.
350 Legget Drive, Suite 200
Ottawa, ON
114032-003 Version 03
PSPC Contract Number: W7719-135217-01
Technical Authority: Vladimir Zotov
Contractor's date of publication: July 2018

Defence Research and Development Canada

Contract Report
DRDC-RDDC-2018-C155
September 2018

Canada

IMPORTANT INFORMATIVE STATEMENTS

Disclaimer: This document is not published by the Editorial Office of Defence Research and Development Canada, an agency of the Department of National Defence of Canada but is to be catalogued in the Canadian Defence Information System (CANDIS), the national repository for Defence S&T documents. Her Majesty the Queen in Right of Canada (Department of National Defence) makes no representations or warranties, expressed or implied, of any kind whatsoever, and assumes no liability for the accuracy, reliability, completeness, currency or usefulness of any information, product, process or material included in this document. Nothing in this document should be interpreted as an endorsement for the specific use of any tool, technique or process examined in it. Any reliance on, or use of, any information, product, process or material included in this document is at the sole risk of the person so using it or relying on it. Canada does not assume any liability in respect of any damages or losses arising out of or in connection with the use of, or reliance on, any information, product, process or material included in this document.

This document was reviewed for Controlled Goods by Defence Research and Development Canada using the Schedule to the *Defence Production Act*.

CAE Inc.

350 Legget Drive, Suite 200
Ottawa, ON, Canada K2K 2W7
Tel: 613-247-0342
Fax: 613-271-0963

LIGHT ARMOURED VEHICLE CREW MODEL DEVELOPMENT

CONTRACT #: W7719-135217-01, CALL-UP #4

FOR

DR. VLADIMIR ZOTOV

Defence Research and Development Canada
Human-Systems Integration Section
1133 Sheppard Ave. West, Toronto, ON M3K 2C9

06 July 2018

Document No. 114032-003 Version 03

APPROVAL SHEET

Document No. 114032-003 Version 03

Document Name: LAV Crew Model Development

Primary Author

Name **Eric Joncas on behalf of Brad Cain**

Position Senior Consultant, Advanced Solutions, Human Factors, CAE Defence & Security – CAE Canada

Reviewer

Name **Jaff Guo**

Position Technical Specialist, Human Factors, CAE Defence & Security – CAE Canada

Secondary Author

Name **Eric Joncas on behalf of Agata Lawrynczyk**

Position Junior Human Factors Consultant, Human Factors, CAE Defence & Security – CAE Canada

Approval

Name **Eric Joncas**

Position Project Manager, Performance Solutions, CAE Defence & Security – CAE Canada

Tertiary Author

Name **Michael Truong**

Position Software Developer, CAE Defence & Security – CAE Canada

REVISION HISTORY

Revision	Reason for Change	Origin Date
Version 01 DRAFT A	Initial document issued.	16 November 2017
Version 01	Official document (based on changes to updated draft) issued.	28 November 2017
Version 02	Section A.2: IPME HBR2 Plugin Debugging added.	09 May 2018
Version 03	Updated with French translation of the abstract.	06 July 2018

This document has been updated to Version 03 and changes are denoted by a black line in the left-hand margin.

TABLE OF CONTENTS

1 INTRODUCTION 1
1.1 Background 1
1.2 Objective 1
1.3 Other Project Reports 2

2 APPROACH 3
2.1 LAV Crew Model Development 3
2.1.1 Task Network Development 3
2.1.2 Speech Recognition Development 5
2.1.3 Speech Production Development 6
2.1.4 HTA Task List Tool 6

3 IMPLEMENTATION 7
3.1 LAV Crew Model Description 7
3.1.1 Composite LAV Crew Model 9
3.1.2 Prove Weapons Drill 9
3.1.3 Opening Up Procedure 9
3.1.4 Turret Power Up Procedure 12
3.1.5 Action Drill 12
3.1.6 Fire Order and Engagement 16
3.1.7 Misfire and Stoppage Drills 22
3.1.8 Start Mode Drill 23
3.1.9 Turret Make Safe Drill 30
3.1.10 Turret Power Down Procedure 30
3.1.11 Close Down Procedure 30
3.2 Model Verification 34
3.3 Speech Recognition Implementation 35
3.3.1 Modifying HBR2Plugin.dll.config 35
3.3.2 Modifying Grammar Files 36
3.4 Speech Production 40
3.5 IPME HTA Task List Tool 41

4 SOFTWARE INSTALLATION 43
4.1 IPME 43
4.2 Speech Recognition and Production 43
4.3 IPME Speech Plugin 43
4.4 Speech Files Source Code 44

5 MODEL EXECUTION 45

6 COMPUTER REQUIREMENTS 46

7 RECOMMENDATIONS 48

8 CONCLUSION 50

9 REFERENCES 51

APPENDIX A LAV CREW MODULE EXECUTION A-1

A.1 Setup for Drill Testing A-1
A.2 IPME HBR2 Plugin Debugging A-2
A.3 25mm Misfire Drill, Immediate Action A-4
A.3.1 Misfire Drill, BPI in SEAR: Drill a1/a1A (LAV III) or Ser #1 (LAV 6) A-4
A.3.2 Misfire Drill, BPI in SEAR, Drill a1B (LAV III) or Ser #2 (LAV 6) A-5
A.3.3 Misfire Drill, BPI in MISFIRE, Cannon Manual Safety ON: Drill a1B (LAV III) or Ser #2 (LAV 6) A-6
A.3.4 Misfire Drill, BPI in MISFIRE, Cannon Manual Safety OFF: Drill a1B (LAV III) or Ser #3 (LAV 6) A-6
A.4 COAX Immediate Action Drill (Stoppage) A-7
A.4.1 Gun Fails to Fire on Initial Burst – Ammo Select Wrong; Drill a1 (LAV III) or SER #1 (LAV 6) A-7
A.4.2 Gun Fails to fire on Initial Burst – Ammo Select Correct; COAX Safety Engaged; Drill a2 (LAV III) or SER #1 (LAV 6) A-7
A.4.3 Gun Fails to Fire on Initial Burst – Ammo Select Correct; COAX Safety Set to FIRE; Drill a2 (LAV III) or SER #1 (LAV 6) A-8
A.4.4 Gun Fails to Fire on Initial Burst – Ammo Select Correct – Working Parts Fully to the Rear; Drill a3/4/5 (LAV III) or SER #1 (LAV 6) A-9
A.5 Fire Order and Engagement Drills A-10
A.6 Action Drill A-12
A.7 Start Mode A-13
A.8 Turret Power Up or Down A-13
A.9 Procedures without Crew Interaction A-13

APPENDIX B CONSOLIDATED MODEL VARIABLE LIST B-1

APPENDIX C SPEECH RECOGNITION FILES C-1

C.1.1 HBR2Blugin.dll.config C-1
C.1.2 GunnerCommands.grxml C-6
C.1.3 Number.grxml C-37

APPENDIX D COMPONENT MODEL TASK LISTS D-1

D.1 Prove Weapons Drill Task List D-1
D.2 Opening Up Drill Task List D-3
D.3 Turret Power Up Procedure Task List D-8
D.4 Action Drill Task List D-9
D.5 Fire Order and Engagement Task List D-12
D.6 Misfire and Stoppage Immediate Action Drills Task List D-15
D.7 Start Mode Drill Task List D-18

D.8 Turret Make Safe Drill Task List D-19
D.9 Close Down Procedure Task List D-20
D.10 Turret Power Down Procedure Task List D-21

APPENDIX E LAV CREW DIALOGUE E-1
E.1 Prove Weapons Drill Verbal Communications E-1
E.2 Opening Up Procedure E-2
E.3 Power Up Procedure E-2
E.4 Action Drill Verbal Communications E-2
E.5 Start Mode Drill Verbal Communications E-3
E.6 Turret Make Safe Drill E-4
E.7 Power Down Procedure E-4
E.8 25mm Misfire Drill Verbal Communications E-5
E.9 Coaxial MG Stoppage Drill Verbal Communications E-5
E.10 Firing Orders E-6
E.11 Speech Recognition Verification Results E-8

APPENDIX F IPME PRX FILE XML SCHEMA F-1

LIST OF FIGURES

Figure 3-1: Example IPME Operator Task Description Interface.................................... 8
Figure 3-2: Composite LAV Crew Model 10
Figure 3-3: Prove Weapons Drill 10
Figure 3-4: Opening Up Procedure (Showing CC, GNR, and DRV Activities) – Top-Level Representation 11
Figure 3-5: Turret Power Up Procedure 13
Figure 3-6: Action Drill – Top-Level Representation.. 14
Figure 3-7: DRV Action Drill Tasks Preparing the Driver Compartment 14
Figure 3-8: Turret Crew Action Drill Activities 15
Figure 3-9: Top-Level Turret Crew Activities during Engagement................................ 18
Figure 3-10: Composition and Issuing a Fire Order 18
Figure 3-11: CC's Activities Confirming the GNR’s Firing Solution............................... 19
Figure 3-12: GNR's Actions while Firing on the Target 19
Figure 3-13: GNR's Lasing Actions to Measure Target Range..................................... 20
Figure 3-14: GNR Activities Configuring the Weapon System when CC Initiates Engagement... 20
Figure 3-15: GNR Activities Configuring the Weapon System when CC Initiates Engagement... 21
Figure 3-16: Misfire and Stoppage Drill Top Level Networks 22
Figure 3-17: COAX MG Stoppage Immediate Action Drill 24
Figure 3-18: Checking CDA Panel for Indications of Weapon Control System Faults... 25
Figure 3-19: Misfire Drill for the 25mm Cannon.. 26
Figure 3-20: CC Misfire Drill Problem Solving when BPI is in Sear Position 27
Figure 3-21: CC Misfire Drill Problem Solving when BPI is in Misfire Position 27
Figure 3-22: Enlarged Image of the GNR's Misfire Drill Task Network from Figure 3-19. 28
Figure 3-23: Start Mode Drill Task Networks.. 29
Figure 3-24: Turret Make Safe Task Network Model.. 31
Figure 3-25: Turret Power Down Procedure Task Network Model 32
Figure 3-26: Close Down Procedure Task Network Model... 33
Figure 3-27: Main Dialogue Window for the HTA Task Information Extraction Tool...... 42
Figure 3-28: Sample Output of Task and Network Information Obtained from IPME HTA Extraction Tool 42

Figure A-1: Selecting Task Network Variables to be Observed during Execution A-1
Figure A-2: Execution and Animation Settings for Task Network Testing A-2
Figure A-3: IPME Sim Engine Console A-3

Figure A-4: User Debug Window A-4

LIST OF TABLES

Table D-1: Prove Weapons Drill Task List D-1
Table D-2: Opening Up Drill Task List D-3
Table D-3: Turret Power Up Procedure Task List D-8
Table D-4: Action Drill Task List D-9
Table D-5: Fire Order and Engagement Task List D-12
Table D-6: Misfire and Stoppage Immediate Action Drills Task List D-15
Table D-7: Start Mode Drill Task List D-18
Table D-8: Turret Make Safe Drill Task List D-19
Table D-9: Close Down Procedure Task List D-20
Table D-10: Turret Power Down Procedure Task List D-21
Table E-1: Prove Weapons Drill Verbal Communications E-1
Table E-2: Power Up Procedure Verbal Communications E-2
Table E-3: Action Drill Verbal Communications E-2
Table E-4: Start Mode Drill Verbal Communications E-3
Table E-5: Turret Make Safe Drill Communications E-4
Table E-6: Power Down Procedure Verbal Communications E-4
Table E-7: Misfire Drill Verbal Communications E-5
Table E-8: Stoppage Drill Verbal Communications E-5
Table E-9: Engagement Fire Order Verbal Communications E-6
Table E-10: Speech Recognition Verification Results E-8

LIST OF ACRONYMS AND DEFINITIONS

AFESS	Automatic Fire Extinguisher/Suppression System
APC	Armoured Personnel Carrier
AUX	Auxiliary
BIT	Built In Test
BMP	Boyevaya Mashina Pekhoty
BPI	Bolt Position Indicator
CAF	Canadian Armed Forces
CC	Crew Commander
CCP	Commander control panel
CCTD	Crew Commander Tactical Display
CDA	Control display assembly
COAX	Coaxial
CSV	Comma Separated Variable
CTC	Combat Training Centre
DAGR	Defense Advanced GPS Receiver
DCB	Driver Control Box
DLL	Dynamic Link Library
DND	Department of National Defence
DRDC	Defence Research and Development Canada
DRV	Driver
DVA	Driver Visual Augmentation
ECPK	Exposed Crew Protection Kits
FO	Fire Order
FOV	Field of View
GCP	Guidance Control Panel
GDU	Gunner Display Unit
GNR	Gunner
GPS	Global Positioning System
GPU	Graphics Processor Unit
HBR	Human Behaviour Representation
HDD	Hard Disk Drive
HE	High Explosive
HTA	Hierarchical Task Analysis

IA	Immediate Action
ICGS	Interim Crew Gunnery System
II	Image Intensifier
IPME	Integrated Performance Modelling Environment
IRTAS	Infra-red Target Acquisition System
LASER	Light Amplification By Stimulated Emission Radiation
LHS	Left Hand Side
LOS	Line of Sight
LRF	Laser Range Finder
LWD	Laser Warning Display
LWR	Laser Warning Receiver
LWS	Laser Warning System
MA&D	Micro Analysis and Design (Alion, Inc.)
MBGD	Multi-barrel grenade dispenser
MG	Machine Gun
MS	Microsoft
NAV	Navigation
NFOV	Narrow Field Of View
PDA	Power distribution assembly
POPIP	Prediction of Performance/Information Processing
RHS	Right Hand Side
SAPI	Speech Application Program Interface
SER	Serial
SME	Subject Matter Expert
SSD	Solid State Drive
STAB	Stabilize
USB	Universal Serial Bus
WFOV	Wide Field Of View
WPN	Weapon
XOR	Exclusive OR (Boolean logic)

ABSTRACT

This document describes the creation of a behavioural model of a Canadian Armed Forces (CAF) Light Armoured Vehicle (LAV) crew performing drills and procedures. This preliminary work is intended to support a study of a human behaviour representation's (HBR) effectiveness for replacing role players in computer-based training. This LAV crew executable model is developed as an Integrated Performance Modelling Environment (IPME) Hierarchical Task Analysis (HTA) representation of crew activities that will be networked during subsequent work with a LAV simulator used by the CAF. The LAV crew behavioural model focuses on the turret crew—the Crew Commander and Gunner—performing their roles, although the intended use is as a Gunner simulation for assessing the effectiveness of HBR for training a Crew Commander; the Crew Commander representation is included to support verification and future assessments, such as the training effectiveness for a Gunner using a Crew Commander HBR. The LAV crew behavioural model incorporates speech recognition and speech production software leveraged from previous work, allowing a Crew Commander student to interact with the Gunner simulation verbally using expressions consistent with the CAF Armour School doctrine. The crew behavioural model procedures were verified using doctrine checklists and by CAF subject matter experts.

RESUME

Ce document décrit la création d'un modèle de comportement d'un équipage de véhicules blindés légers (VBL) des Forces armées canadiennes (FAC) effectuant des exercices et des procédures. Ce travail préliminaire est prévu pour soutenir une étude de l'efficacité d’une représentation de comportement humain (RCH) pour remplacer les acteurs dans la formation assistée par ordinateur. Ce modèle d'équipage de VBL est conçu avec le logiciel « Integrated Performance Modeling Environment » (IPME) comme une représentation hiérarchique des activités de l'équipage qui sera mise en réseau au cours de travaux subséquents avec un simulateur de VBL utilisé par les FAC. Le modèle comportemental de l'équipage du VBL se concentre sur les activités de l'équipage de la tourelle - le commandant d'équipage et l'artilleur - bien que l'utilisation prévue soit une simulation de tireur pour évaluer l'efficacité de RCH pour l'entraînement du commandant d'équipage; la représentation du commandant d'équipage est incluse pour permettre la vérification et les évaluations futures, comme par exemple l'efficacité de la formation d'un artilleur utilisant une RCH du commandant d’équipage. Le modèle comportemental de l'équipage du VBL intègre un logiciel de reconnaissance vocale et de production de la parole tiré d'un travail antérieur, permettant à un élève en formation pour devenir commandant d'équipage d'interagir verbalement avec la simulation de l’artilleur en utilisant des expressions conformes à la doctrine de l’école de l’arme blindée des FAC. Les procédures du modèle comportemental de l'équipage ont été vérifiées à l'aide de listes de contrôle de la doctrine et d'experts en la matière des FAC.

1 INTRODUCTION

This document, prepared by CAE Inc. (CAE), is for Defence Research and Development Canada (DRDC) under contract number W7719-135217-01 Call-up #4: Development of a Light Armoured Vehicle (LAV) crew model, describes the collection of information and development of a Hierarchical Task Analysis (HTA) structured, executable model of Canadian Armed Forces (CAF) LAV crew actions in the Integrated Performance Modelling Environment (IPME[1]).

1.1 Background

The CAF, as with many allied nations, are looking to simulation both as a means to provide appropriate levels of training as well as to provide training in a cost-effective manner. To accomplish these objectives, it is important to understand what features of simulation are effective for the training objectives.

DRDC is exploring whether virtual crew members can be effectively employed to train armoured vehicle crew members to provide a collective training context even when some crew members are not available. The concept of using a virtual crew is intended to allow individual training that optimizes collective training in a crew context. This may be considered as a complementary approach to large-formation training provided with computer-generated forces or semi-automated forces, but is directed towards individual or vehicle crew training. The focus on modelling individual operators is often referred to as human behaviour representation (HBR), an approach that provides opportunities to include detailed human capability and performance models that can moderate task execution.

Exploring the effectiveness of crew training using virtual, HBR crew members are expected to provide insight to determine whether it is a cost-effective investment within the greater CAF training programme. A previous training effectiveness study for the Landing Signals Officer role in a helicopter-deck landing scenario (Cain, Magee, & Belyavin, 2011; Cain, Magee, & Kersten, 2011) indicated that the Sea King HBR Pilot model was effective for learning procedures by the Landing Signals Officer while ashore, without the cost or inconvenience of training while underway at sea. Without additional, diverse simulation-based training effectiveness data, cost-effectiveness cannot be determined reliably and its usefulness in specific domains will be speculative.

1.2 Objective

The intent of LAV crew modelling is to create a behavioural simulation of crew actions that is suitable for supporting training of other crew members performing selected drills or procedures in a LAV vehicle simulation (MacQuarrie, 2017). Each member of the crew will be represented individually within the crew model to facilitate substitution of students at the three crew positions, crew commander (CC), gunner (GNR) or driver (DRV), while providing simulations for missing crew members using HBR.

[1] IPME: Alion Science and Technology, http://www.alionscience.com/Technologies/Human-Performance-Analysis/IPME

1.3 Other Project Reports

The following documents were also prepared for DRDC under contract W7719-135217-01 Call-up #4:

[114032-001]	Comparison of Armour Crew Drills: Potential for a Common Crew Model. CAE Document Number: 114032-001.
[114032-002]	LAV Crew IPME Model and Interim Crew Gunnery Simulation: Integration Strategy. CAE Document Number: 114032-002.

The above-listed documents describe an assessment of the feasibility of creating a common, generic armoured crew model that may be useful for simulating CAF armoured vehicle crews (Cain, 2017), and provide an assessment of the feasibility of integrating the LAV crew model with a LAV vehicle simulation to support crew procedures training (Truong, Cain, & Lawrynczyk, 2017).

2 APPROACH

2.1 LAV Crew Model Development

The LAV crew model comprises three main software elements:

1. IPME LAV crew behavioural model:
    a. LAV_CrewModel.prx.
2. speech recognition:
    a. HBR2Plugin.dll;
    b. HBR2Plugin.dll.config;
    c. Number.grxml; and
    d. Microsoft Speech Platform 11.
3. speech production:
    a. HBR2Plugin.dll; and
    b. AT&T Natural Voices with voice files.

Development of these three components are described in subsequent sections. The complete model will function with minimal user input, although there are several contextually relevant IPME variables the user may change to tailor the simulation to different situations (e.g., LAV vehicle variant; target type). Information about these user-adjustable variables that affect simulation may be found in Appendix A. Model execution is described in Section 5.

The HBR model (i.e., LAV_CrewModel.prx) comprises several component models that may be added or removed to customize the crew model to study requirements. Each of the component models may be executed individually to simplify modification and verification.

The model includes common crew drills and procedures outlined in the LAV aide memoires (DND, 2015, 2016a). Some drills and procedures were not included (e.g., Nine Daily Tests, Weapon System Upload) as these were considered irrelevant to the anticipated study.

2.1.1 Task Network Development

CAE developed an IPME model of the LAV crew, documenting crew member behaviours as an HTA (Annett, Duncan, Stammers, & Gray, 1971; Annett & Stanton, 2000) representation of the activities’ component tasks. This approach substitutes the IPME task flow diagrams in place of the more traditional task list diagrams originally proposed by Annett and Duncan (1967), but retains the essential, hierarchical, task decomposition philosophy of their HTA method. The LAV crew IPME HTA model expresses the associated tasks and communications as well as logical C# expressions required to make it executable within the IPME simulation framework. However, the HTA representation is superficial and does not include many of the data elements that a full HTA would typically include for applications such as system design or training needs analyses. A description of the IPME LAV crew model is in Section 3.1.

2.1.1.1 Information Collection

The first step in the model development was to review documentation and training standards for the LAV CC provided by the DRDC (DND, 2014); additional general information was also found on-line in unvetted, public forums. These documents were reviewed to provide contextual orientation for the CAE team. However, these documents contained little detail about specific crew tasks.

Subsequently, CAE and DRDC personnel met with Canadian Army LAV turret crew subject matter experts (SMEs) from the Combat Training Centre (CTC) in Gagetown, New Brunswick, from 10–13 January 2017. The SMEs provided descriptions of the LAV operational environment to clarify the context and background information for the initial framing of the model. The SMEs confirmed the drills in the documents represent a standard procedure that is taught and expected in practice to ensure the LAV is used effectively, efficiently, and safely.

CTC provided relevant documentation on the important operational crew drills and procedures (DND, 2004, 2015, 2016a, 2016b) used to create the procedural task flows in IPME. These documents describe the crew actions during various procedures and drills used during operations in the LAV. The descriptions of the drills provide explicit, step-by-step actions for each crew member, including verbal communications. However, these documents do not explicitly describe the task goals, cues, or action details. In most instances, the task goals and cues can be reliably inferred from the procedural descriptions, while the specific details about a task's physical actions were not generally required for the modelling.

The SMEs demonstrated several of the common drills in a LAV III, which provided rudimentary task timings for these procedures. Discussions also identified differences in each turret crew member's role between the LAV III and LAV 6.0 variants. There was no LAV DRV SME available during the Gagetown site visits, so the focus of the model is on the activities of the LAV turret crew (i.e., the CC and the GNR). Verbal communications, both initiated and received, were identified from the documentation and included in the model. Subsequent conversations with developers of the interim crew gunnery system (ICGS) LAV vehicle simulation (i.e., ADGA Group Consultants, Inc., Kingston, Ontario) indicated simulating vehicle manoeuvring is very difficult in the ICGS virtual environment as it relies on visual perception to interpret terrain and cultural features. Because of this, the DRV representation in the LAV crew model is superficial and not currently suitable for vehicle control.

2.1.1.2 LAV Crew Modelling

The LAV crew model that focussed on the CC and GNR activities was presented to an SME during a subsequent visit to CTC Gagetown, from 20–21 February 2017. The hierarchical layout and procedures were presented as task flows for verification. Discussions about the logical tasks flows clarified many of the procedures described in the LAV documents. Modifications or corrections were made to the IPME LAV crew model based on these SME discussions.

Subsequently, the Application of Fire documents (DND, 2004, 2016b) were reviewed to provide additional task descriptions. However, much of the background information appears to be tacit knowledge acquired by crew members through experience and instruction. In many instances, task goals may be identified from the nature of the task; in most cases, information sources were inferred from the task description or control panel's abbreviations reported in the checklists.

Most LAV crew activities in the documentation were organized as drills or procedures in serial, textual flow charts that present activities sequentially amongst operators. While this form of presentation is suitable for checklists and is easy to follow, it presents complications for modelling activities efficiently and avoiding task duplication. Therefore, it was decided to initially capture crew activities in a way similar to that presented in the documentation, but keep each crew members' role as separate task networks so drills could be reorganized by crew members. Thus, execution of each crew member's tasks in a drill flowed sequentially and transitions were controlled by variable values corresponding to events such as verbal orders and reports. This modelling approach employs implied operator tasks corresponding to monitoring the environment in anticipation of an event, such as the appearance of a target, receiving an order or report initiating a corresponding task flow in the operator's portion of a drill or procedure.

Individual crew drills and procedures were modelled in separate IPME project files, typically referred to herein as "component models". This modelling approach allowed changes and verifications to each model module without disturbing the composite IPME LAV crew model that includes all of the networked modules. This approach was also most effective if a common variable library was used as it reduced the number of variables related to similar actions, which would improve understanding of the composite model. Changes within a module could be implemented within the composite model using copy/paste functionality between two instances of the IPME modelling environment. Keeping component models separate was also intended to facilitate study design requirements that may require a subset of crew drills and procedures.

Cues used in the IPME LAV crew model include environmental or contextual factors, simulation events, and verbal communications. Some environmental events were instantiated as placeholders in the model, anticipating ICGS capabilities based on information obtained during Task 5.3 of the Statement of Work, concerning integration specification for the crew model and the ICGS simulation (MacQuarrie, 2017).

2.1.2 Speech Recognition Development

The Microsoft Speech Platform 11[2] interface developed for IPME to recognize Leopard 2 CC expressions for an HBR task (Dubreuil, 2015) was leveraged for the LAV crew model. First, the CC's orders and reports that affect GNR actions were incorporated in a LAV crew grammar file that the speech recognition software engine used to constrain the word search operation. Following that, the LAV CC speech recognition expressions associated with the LAV crew

[2] Microsoft Speech Platform: https://msdn.microsoft.com/en-us/library/office/hh361572(v=office.14).aspx

model were used to assign corresponding values to the LAV crew model variables. Details about the modification of the original interface for the LAV crew speech recognition model are in Section 3.3 while installation instructions are in Section 4.3.

The current model implementation is restricted to recognizing CC expressions only as the DRDC indicated the CC would be the focus of a subsequent study. Implementation of speech recognition for the GNR is feasible, but multiple, concurrent role-speech recognition is not currently feasible. Development of a speech recognition system that can recognize and differentiate amongst multiple individual speakers would require implementing a mechanism to identify the source of the speech input to the speech recognition software (e.g., separate microphone channel identifiers).

2.1.3 Speech Production Development

The speech production (synthesis) system used in the LAV crew model differs from that used for the Leopard 2 model (Dubreuil, 2015). DRDC noted that the Microsoft (MS) Speech Application Program Interface (SAPI) speech production sounded too artificial so the AT&T Natural Voices[3] package used in an HBR model for a Landing Signals Officer training study was applied in an attempt to improve synthesized speech (Cain, Magee, & Kersten, 2011). The HBR2Plugin.dll.config file was modified so it could use the Natural Voices software as well as incorporate IPME variables that simplify using speech production within a model. A description of the speech production implementation is in Section 3.4.

2.1.4 HTA Task List Tool

While IPME provides graphical and spreadsheet presentations of operator tasks, this is largely restricted for use within the IPME modelling environment. The graphical task networks used in the current work are equivalent to individual levels of the traditional HTA task lists, but the IPME task flows convey additional information about inter-dependencies or the execution order of the tasks. IPME does not provide a convenient method of listing operator tasks or task attributes suitable for inclusion in reports. Therefore, a software utility was developed to parse IPME project files (.PRX), which are XML[4] text files, to extract object attributes (e.g., task names, task durations, task descriptions,) and store the data in a CSV[5] file format that may be readily imported into a spreadsheet for further analysis or reporting. This tool could also be used to create an input file for a traditional HTA graphical task list.

[3] AT&T Natural Voices: http://www.wizzardsoftware.com/text-to-speech-sdk.php
[4] XML: Extensible Markup Language (text file)
[5] CSV: Comma Separated Variable (text file)

3 IMPLEMENTATION

3.1 LAV Crew Model Description

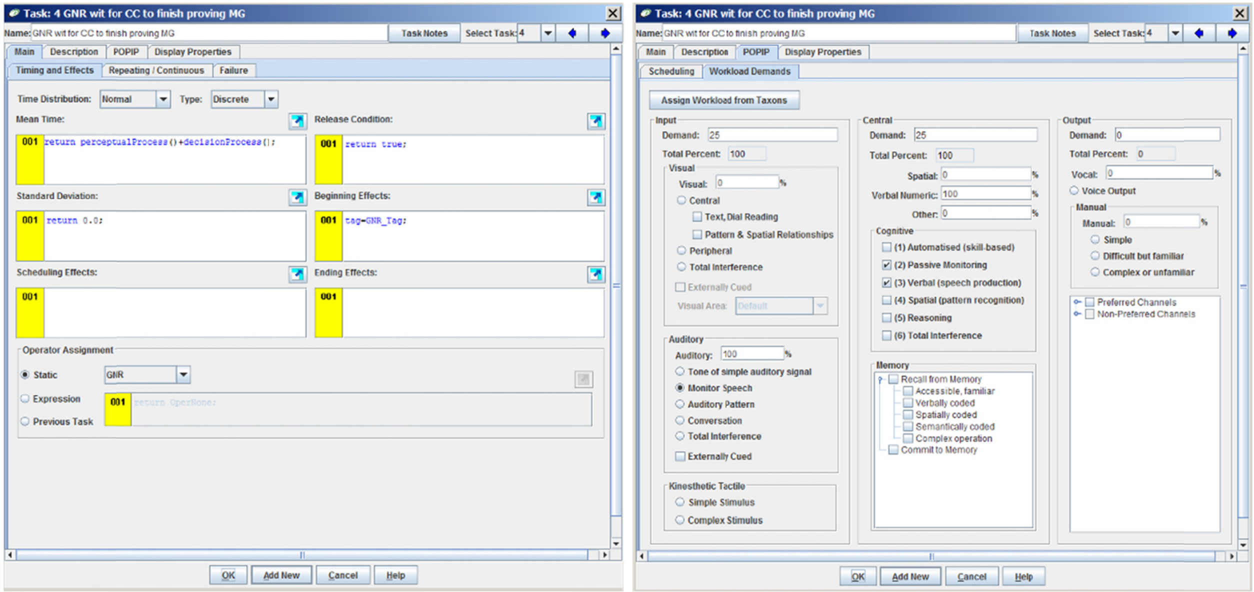
The following sections describe the composite LAV crew model as well as each of the component models. The IPME models comprise both LAV III and LAV 6.0 vehicle crew representations, primarily focussing on the interaction between the CC and the GNR, but also including the DRV where information was readily available. A list of variables used within the composite and component models is provided in Appendix B, citing the component model in which the variable is used—whether the variable corresponds to a speech recognition expression or the variable is used during integration with the LAV vehicle simulation.

The component models that populate each of the functional blocks of the composite model follow a similar design. CC tasks are represented in a network at the top of the component model IPME task network diagram window, while the GNR tasks are represented in a network at the bottom of the component model network diagram. Execution of each crew member task representations in these networks can be controlled separately by initializing the associated IPME LAV Crew model logical variable (i.e., HBR_CC, HBR_GNR, HBR_DRV) to true (enable execution) or false (disable execution).

The execution flow of the model, from task to task, is indicated by connecting arrows in the task network diagram; however, there may be dependencies amongst tasks in distinct networks. Although the CC and GNR networks are shown as distinct networks in each network diagram, they are coupled through the use of logical variables that represent events, such as orders or reports. The logical variables control task execution so that each operator performs tasks in the prescribed sequence. When both the CC and GNR roles are simulated, the coupling variables are assigned values within the model; when only the GNR role is simulated, the coupling variables are assigned values through the speech recognition system for the CC orders and reports, or by speech synthesis for the GNR reports.

The rectangles with square corners represent subnetworks used to organize details such as tasks or other networks so the higher-level representation is easier to understand; the network objects perform no role in the model execution. Rectangles with rounded corners represent operator tasks (white fill) or network control tasks (gray fill). These tasks typically contain executable expressions as well as task performance information (e.g., task demands), as shown in the left-hand pane of Figure 3-1, that control the flow of the task network and provide the mechanism to interact with the ICGS. Task field role descriptions can be found in the IPME user guide (MA&D, 2015). Network control tasks are used to provide execution control for the user, improve understanding, simplify network structures, and occasionally set variable states. Network control tasks do not represent LAV crew actions.

The LAV crew model only involved modifying the “main/timing and effects” and the “prediction of performance/information processing (POPIP)/workload demands” tabs of the task description interface of the operator tasks (i.e., the right-hand pane of Figure 3-1). The data in these fields affect the execution of the model and substitute for the more detailed activities the operator performs during the task. There are additional tasks that have a yellow fill in some of the component networks. The yellow fill in the network diagrams was used to indicate tasks that require further development during integration with the LAV vehicle simulation (ICGS).



Task: 4 GNR wit for CC to finish proving MG

Name: GNR wit for CC to finish proving MG

Select Task: 4

Time Distribution: Normal

Type: Discrete

Mean Time:

```
001 return perceptualProcess()+decisionProcess();
```

Release Condition:

```
001 return true;
```

Standard Deviation:

```
001 return 0.0;
```

Beginning Effects:

```
001 tag=GNR_Tag;
```

Scheduling Effects:

```
001
```

Ending Effects:

```
001
```

Operator Assignment

● Static GNR

○ Expression

```
001 return OperNone;
```

○ Previous Task

Task: 4 GNR wit for CC to finish proving MG

Name: GNR wit for CC to finish proving MG

Select Task: 4

Input

Demand: 25

Total Percent: 100

Visual

Visual: 0 %

○ Central

□ Text, Dial Reading

□ Pattern & Spatial Relationships

○ Peripheral

○ Total Interference

□ Externally Cued

Visual Area: Default

Auditory

Auditory: 100 %

○ Tone of simple auditory signal

● Monitor Speech

○ Auditory Pattern

○ Conversation

○ Total Interference

□ Externally Cued

Kinesthetic Tactile

○ Simple Stimulus

○ Complex Stimulus

Central

Demand: 25

Total Percent: 100

Spatial: 0 %

Verbal Numeric: 100 %

Other: 0 %

Cognitive

□ (1) Automatised (skill-based)

☑ (2) Passive Monitoring

☑ (3) Verbal (speech production)

□ (4) Spatial (pattern recognition)

□ (5) Reasoning

□ (6) Total Interference

Memory

- □ Recall from Memory
  - □ Accessible, familiar
  - □ Verbally coded
  - □ Spatially coded
  - □ Semantically coded
  - □ Complex operation
- □ Commit to Memory

Output

Demand: 0

Total Percent: 0

Vocal: 0 %

○ Voice Output

Manual

Manual: 0 %

○ Simple

○ Difficult but familiar

○ Complex or unfamiliar

□ Preferred Channels

□ Non-Preferred Channels

Figure 3-1: Example IPME Operator Task Description Interface

One activity that is not represented adequately in the LAV crew model is the supervision of the crew by the CC. It is the CC's responsibility (or a designated alternate when the CC is absent) to observe crew activities to ensure crew safety. This activity is complicated in practice, as many tasks are performed concurrently and the DRV actions typically are not observable by the CC. However, the most hazardous task—employment of the weapon system—is more readily observable and the CC's role supervising the GNR during weapon employment or verification is a significant element in turret crew training. However, it is expected that the CC role will be performed by a study subject rather than be simulated, so the superficial representation of the CC supervision tasks should have no consequence to the use of the model.

The execution duration for the individual operator tasks was not measured, so values were estimated using the IPME Micro Models (MA&D, 2015, p. 16–45; PDF p.417) that corresponded most closely to the task characteristics. The LAV crew model time defaults to seconds. In many tasks, manual actions and speaking occur simultaneously; in these cases, the task duration was selected as the greater of the manual and speaking time estimates. An operator trait "SpeakingRate" was defined in IPME to control the duration of the speech portion of the task, although this variable does not affect the speaking rate of the speech production software.

Individual tasks did not impose a significant mental demand on the operator. The workload demands were assigned a low-to-moderate value and the associated workload channel was identified in the POPIP workload assignment task tab. The workload demand is usually important during multi-tasking, but most of the LAV crew drills and procedures are designed to be performed serially to ensure reliability and safety. Thus, workload demands are not expected to affect execution of the network significantly.

3.1.1 Composite LAV Crew Model

The composite model that contains all of the networked component models is stored in the IPME project file “LAV_CrewModel.prx”. An image of the top level, composite LAV crew model is shown in Figure 3-2, showing each of the high-level functional networks (represented as rectangles). The arrows connecting the networks indicate the execution flow direction, including repetition of earlier functions (e.g., Fire Order and Engagement followed by Start Mode that can then repeat the Fire Order and Engagement). These top level, functional networks are executed individually; branches imply an exclusive OR (XOR) (i.e., the Start Model or the misfire drill may be performed, but not simultaneously). Tasks within the networks may be performed in parallel and depend on executable logic statements to control task flows.

3.1.2 Prove Weapons Drill

The Prove Weapons Drill (ProveWeaponsDrill.prx) is performed whenever the turret crew (CC and GNR) enter the turret and are unsure of the state of the weapons (i.e., whether the weapons are loaded). This may occur after a vehicle crew change or reassignment to a different vehicle. The Prove Weapons Drill is not typically performed when the vehicle has been in the crew’s control and they are aware of the weapon system state from previous knowledge. The top-level functional networks of the prove weapons drill are shown in Figure 3-3, with the CC activities in the upper network and the GNR activities in the lower network.

Each of the sub-networks within this model contain tasks that reflect specific operator actions required to achieve the implicit goal of the sub-network, which is typically reflected in the sub-network or task name.

3.1.3 Opening Up Procedure

The Opening Up Procedure (OpeningUpDrill.prx) is typically performed at the beginning of the operational day. The procedure involves inspecting the vehicle to ensure there is no damage or equipment conditions that would interfere with the operation of the vehicle. Each crew member has their own set of responsibilities, but the crew has no formal interaction. The high-level opening up procedure is shown in Figure 3-4. Although the Opening Up Procedure is presented linearly, there are no dependencies amongst the tasks. The execution of the tasks in the model represents the procedure from the aide memoires (DND, 2015, 2016a), but in practice, the tasks may be performed in any order.

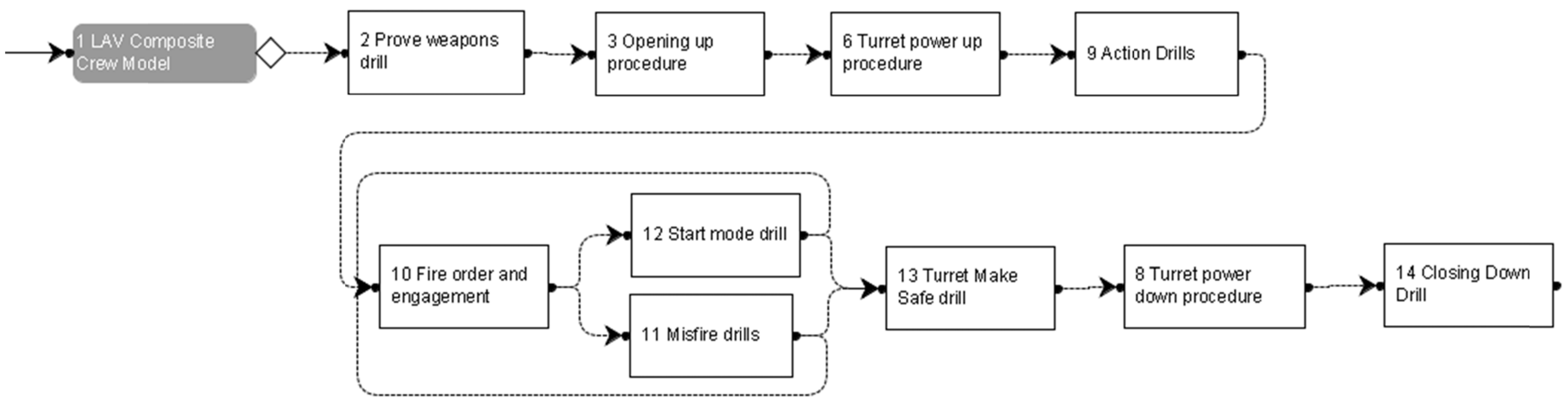

1 LAV Composite Crew Model
2 Prove weapons drill
3 Opening up procedure
6 Turret power up procedure
9 Action Drills
10 Fire order and engagement
12 Start mode drill
11 Misfire drills
13 Turret Make Safe drill
8 Turret power down procedure
14 Closing Down Drill

Figure 3-2: Composite LAV Crew Model

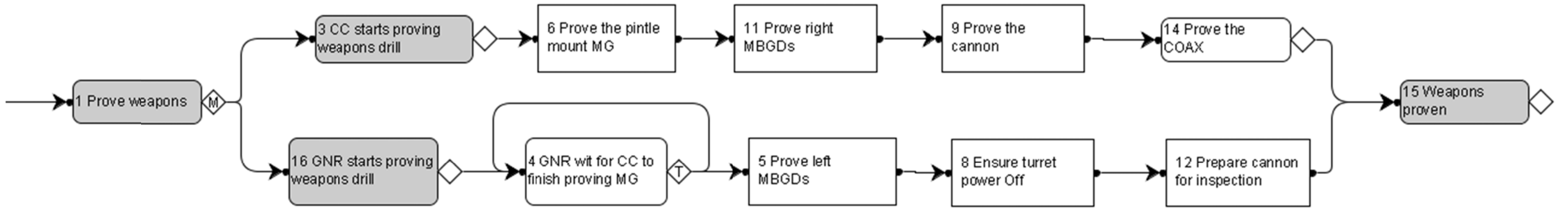

1 Prove weapons
M
3 CC starts proving weapons drill
6 Prove the pintle mount MG
11 Prove right MBGDs
9 Prove the cannon
14 Prove the COAX
16 GNR starts proving weapons drill
4 GNR wit for CC to finish proving MG
T
5 Prove left MBGDs
8 Ensure turret power Off
12 Prepare cannon for inspection
15 Weapons proven

Figure 3-3: Prove Weapons Drill

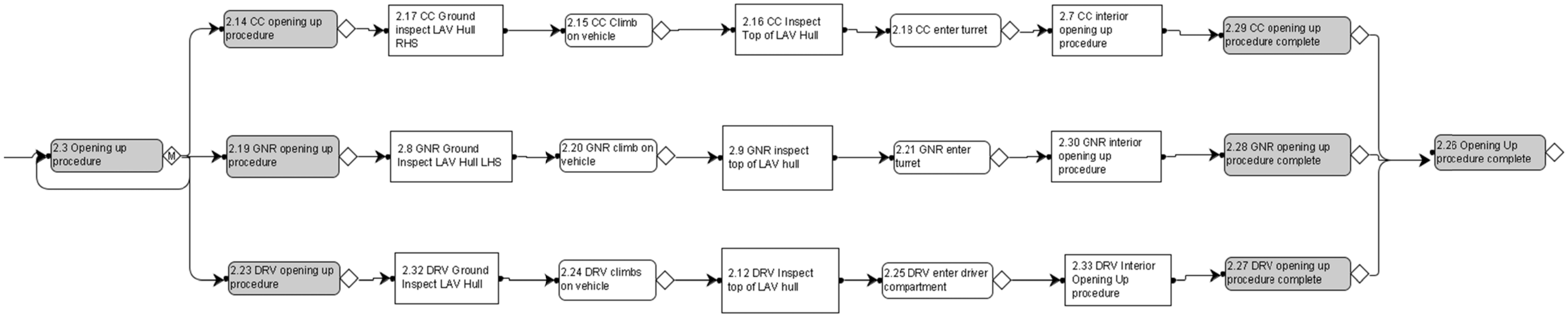

2.3 Opening up procedure
M
2.14 CC opening up procedure
2.17 CC Ground inspect LAV Hull RHS
2.15 CC Climb on vehicle
2.16 CC Inspect Top of LAV Hull
2.18 CC enter turret
2.7 CC interior opening up procedure
2.29 CC opening up procedure complete
2.19 GNR opening up procedure
2.8 GNR Ground Inspect LAV Hull LHS
2.20 GNR climb on vehicle
2.9 GNR inspect top of LAV hull
2.21 GNR enter turret
2.30 GNR interior opening up procedure
2.28 GNR opening up procedure complete
2.23 DRV opening up procedure
2.32 DRV Ground Inspect LAV Hull
2.24 DRV climbs on vehicle
2.12 DRV Inspect top of LAV hull
2.25 DRV enter driver compartment
2.33 DRV Interior Opening Up procedure
2.27 DRV opening up procedure complete
2.26 Opening Up procedure complete

Figure 3-4: Opening Up Procedure (Showing CC, GNR, and DRV Activities) – Top-Level Representation

3.1.4 Turret Power Up Procedure

The Turret Power Up Procedure, shown in Figure 3-5, is typically performed at the start of the operational day as part of the Opening Up Procedure or after a temporary shut down (e.g., for resupply). Both turret Power Up and Power Down procedures are stored in the same IPME project (TurretPowerUpDown.prx) although they are modelled as distinct task networks. The CC and the GNR perform tasks sequentially with the CC supervising the GNR performing the power up tasks, then performing the CC tasks. The DRV is not involved with the turret power up tasks, having completed the vehicle power on tasks during the opening up procedure.

3.1.5 Action Drill

The Action Drill (ActionDrill.prx) involves all crew members and represents the CC's intent to begin to employ the vehicle operationally. The top-level representation of the Action Drill model is shown in Figure 3-6. At this level, the CC has some responsibilities outside the vehicle to prepare for action (e.g., removing muzzle covers, securing equipment) followed by joint activities with the GNR in the turret; the DRV has separate activities in the driver compartment but does not interact formally with the turret crew.

The task “Assign arcs of responsibility” is an example of the tasks that will require explicit integration with the LAV vehicle simulation, including more advanced logical expressions to represent features such as operator perception (e.g., visually identifying objects in the simulation) and motor control (e.g., moving the turret). Currently, these tasks contain no executable expressions and would be part of the LAV vehicle crew model integration process.

The DRV's action drill tasks are shown in Figure 3-7 while the turret interior Action Drill activities are shown in Figure 3-8. These tasks configure the vehicle to a standard state in preparation for employment, including the general vehicle systems as well as the weapon systems. At completion of the Action Drill, the vehicle and crew are ready for employment.

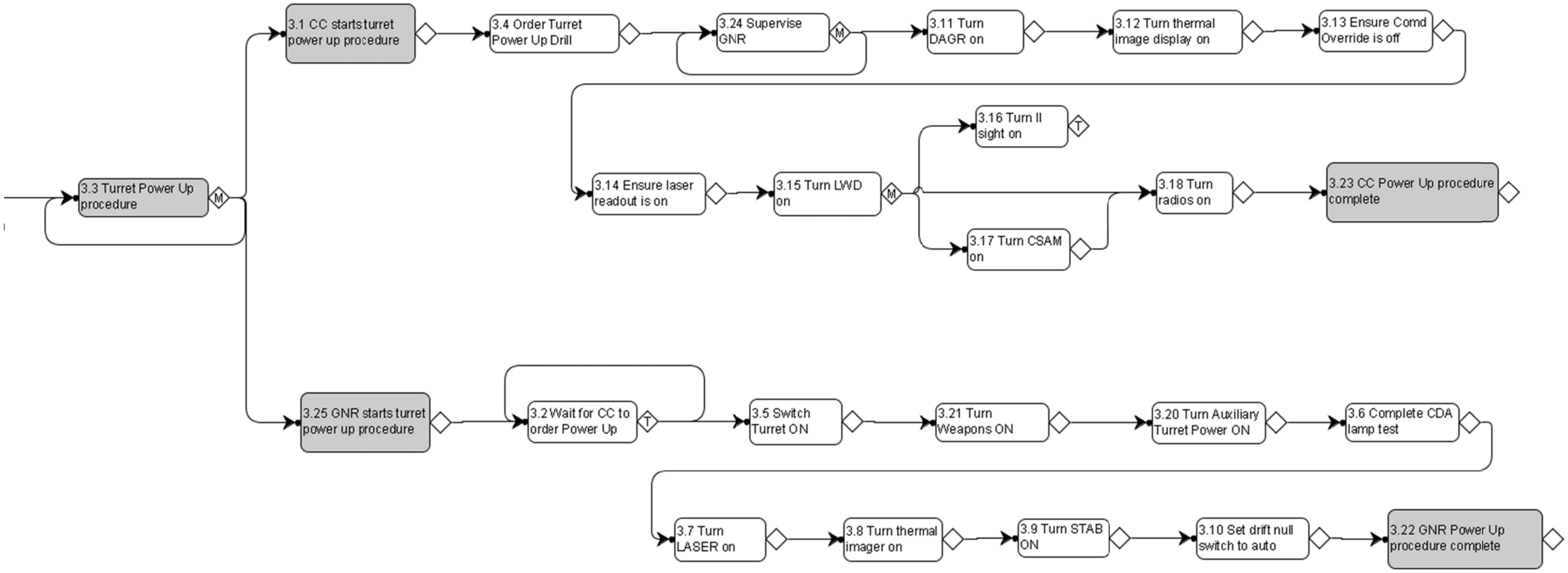

3.3 Turret Power Up procedure
M
3.1 CC starts turret power up procedure
3.4 Order Turret Power Up Drill
3.24 Supervise GNR
M
3.11 Turn DAGR on
3.12 Turn thermal image display on
3.13 Ensure Comd Override is off
3.14 Ensure laser readout is on
3.15 Turn LWD on
M
3.16 Turn II sight on
T
3.17 Turn CSAM on
3.18 Turn radios on
3.23 CC Power Up procedure complete
3.25 GNR starts turret power up procedure
3.2 Wait for CC to order Power Up
T
3.5 Switch Turret ON
3.21 Turn Weapons ON
3.20 Turn Auxiliary Turret Power ON
3.6 Complete CDA lamp test
3.7 Turn LASER on
3.8 Turn thermal imager on
3.9 Turn STAB ON
3.10 Set drift null switch to auto
3.22 GNR Power Up procedure complete

Figure 3-5: Turret Power Up Procedure

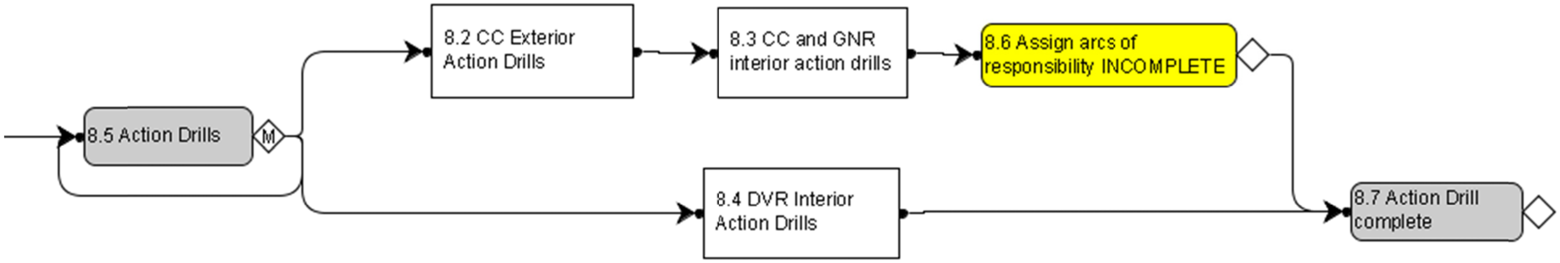

8.5 Action Drills
M
8.2 CC Exterior Action Drills
8.3 CC and GNR interior action drills
8.6 Assign arcs of responsibility INCOMPLETE
8.4 DVR Interior Action Drills
8.7 Action Drill complete

Figure 3-6: Action Drill – Top-Level Representation

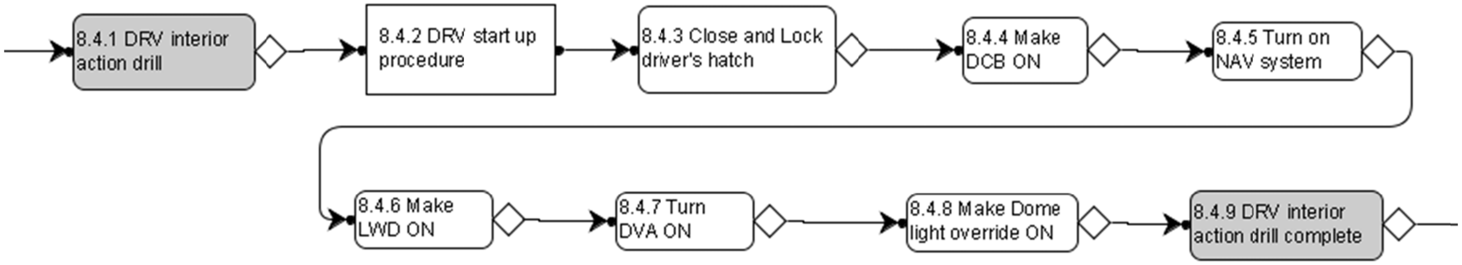

8.4.1 DRV interior action drill
8.4.2 DRV start up procedure
8.4.3 Close and Lock driver's hatch
8.4.4 Make DCB ON
8.4.5 Turn on NAV system
8.4.6 Make LWD ON
8.4.7 Turn DVA ON
8.4.8 Make Dome light override ON
8.4.9 DRV interior action drill complete

Figure 3-7: DRV Action Drill Tasks Preparing the Driver Compartment

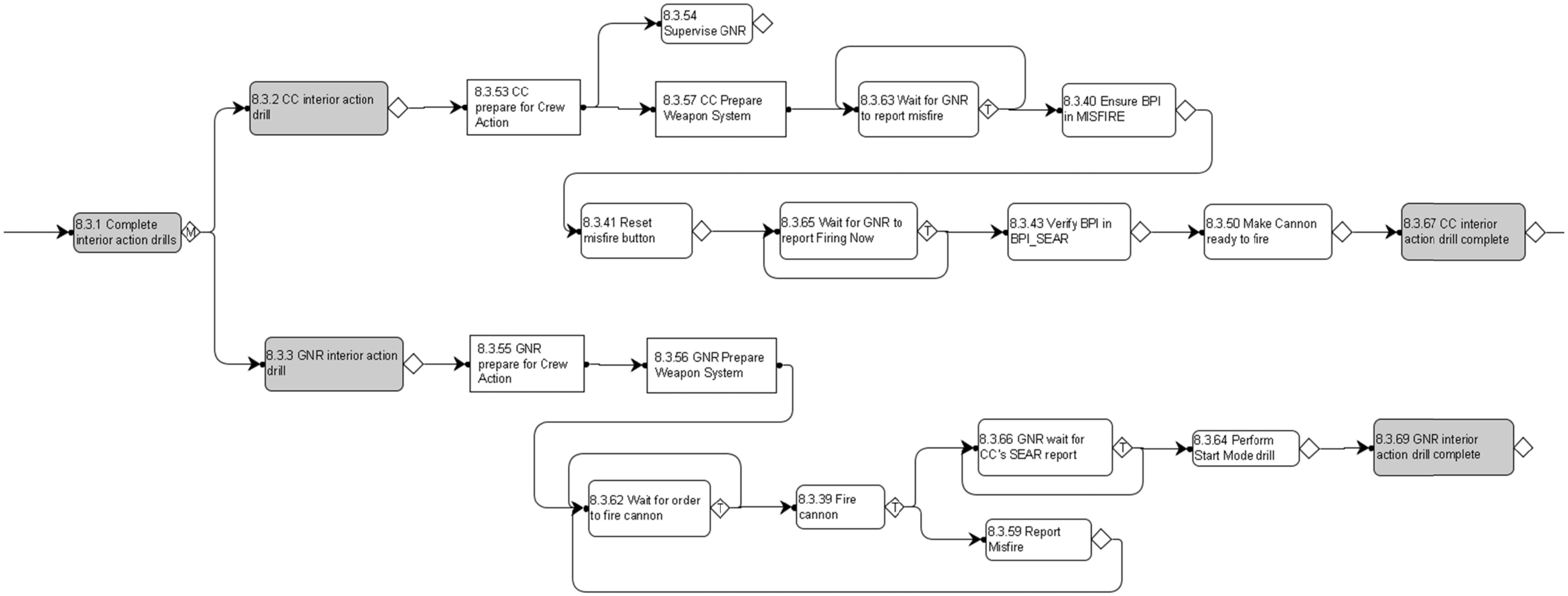

8.3.1 Complete interior action drills
M
8.3.2 CC interior action drill
8.3.53 CC prepare for Crew Action
8.3.54 Supervise GNR
8.3.57 CC Prepare Weapon System
8.3.63 Wait for GNR to report misfire
T
8.3.40 Ensure BPI in MISFIRE
8.3.41 Reset misfire button
8.3.65 Wait for GNR to report Firing Now
T
8.3.43 Verify BPI in BPI_SEAR
8.3.50 Make Cannon ready to fire
8.3.67 CC interior action drill complete
8.3.3 GNR interior action drill
8.3.55 GNR prepare for Crew Action
8.3.56 GNR Prepare Weapon System
8.3.62 Wait for order to fire cannon
T
8.3.39 Fire cannon
T
8.3.66 GNR wait for CC's SEAR report
T
8.3.64 Perform Start Mode drill
8.3.69 GNR interior action drill complete
8.3.59 Report Misfire

Figure 3-8: Turret Crew Action Drill Activities

3.1.6 Fire Order and Engagement

The Fire Order and Engagement component model (FireOrder_Engagement.prx), shown in Figure 3-9, represents the turret crew's activities while operating the weapon systems (which includes detecting a target, issuing a fire order, and firing the weapon). These activities are the most complicated networks of the crew model, comprising numerous alternative branches that capture options available for employing the weapon system. Typically, the CC is scanning the environment visually, either unaided or with binoculars, while standing up in the CC hatch, which is located on the top of the turret. The GNR scans the environment with the weapon sights, moving the turret to view different regions of the GNR's arc of responsibility. A target may be detected by any crew member; however, the CC is responsible for the executive order to fire on a target.

When the CC detects a target, the CC quickly compose a Fire Order as shown in Figure 3-10 and verbally relay information to the GNR. The GNR responds to the Fire Order, repeating the key elements to confirm the order was understood and implemented as directed. The CC may intercede, controlling the turret directly to get the GNR close to the target, then let the GNR continue with getting the target sighted while the CC maintains situation awareness. When the GNR believes the target is sighted, the GNR report "ON". At this point, the CC will drop back down into the turret to ensure the GNR has the correct target, aim point and range information entered in the weapon system, as shown in Figure 3-11.

The model exploits the CC's verbal request for information, "ON", to signal the GNR simulation that a Fire Order has been completely issued. This is a convenience that may cause execution complications with the assignment of scanning arcs of responsibility. If this strategy leads to performance problems, a more complex method of indicating the completion of a fire order will need to be developed.

If the CC is satisfied with the GNR's firing solution, the CC gives the order to fire and the GNR immediately fires the weapon and reports the "fall of shot", as shown in Figure 3-12. If the CC is not satisfied with the firing solution, the CC may correct the aim point, usually by directing the GNR verbally. The CC may also order the GNR to relase the target to confirm the range to the target, as shown in Figure 3-13. Currently, no representation of the GNR's aim point control has been implemented as this will require interaction with the ICGS simulation.

NOTE: "Relase" repeats the target range measurement using the laser range finder (LRF). (See Section 3.2.)

The GNR's task flow differs slightly if the engagement is initiated by the CC or by the GNR. If the CC identifies the target, the GNR gets the target in the sight, initially with a wide field of view (FOV), refining the aim point by narrowing the sight FOV, then enters the range estimate provided by the CC or lases the target as directed. If the GNR initiates the engagement, the GNR has detected the target with the sight in the wide FOV setting, then narrows the FOV to confirm that it is a target. The GNR then reports the contact, including the target type and a categorical distance (i.e., NEAR, MIDDLE, FAR) to the CC. The CC then takes control of the engagement as with a CC-initiated engagement. The two task flows for these situations are shown in Figure 3-14 and Figure 3-15.

The procedures for the LAV III and the LAV 6 differ slightly due to the advanced weapon system in the LAV 6. It is typical for the LAV 6 GNR to automatically lase a target without the CC's order as the LAV 6 ballistic computer is most effective with a laser measurement of the target's range and speed. For the LAV III, the GNR typically waits for the CC to order a laser range measurement. In some tactical situations, no lasing is used to prevent alerting the enemy to the LAV's presence.

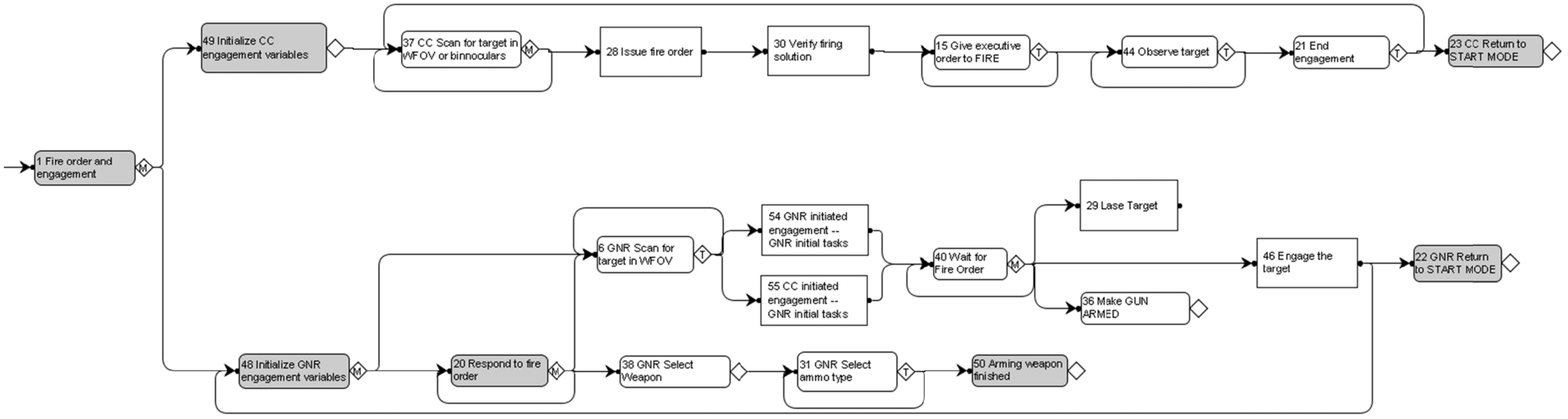

1 Fire order and engagement
M
49 Initialize CC engagement variables
37 CC Scan for target in WFOV or binnoculars
M
28 Issue fire order
30 Verify firing solution
15 Give executive order to FIRE
T
44 Observe target
T
21 End engagement
T
23 CC Return to START MODE
48 Initialize GNR engagement variables
M
6 GNR Scan for target in WFOV
T
54 GNR initiated engagement -- GNR initial tasks
55 CC initiated engagement -- GNR initial tasks
40 Wait for Fire Order
M
29 Lase Target
36 Make GUN ARMED
46 Engage the target
22 GNR Return to START MODE
20 Respond to fire order
M
38 GNR Select Weapon
31 GNR Select ammo type
T
50 Arming weapon finished

Figure 3-9: Top-Level Turret Crew Activities during Engagement

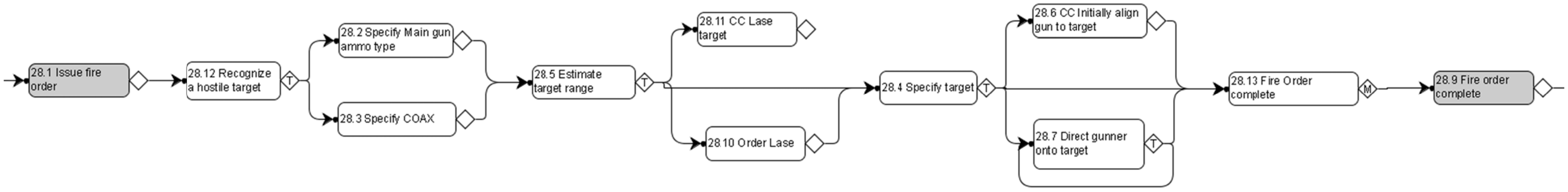

28.1 Issue fire order
28.12 Recognize a hostile target
T
28.2 Specify Main gun ammo type
28.3 Specify COAX
28.5 Estimate target range
T
28.11 CC Lase target
28.10 Order Lase
28.4 Specify target
T
28.6 CC Initially align gun to target
28.7 Direct gunner onto target
T
28.13 Fire Order complete
M
28.9 Fire order complete

Figure 3-10: Composition and Issuing a Fire Order

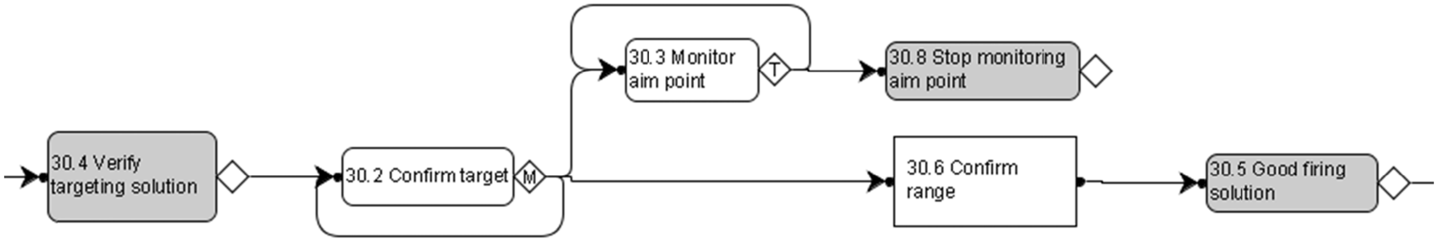

30.4 Verify targeting solution
30.2 Confirm target
M
30.3 Monitor aim point
T
30.8 Stop monitoring aim point
30.6 Confirm range
30.5 Good firing solution

Figure 3-11: CC's Activities Confirming the GNR's Firing Solution

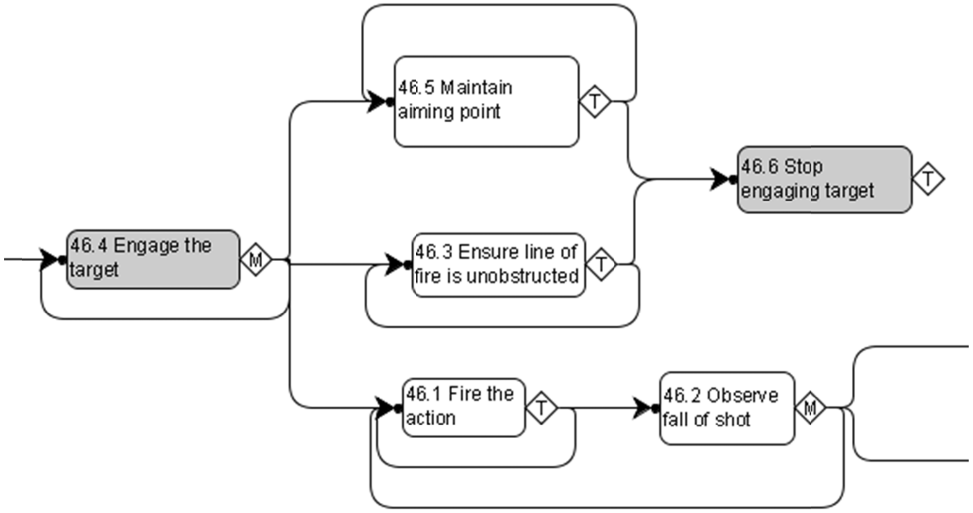

46.5 Maintain aiming point
T
46.6 Stop engaging target
T
46.4 Engage the target
M
46.3 Ensure line of fire is unobstructed
T
46.1 Fire the action
T
46.2 Observe fall of shot
M

Figure 3-12: GNR's Actions while Firing on the Target

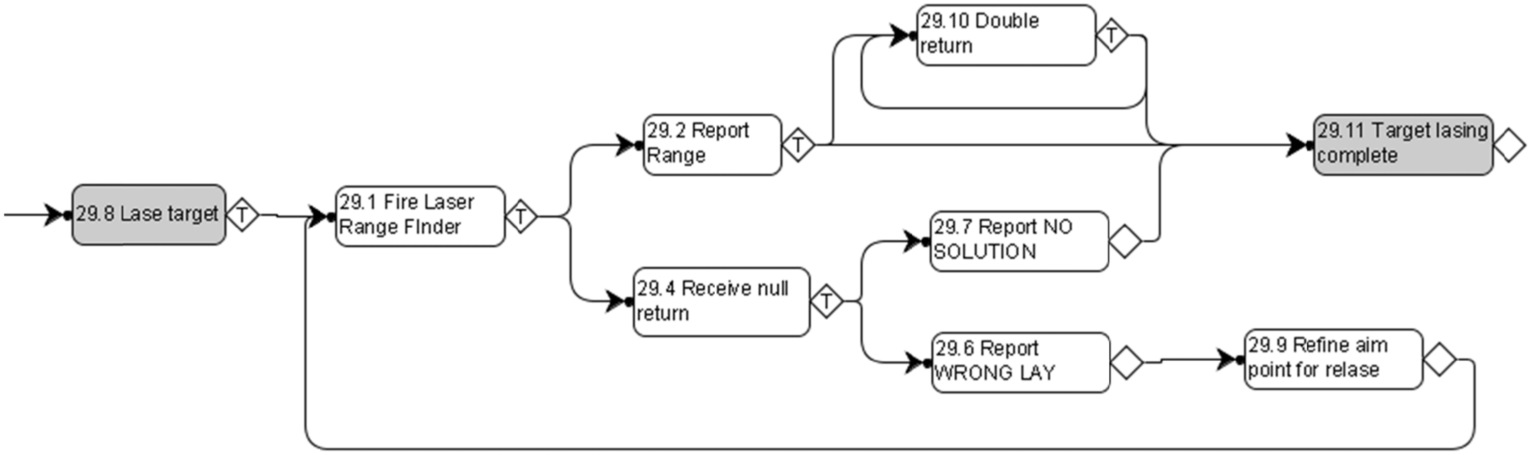


Figure 3-13: GNR's Lasing Actions to Measure Target Range

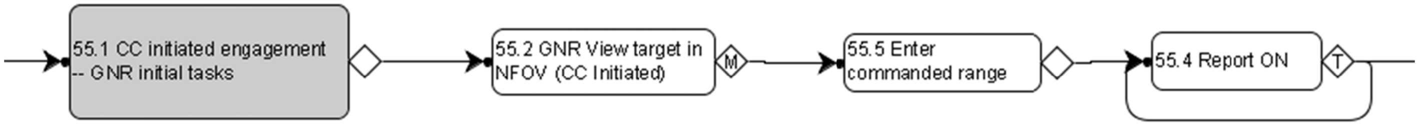

55.1 CC initiated engagement -- GNR initial tasks
55.2 GNR View target in NFOV (CC Initiated)
M
55.5 Enter commanded range
55.4 Report ON
T

Figure 3-14: GNR Activities Configuring the Weapon System when CC Initiates Engagement

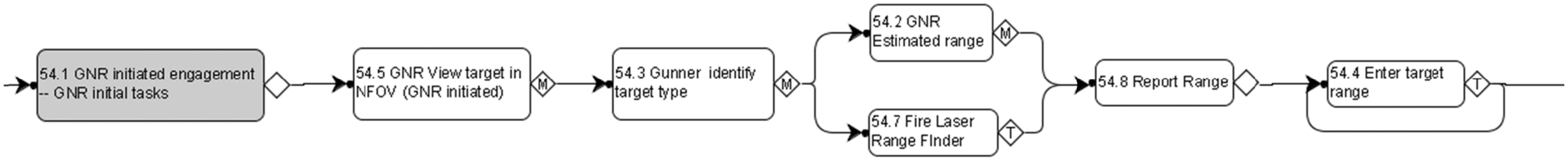

54.1 GNR initiated engagement -- GNR initial tasks
54.5 GNR View target in NFOV (GNR initiated)
M
54.3 Gunner identify target type
M
54.2 GNR Estimated range
M
54.7 Fire Laser Range FInder
T
54.8 Report Range
54.4 Enter target range
T

Figure 3-15: GNR Activities Configuring the Weapon System when CC Initiates Engagement

3.1.7 Misfire and Stoppage Drills

The Misfire (25 mm cannon) and Stoppage (COAX MG) Drills are performed when a fault occurs in the weapon system while firing. Turret crew proficiency at resolving a fault that prevents firing the LAV weapon system is critical for LAV survivability. Faults may arise within the weapons or supporting electronic systems in the vehicle. The Misfire Drill is applied when the 25mm cannon experiences a fault while the Stoppage Drill is applied for the coaxial (COAX) machine gun (MG) when a fault occurs; the different names are used to differentiate between the two weapons as some tasks are unique to each weapon, while others, such as control display assembly (CDA) electronic control checks, are common tasks. The COAX MG is the simpler of the two weapons, although as both weapons are controlled through the same electronic control system, some of the drill complexity is also shared.

Misfire and Stoppage Drills are located in a common IPME project file (MisfireDrills.prx) although they are modelled as distinct networks as shown in Figure 3-16. Only the immediate action (IA) portion of these drills has been modelled as secondary actions are not readily supported by the ICGS LAV vehicle simulation; secondary actions primarily involve physical inspection of the weapons to remediate faults. The CC has the option of using the battle override to bypass minor faults that prevent the weapon from firing, but this option may only be applied in combat and it is only suitable for faults within the electronic, weapon control system.

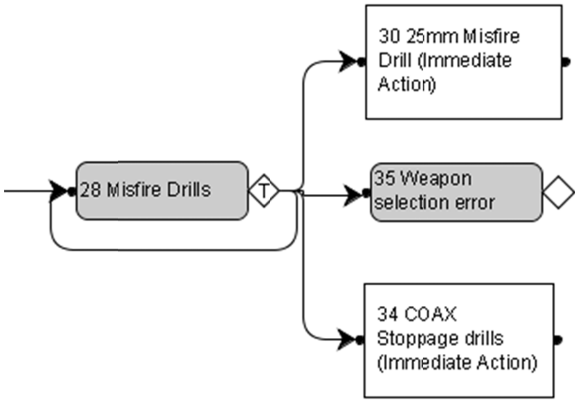

30 25mm Misfire Drill (Immediate Action)
28 Misfire Drills
T
35 Weapon selection error
34 COAX Stoppage drills (Immediate Action)

Figure 3-16: Misfire and Stoppage Drill Top Level Networks

The COAX MG is the simpler of the two weapons, although as both weapons are controlled through the same electronic control system, some of the drill complexity is also shared. The COAX MG Stoppage IA Drill is shown in Figure 3-17. Much of the Stoppage Drill is a physical inspection and correction. Inspection of the CDA panel for indicator lights corresponding to specific control system faults is common for both weapons; the procedure is shown in Figure 3-18. The GNR initiates the Stoppage Drill when the COAX MG fails to fire by reporting “STOPPAGE”; after correcting the fault, the CC orders the GNR to resume the engagement by ordering “READY” or “RESET”, depending on the corrective action taken.

The 25mm cannon Misfire Drill (shown in Figure 3-19) is more involved than the Stoppage Drill, both for the CC and the GNR. There are several possible fault causes that can be rectified by performing the Misfire Drill; however, if the weapon fails to fire after two sequential misfire and correction steps, the CC is required to order “turret make safe” and disengage from combat to perform more detailed fault correction procedures, possibly with the aid of a weapon system technician. The GNR initiates the Misfire Drill when the gun fails to fire by reporting “MISFIRE”; the CC immediately orders “MISFIRE WAIT” to stop the engagement and begin to correct the fault. The CC orders the GNR to resume the engagement after correcting the fault by ordering “READY” or “RESET”, depending on the corrective action taken.

There are two main problem-solving branches for the CC in the Misfire Drill that depend on the state of the bolt position indicator (BPI). If the gun misfires when the BPI is in the sear position, which is the normal firing state, the CC checks to ensure the gun safety is not engaged then inspects the CDA for fault indications (Figure 3-20). The other problem-solving branch occurs when the BPI is in misfire, which involves greater interaction with the GNR to resolve the problem (Figure 3-21). The GNR’s Misfire Drill model (Figure 3-22) involves complicated branching, which is difficult to interpret without investigating the underlying branching logic. For such complicated networks, it is easier to verify the logic by running the simulation and inspecting the resulting operation sequence or creating operation sequence diagrams.

3.1.8 Start Mode Drill

The Start Mode drill is performed to put switches on both power controllers (i.e., weapon control yokes) and LRF displays in a default configuration, engaging the electronic safety, selecting the FIRST (i.e., closest) LRF display option, a wide FOV for the sights, the main gun (i.e., 25mm cannon), and a firing rate of 200 rounds/minute. This is the state of the weapon system at the end of the Action Drill, and the Start Mode drill is performed at the termination of every engagement by both the CC and the GNR. The Start Mode task networks are shown in Figure 3-23.

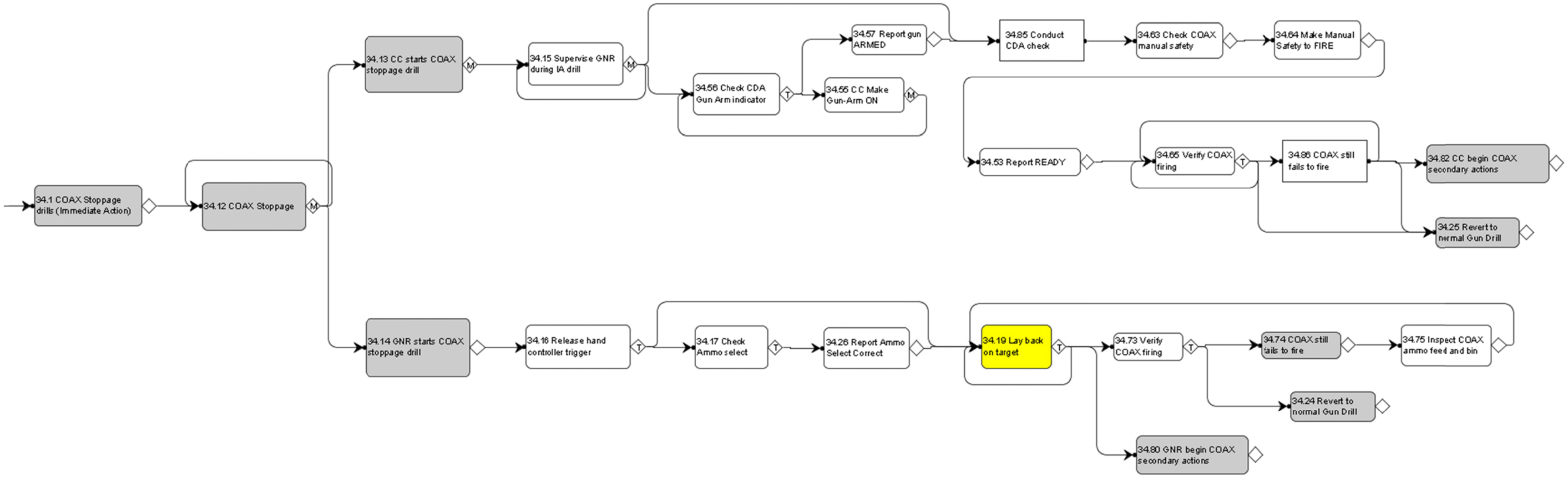

34.1 COAX Stoppage drills (Immediate Action)
34.12 COAX Stoppage
M
34.13 CC starts COAX stoppage drill
M
34.15 Supervise GNR during IA drill
M
34.56 Check CDA Gun Arm indicator
T
34.57 Report gun ARMED
34.55 CC Make Gun-Arm ON
M
34.85 Conduct CDA check
34.63 Check COAX manual safety
34.64 Make Manual Safety to FIRE
34.53 Report READY
34.65 Verify COAX firing
T
34.86 COAX still fails to fire
34.82 CC begin COAX secondary actions
34.25 Revert to normal Gun Drill
34.14 GNR starts COAX stoppage drill
34.16 Release hand controller trigger
T
34.17 Check Ammo select
T
34.26 Report Ammo Select Correct
34.19 Lay back on target
T
34.73 Verify COAX firing
T
34.74 COAX still fails to fire
34.75 Inspect COAX ammo feed and bin
34.24 Revert to normal Gun Drill
34.80 GNR begin COAX secondary actions

Figure 3-17: COAX MG Stoppage Immediate Action Drill

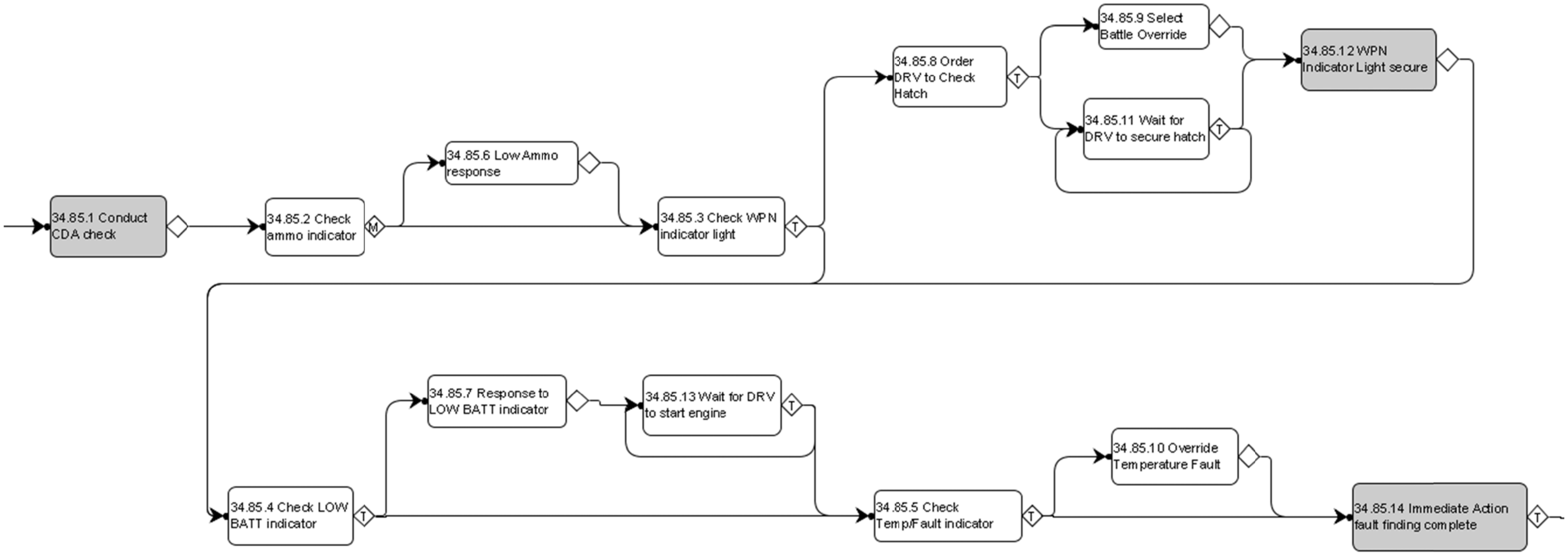

34.85.1 Conduct CDA check
34.85.2 Check ammo indicator
M
34.85.6 Low Ammo response
34.85.3 Check WPN indicator light
T
34.85.8 Order DRV to Check Hatch
T
34.85.9 Select Battle Override
34.85.11 Wait for DRV to secure hatch
T
34.85.12 WPN Indicator Light secure
34.85.4 Check LOW BATT indicator
T
34.85.7 Response to LOW BATT indicator
34.85.13 Wait for DRV to start engine
T
34.85.5 Check Temp/Fault indicator
T
34.85.10 Override Temperature Fault
34.85.14 Immediate Action fault finding complete
T

Figure 3-18: Checking CDA Panel for Indications of Weapon Control System Faults

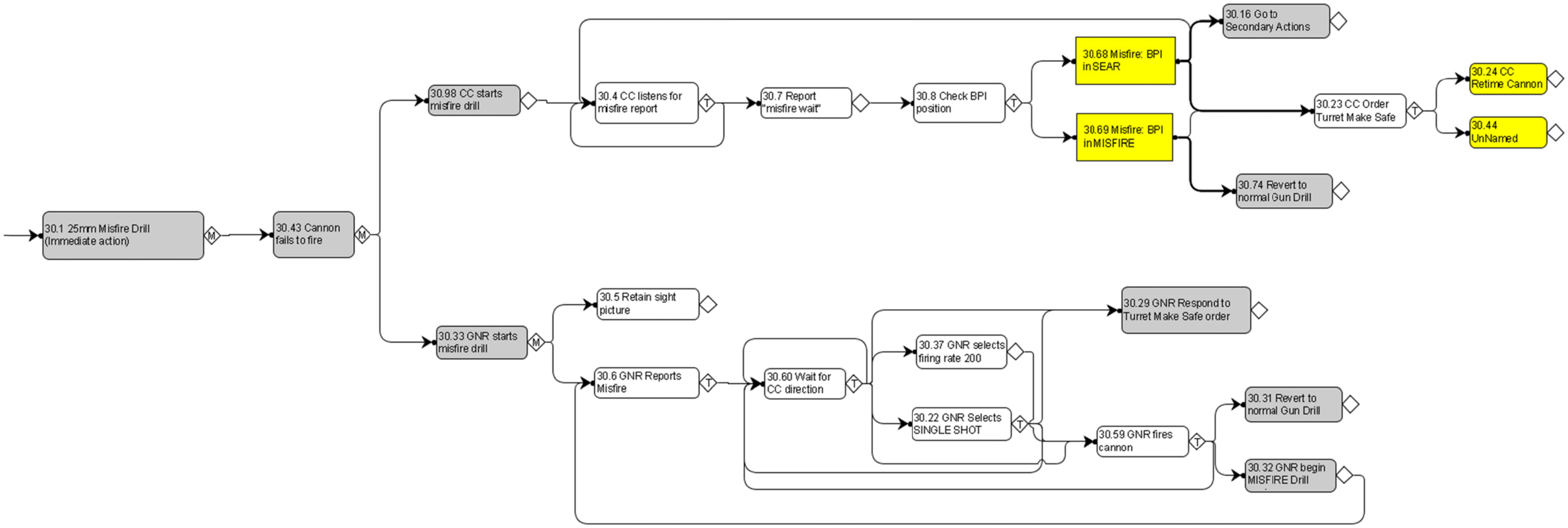

30.1 25mm Misfire Drill (Immediate action)
M
30.43 Cannon fails to fire
M
30.98 CC starts misfire drill
30.4 CC listens for misfire report
T
30.7 Report "misfire wait"
30.8 Check BPI position
T
30.68 Misfire: BPI in SEAR
30.69 Misfire: BPI in MISFIRE
30.16 Go to Secondary Actions
30.23 CC Order Turret Make Safe
T
30.24 CC Retime Cannon
30.44 UnNamed
30.74 Revert to normal Gun Drill
30.33 GNR starts misfire drill
M
30.5 Retain sight picture
30.6 GNR Reports Misfire
T
30.60 Wait for CC direction
T
30.37 GNR selects firing rate 200
30.22 GNR Selects SINGLE SHOT
T
30.29 GNR Respond to Turret Make Safe order
30.59 GNR fires cannon
T
30.31 Revert to normal Gun Drill
30.32 GNR begin MISFIRE Drill

Figure 3-19: Misfire Drill for the 25mm Cannon

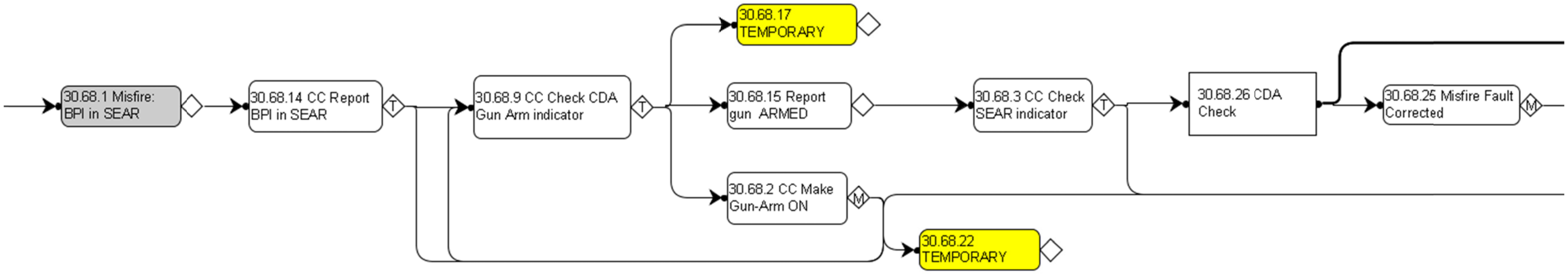


Figure 3-20: CC Misfire Drill Problem Solving when BPI is in Sear Position

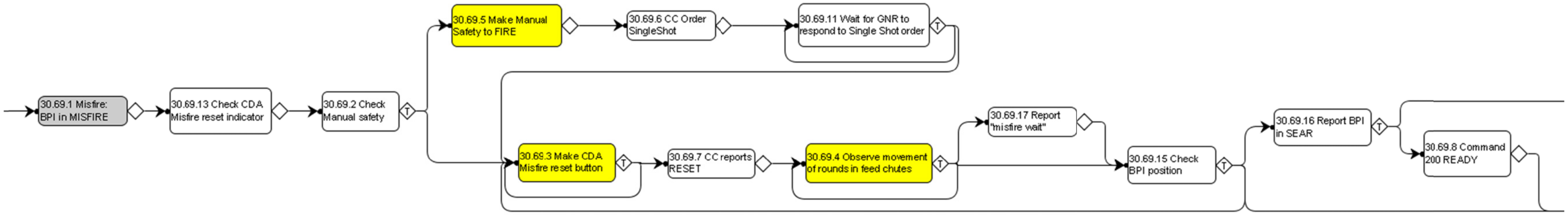


Figure 3-21: CC Misfire Drill Problem Solving when BPI is in Misfire Position

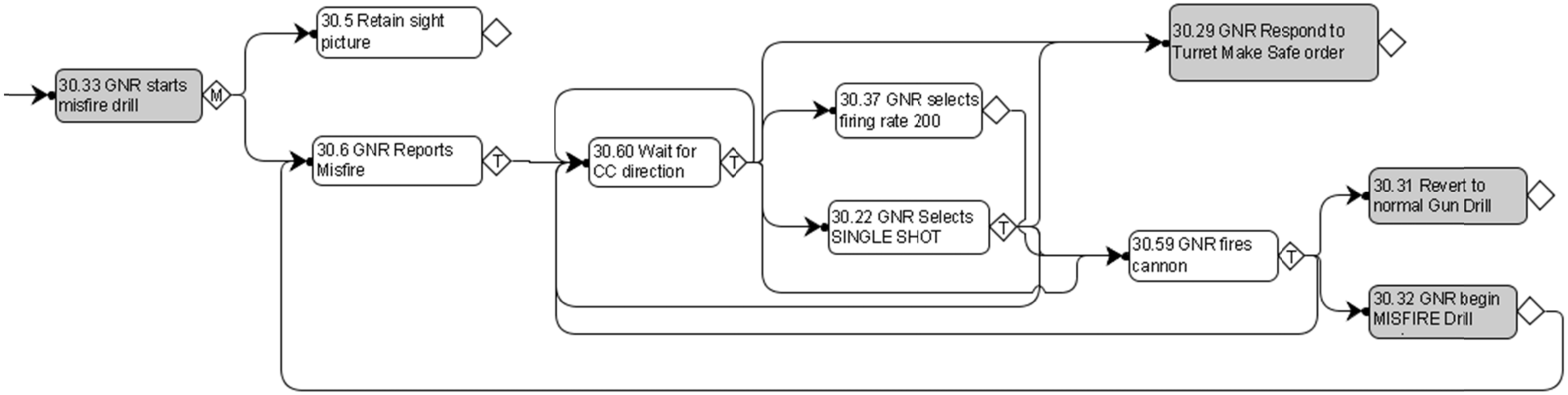

30.33 GNR starts misfire drill
M
30.5 Retain sight picture
30.6 GNR Reports Misfire
T
30.60 Wait for CC direction
T
30.37 GNR selects firing rate 200
30.22 GNR Selects SINGLE SHOT
T
30.29 GNR Respond to Turret Make Safe order
30.59 GNR fires cannon
T
30.31 Revert to normal Gun Drill
30.32 GNR begin MISFIRE Drill

Figure 3-22: Enlarged Image of the GNR's Misfire Drill Task Network from Figure 3-19.

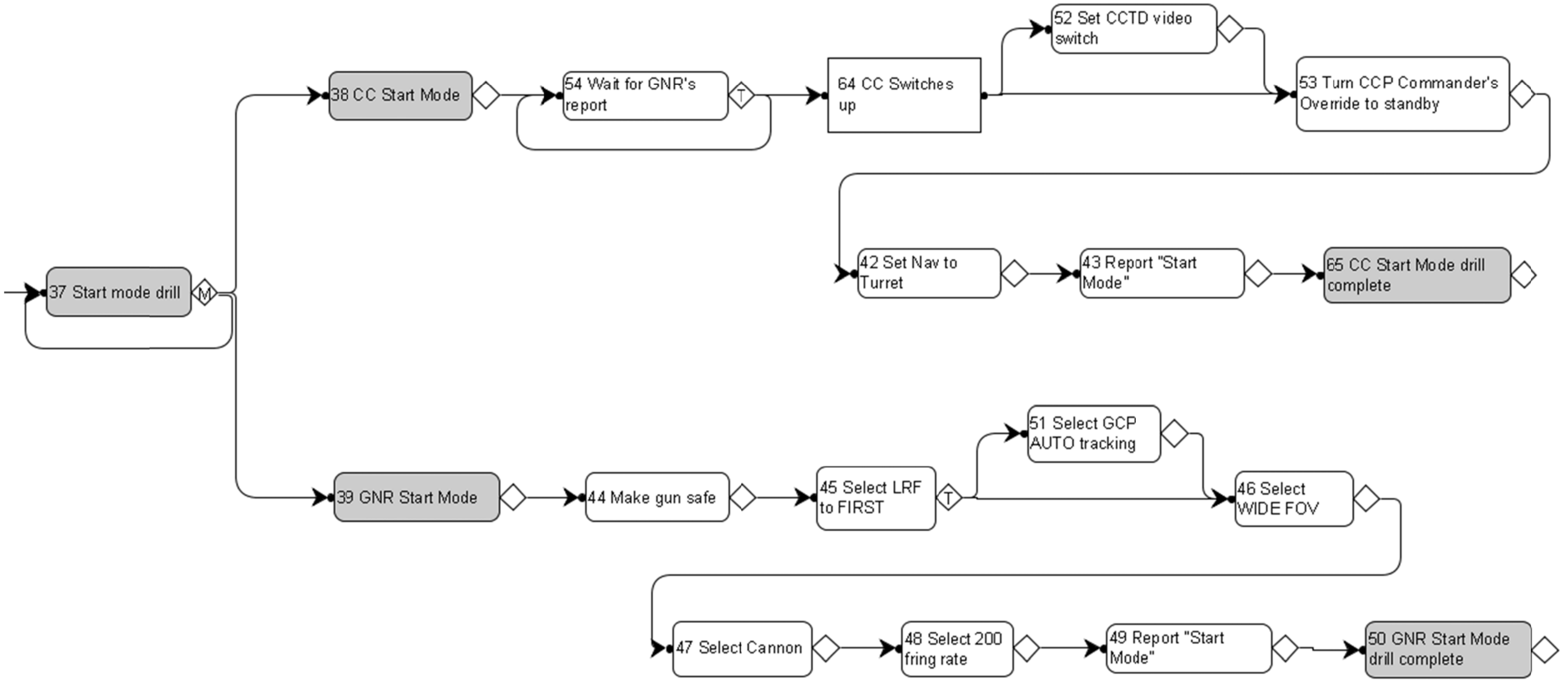


Figure 3-23: Start Mode Drill Task Networks

3.1.9 Turret Make Safe Drill

The CC will order the Turret Make Safe Drill to ensure that weapons cannot be fired or the turret moved accidentally. This drill is typically part of a Misfire or Stoppage Drill, but it may be ordered at any time to ensure safety. The GNR's turret make safe tasks differ slightly between the two LAV variants, but the outcomes are very similar. The Turret Make Safe Drill task network model is shown in Figure 3-24.

3.1.10 Turret Power Down Procedure

The Turret Power Down Procedure is performed at the end of the operational day as part of the Close Down Procedure, or at any time when it is necessary, to prevent the turret and weapon system from operating (e.g., personnel will be on the top of the LAV hull for resupply, driving with hatches up). The Turret Power Down procedure model is stored in the IPME project "TurretPowerUpDown.prx" file and the task network model is shown in Figure 3-25.

3.1.11 Close Down Procedure

The Close Down Procedure is performed immediately after the Turret Power Down Procedure when the LAV crew disembarks the vehicle for an extended period and the LAV will be unattended. The Close Down Procedure secures the LAV physically from unauthorized entry. The task network model is shown in Figure 3-26.

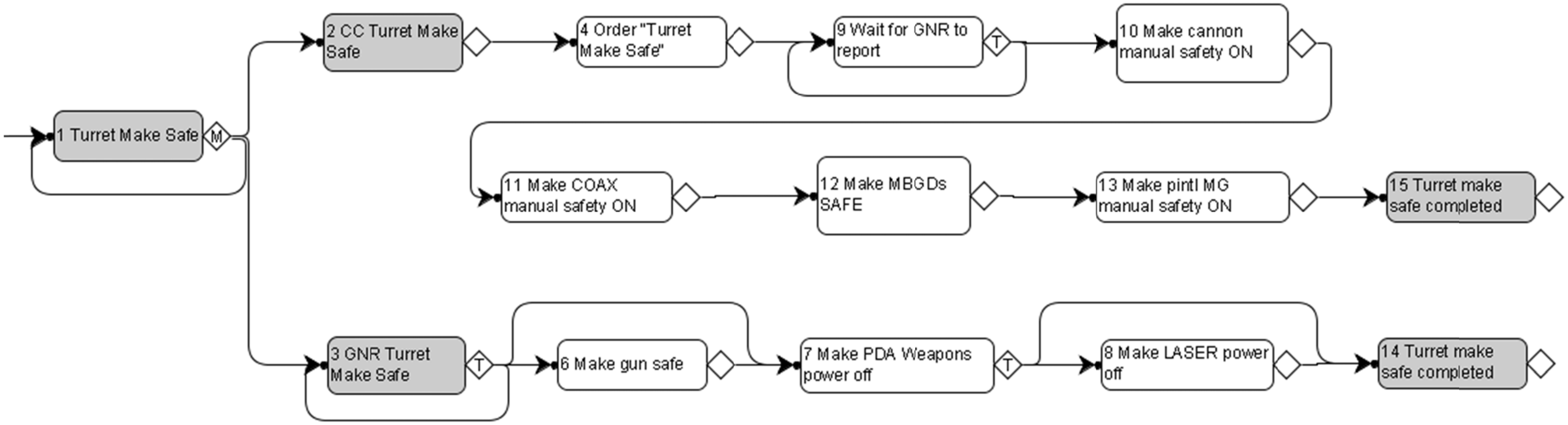

1 Turret Make Safe
M
2 CC Turret Make Safe
4 Order "Turret Make Safe"
9 Wait for GNR to report
T
10 Make cannon manual safety ON
11 Make COAX manual safety ON
12 Make MBGDs SAFE
13 Make pintl MG manual safety ON
15 Turret make safe completed
3 GNR Turret Make Safe
T
6 Make gun safe
7 Make PDA Weapons power off
T
8 Make LASER power off
14 Turret make safe completed

Figure 3-24: Turret Make Safe Task Network Model

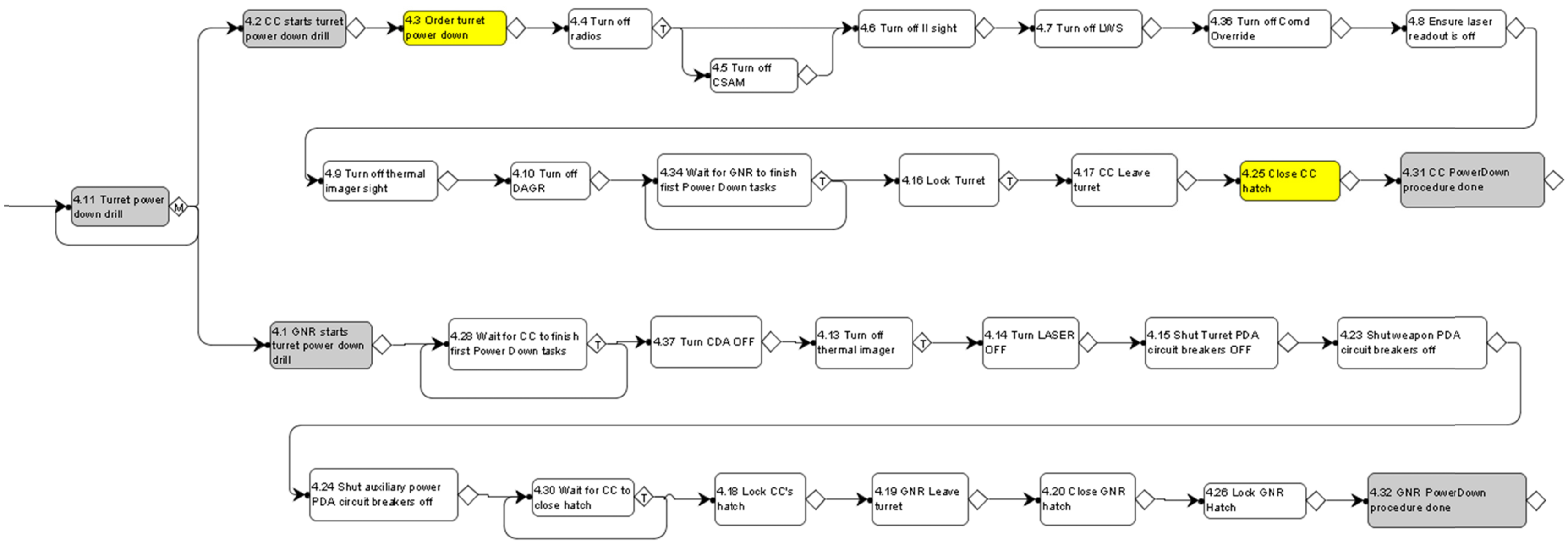

4.11 Turret power down drill
M
4.2 CC starts turret power down drill
4.3 Order turret power down
4.4 Turn off radios
T
4.5 Turn off CSAM
4.6 Turn off II sight
4.7 Turn off LWS
4.36 Turn off Comd Override
4.8 Ensure laser readout is off
4.9 Turn off thermal imager sight
4.10 Turn off DAGR
4.34 Wait for GNR to finish first Power Down tasks
T
4.16 Lock Turret
T
4.17 CC Leave turret
4.25 Close CC hatch
4.31 CC PowerDown procedure done
4.1 GNR starts turret power down drill
4.28 Wait for CC to finish first Power Down tasks
T
4.37 Turn CDA OFF
4.13 Turn off thermal imager
T
4.14 Turn LASER OFF
4.15 Shut Turret PDA circuit breakers OFF
4.23 Shutweapon PDA circuit breakers off
4.24 Shut auxiliary power PDA circuit breakers off
4.30 Wait for CC to close hatch
T
4.18 Lock CC's hatch
4.19 GNR Leave turret
4.20 Close GNR hatch
4.26 Lock GNR Hatch
4.32 GNR PowerDown procedure done

Figure 3-25: Turret Power Down Procedure Task Network Model

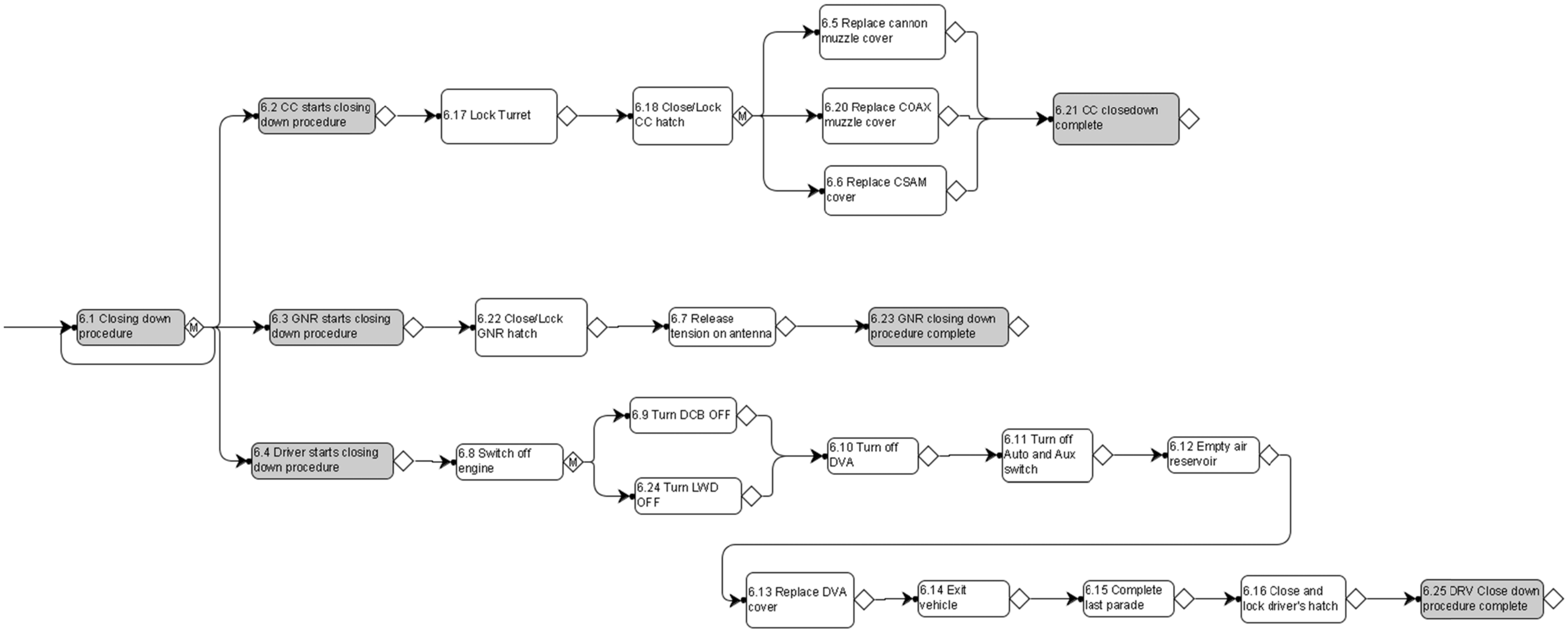

6.1 Closing down procedure
M
6.2 CC starts closing down procedure
6.17 Lock Turret
6.18 Close/Lock CC hatch
M
6.5 Replace cannon muzzle cover
6.20 Replace COAX muzzle cover
6.6 Replace CSAM cover
6.21 CC closedown complete
6.3 GNR starts closing down procedure
6.22 Close/Lock GNR hatch
6.7 Release tension on antenna
6.23 GNR closing down procedure complete
6.4 Driver starts closing down procedure
6.8 Switch off engine
M
6.9 Turn DCB OFF
6.24 Turn LWD OFF
6.10 Turn off DVA
6.11 Turn off Auto and Aux switch
6.12 Empty air reservoir
6.13 Replace DVA cover
6.14 Exit vehicle
6.15 Complete last parade
6.16 Close and lock driver's hatch
6.25 DRV Close down procedure complete

Figure 3-26: Close Down Procedure Task Network Model

3.2 Model Verification

The preliminary verification of each drill in the component model execution was confirmed prior to development of the speech recognition system, simulating both the GNR and the CC as HBR operators. The HBR CC operator model uses the same variables as the speech recognition system for the CC. The IPME execution dialogue "Show Watch Variables" was used both to assign situational context variable values according to the procedures listed in Appendix A as well as to observe the resulting changes to dependent variables. The models were executed and the resulting values of the TextVoice and TextMessage variables of the GNR and CC were compared to the checklists in the aide memoires (DND, 2015, 2016a).

Execution of the models is faster than real time unless the simulation is constrained to run in real time. If the models are executed without synchronizing the simulation to a real-time clock, the IPME Animate Execution in the Execution dialogue should be enabled and the animation speed slider set to between one-half and three-fourths full speed. This allows the user to observe task execution animations and the Watch Variables list while maintaining a reasonable execution speed. The slider may be adjusted during execution to speed up or slow down the simulation as desired.

Each component models executed as intended and replicated the sequence of events in the aide memoires. In most cases, this is straight forward; however, the network representing the engagement process is dynamic, depending on external events such as the appearance or destruction of a threat. The engagement contextual variables were adjusted manually during execution to simulate information that would be provided by the ICGS and the model performed in accordance with the aide memoire engagement drills. Differences between the LAV III and LAV 6.0 variant drills were demonstrated by the model.

There was insufficient time in the contract to conduct extensive testing of the component models or the composite model with the speech recognition system for the CC expressions. Instead, a simple IPME model was used with the speech recognition system to demonstrate that CC expressions resulted in the desired assignments to the CC variables. The results of this testing are provided in Section E.11.

The speech recognition system successfully recognized 45 of the 47 CC expressions, correctly assigning values to the CC variables that interact with the GNR network. Further tests with selected component models indicated the models and speech recognition functioned as intended.

Of the two failures noted in Section E.11, only the CC_FO_Range variable was considered critical as this plays a key role in the CC's instructions to the GNR during engagements. The failure to recognize the verbal range specification prevented testing of the engagement procedures model; however, based on the other tests, this model is expected to run properly once the speech recognition problem with the CC_FO_Range variable is resolved.

The other recognition failure was with the variable CC_ORDER_Relase, which is associated with the CC's order to relase (i.e., fire the laser at the target again). The expression "relase" is not a normal English word, which may be the cause of the lack of recognition. Several phonetic variations of “relase” were attempted without success. Further development work is required to resolve this recognition failure.

3.3 Speech Recognition Implementation

The speech recognition capability leveraged previous work conducted for the Leopard 2 model (Dubreuil, 2015). Once the model neared completion, an XML grammar data file (GunnerCommands.grxml) was developed with LAV-specific expressions for the CC. A second, definition file with the relationships amongst the speech recognition variables and the associated IPME LAV Crew model variables (HBR2Plugin.dll.config) was also developed. An additional grammar data file that recognizes numbers (Number.grxml) was adopted from the VLeo2 project, however, number recognition for range information did not function as intended and further development is required in this area.

The HBR2Plugin.dll is loaded by IPME when the model is executed, and this Dynamic Link Library (DLL) loaded several additional files (e.g., the grammar data files) to interpret speech and assign appropriate model variable values corresponding to the CC expression. Currently, all three files must reside in the IPME CONSOLE directory although the location of the grammar files may be modified provided path information to the new location is added to the HBR2Plugin.dll.config file in the <Grammars> tag definition. Listings of each of the three speech recognition files are provided in Appendix C.

IPME will only use the speech recognition files (i.e., HBR2Plugin.dll.config, GunnerCommands.grxml, and Number.grxml) if the IPME task network model contains a boolean variable “UseHBR2Plugin” that has a “true” value assignment.

It is important for the user to avoid speaking too loudly as this will saturate the microphone resulting in clipping of the analogue speech signal, which will in turn lead to poor signal detection and poor recognition. Reducing the gain on the microphone using the Microsoft audio controls may be sufficient; however, positioning the microphone away from or below the user’s mouth may be necessary to keep sound pressure levels within a usable range. Also, tests with the speech recognition system indicated that individual phrases spoken in quick succession resulted in only the first expression being recognized. This may present a problem, forcing the user to speak more slowly than natural or common for the crew, particularly with well rehearsed, successive phrases. Additional development is required to allow for spoken variability of CC phrases.

3.3.1 Modifying HBR2Plugin.dll.config

This section describes the HBR2Plugin.dll.config file (Section C.1.1) that defines the mapping of the speech recognition variables involved and selects the grammar files to use. Hierarchical indenting is used to improve readability, keeping related lines of code together at a common indent level.

The <MapperConfig> setting in the HBR2Plugin.dll.config file contains the associations between each of the speech recognition grammar variables and the corresponding IPME variable. In the example below, CC_Test_Action is the speech recognition variable (InVariable) used inside of the grammar list while the CC_Test_Value is the variable in IPME (IPMEVariable), which must be defined in the IPME model with the same name and variable type to transfer values to the model. An example is shown as follows:

```
<MapperConfig>
    <Incoming>
      <InMap InVariable="CC_Test_Action" IPMEVariable="CC_Test_Value"
Type="bool" />
    </Incoming>
  </MapperConfig>
```

The <SpeechCommandConfig> setting defines the grammar file to be used. In the example shown below, the only grammar file loaded is named GunnerAndLoaderCommands.grxml; a full or partial path can be specified to locate the grammar file. For example, the text can be replaced with “C:\GunnerAndLoaderCommands.grxml”, which will cause the speech recognition software to search for the grammar file in the root directory of the C-Drive. Currently, all speech files are located in the IPME console directory so only relative path information is required.

```
<SpeechCommandConfig  Enable="true" LangLocale="en-US">
    <Grammars>
      <Grammar>GunnerAndLoaderCommands.grxml</Grammar>
    </Grammars>
  </SpeechCommandConfig>
```

3.3.2 Modifying Grammar Files

There are two “rule” sections in the GunnerCommands.grxml grammar file that require modification to add or edit spoken phrases that are to be recognized. The first <rule> tag near the top of the file contains the definition of the communication link to IPME variables through definitions in the HBR2Plugin.dll.config file. The second <rule> tag contains the associated verbal expressions to be recognized. A few examples will be given to describe the different rules such as types or numbers.

3.3.2.1 First Grammar Rule

The first grammar rule example is shown below:

```
  <rule id="GunnerAndLoaderCommands" scope="public">
    <one-of>      <item>
        <ruleref uri="#TestAction" />
        <tag> $.Command = 'GunLoader.TestAction'; </tag>
        <tag> $.CC_Test_Test = 'true'; </tag>
      </item>
    </one-of>
  </rule>
```

<ruleref uri=”#TestAction’/> defines the grammar keyword if the corresponding utterance was recognized. This keyword corresponds to the resulting actions that will be described in the second rule section, which is described below in Section 3.3.2.2. The first tag, <tag> $.Command = ‘GunLoader.TestAction’; </tag>, labels the command, but ‘GunLoader.TestAction’ only describes the action and does not cause any action to occur. The second tag, <tag> $.CC_Test_Test = 'true'; </tag>, changes the local variable value that is linked with the value in the HBR2Plugin.dll.config (InVariable).

New entries can be created by modifying a copy of the example and inserting it below the first item in the file, similar to the following:

```
<rule id="GunnerAndLoaderCommands" scope="public">
  <one-of>
    <item>
      <ruleref uri="#TestAction" />
      <tag> $.Command = 'GunLoader.TestAction'; </tag>
      <tag> $.CC_Test_Test = 'true'; </tag>
    </item>
    <item>
      <ruleref uri="#AnotherAction" />
      <tag> $.Command = 'GunLoader.AnotherAction'; </tag>
      <tag> $.CC_Other_Test = 'true'; </tag>
    </item>
  </one-of>
</rule>
```

3.3.2.2 Second Grammar Rule

The next set of grammar rule tags describe the spoken expressions to be recognized. For example, the following rule will recognize the verbal expression “test action”:

```
<rule id="TestAction" scope="private">
  <example>TEST ACTION</example>
  <item> Test </item>
  <item> action </item>
</rule>
```

The rule id is the link with the ruleref uri as shown in the section First Rule, keeping the same name but omitting the # symbol. The example field shows how the sentence should be spoken, using the words defined individually in the subsequent “item” list. Alternatively, the expression can be listed as a single “item” statement (e.g., <item> test action </item>) and the “example” tag may be omitted. The example below shows how a grammar phrase can be added.

```
<rule id="AnotherAction" scope="private">
  <example>ANOTHER ACTION</example>
```

```
    <item> another </item>
    <item> action </item>
  </rule>
```

3.3.2.3 Type Rule

For cases where different parameters can be recognized and assigned to the same IPME variable, “types” can be helpful to indicate the parameters without having to repeat the rule. These rules are defined in the “Second Grammar Rule” section of the file. The following rule example shows how one of several different ammunition types may be recognized and assigned to the IPME variable “AmmoType”; it also demonstrates using phonetic variations to assist recognition:

```
<rule id="AmmoType" scope="private">
   <one-of>
      <item> sabot <tag> $='SABOT'; </tag></item>
      <item> saw bow <tag> $='SABOT'; </tag></item>
      <item> h e <tag> $='HE'; </tag></item>
      <item> frang <tag> $='FRANG'; </tag></item>
      <item> frange <tag> $='FRANG'; </tag></item>
      <item> coax <tag> $='COAX'; </tag></item>
      <item> co axe <tag> $='COAX'; </tag></item>
   </one-of>
</rule>
```

Creative spelling and phonetic alternatives may circumvent limitations in the speech recognition software in instances where unusual words are used (e.g., Sabot, Frang, COAX). For this example, the speech will attempt to recognize “saw bow” and if it recognizes that expression, it will return the value “SABOT”. The “tag” contained within the item tag is not required but can be used to indicate what the return value should be. The example below shows how the rule “AmmoType” can be used:

```
<rule id="TestAmmo" scope="private">
   <example> Set ammo to sabot</example>
   <item> Set </item>
   <item> Ammo </item>
   <item> to </item>
   <item>
      <ruleref uri="#AmmoType" />
      <tag> $.MyAmmoVariable = $$; </tag>
   </item>
</rule>
```

The “MyAmmoVariable” is the local speech recognition variable that will contain the tag value passed in as noted with $$. In this case, the “First Grammar Rule” will look like this:

```
<item>
   <ruleref uri="#TestAmmo" />
   <tag> $.Command = 'CC.AmmoTest'; </tag>
   <tag> $.CC_Ammo = $$.MyAmmoVariable; </tag>
</item>
```

The CC_Ammo local variable will be equivalent to the MyAmmoVariable and could be used to pass the value to IPME.

3.3.2.4 Number Grammar Rule

Spoken numbers can be recognized in expressions but the options are limited to formats defined in the Number.grxml grammar file (Section C.1.3). For example, the number 20 in the following expression would be recognized as a word, but can also take on a numerical value for use in computational expressions (the example is declared at the “Second Rule”):

```
<rule id="AimLeft" scope="private">
   <example>Aim Left 20 degrees</example>
   <item> Aim </item>
   <item> Left </item>
   <ruleref uri="Number.grxml#DIGITS" />
   <tag> $.Degrees = $DIGITS.Number; </tag>
   <item> Degrees </item>
</rule>
```

The speech recognition software refers to the Number.grxml file to interpret the number and the value is passed into the “.Degrees” local variable.

NOTE: This variable name can be user defined.

The “$DIGITS.Number” must be used exactly as written and this variable contains the number recognized in the Number.grxml grammar file. The user may then assign the numerical value to an expression that can be used in an IPME computation using a rule defined in the “First Grammar Rule” section as follows:

```
<item>
   <ruleref uri="#AimLeft" />
   <tag> $.Command = 'CC.AimLeft'; </tag>
   <tag> $.CC_AimLeft = '1'; </tag>
   <tag> $.CC_Number = $$.Degrees; </tag>
</item>
```

In this example, CC_AimLeft will be assigned an integer value of 1 and CC_Number will contain the number stored locally in the Degrees variable. If there are multiple variables being set in the

“First Rule” section, then the declaration of the variable in the HBR2Plugin.dll.config will need to be set as follows:

```
<IncomingMulti>
  <InMultiMap>
    <InVars>
      <Variable InVariable="CC_AimLeft" />
      <Variable InVariable="CC_Number" />
    </InVars>
    <OutVars>
    <Variable IPMEVariable="AimLeft" IPMEVariableValue="$.CC_AimLeft"
Type="int"/>
     <Variable IPMEVariable="TestNumber" IPMEVariableValue="$.CC_Number"
Type="int"/>
    </OutVars>
  </InMultiMap>
</IncomingMulti>
```

The <IncomingMulti> tag should be declared after the </Incoming> end tag. The <InVars> section determines the local variables that are passed from the grammar file while the <OutVars> secition contains the IPME variable names.

3.4 Speech Production

The speech production (synthesis) implementation used for this project uses the AT&T Natural Voices voice files “mike” (GNR), “crystal” (CC), and “ray” (DRV). The AT&T speech software was selected in an attempt to improve on the quality of the speech synthesis; however, little if any improvement beyond the VLeo2 project results was observed.

The speech production strategy of the Leopard 2 model required creating an expression file with an index corresponding to the desired phrase associated with a task. This index was passed from IPME and the text rendered into speech using the MS SAPI. This approach was modified in two ways.

First, as noted, the AT&T Natural Voices software was used instead of using the MS SAPI directly. Second, the interface implementation of the text to speech synthesis was simplified, allowing the IPME model to be modified more conveniently.

Two IPME string variables were defined to denote the voice file to be used and the text to be spoken: TextVoice and TextMessage. These variables are also defined in the HBR2Plugin.dll.config and they are used to pass the assigned values to the AT&T Natural Voices speech production software. Sample assignments are as follows:

- TextVoice = “mike”; and
- TextMessage = “This is the message that will be spoken”.

These variables are assigned values within the associated task definition, making use of an IPME user-defined function, SpeechProduction(Operator, Text). The function may be copied and pasted into new tasks and the only change required is to specify the text to be spoken by the speech production software. The IPME function expression corresponding to the above example is:

- SpeechProduction(Entity.AssignedOp.Name, “This is the message that will be spoken”).

The voice file name associated with the HBR crewmember associated with the task is assigned automatically through the IPME system variable, Entity.AssignedOp.Name. The function selects a predetermined voice file that corresponds to the operator assigned to the task; if no operator or an undefined operator is assigned to a task that makes a call to the function, a runtime error occurs and an error message is displayed. The text voice needs to be spelled the same as the voice file name and the spelling is case sensitive. If no voice file with the selected name exists, no text will be spoken, nor does the software provide an error message. If the modeller wishes to use a different voice file, only the IPME SpeechProduction() function is changed to reflect the assignment to the appropriate HBR crew member operator.

The text message to be spoken must be contained within quotes; alternatively, a string variable may be inserted in the SpeechProduction() function. The message may contain punctuation, although the quality of the interpreted intonation by the AT&T software does not create a natural-sounding expression. The SpeechProduction() function was coded in task Beginning Effects fields so that the phrase would be spoken during the task execution rather than after the task completed in an attempt to compensate for the speech production software latency.

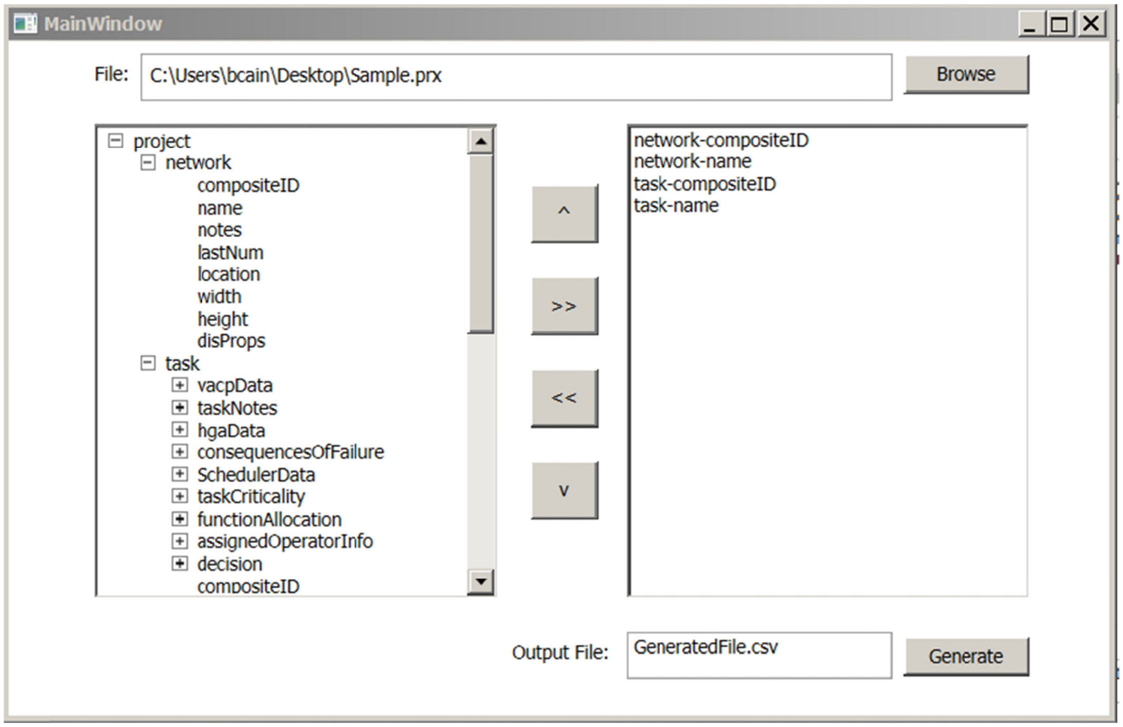
The speech production software requires that the subsequent text message must be different from the preceding message. This approach prevents generating multiple, identical speech messages, for example from spinner task that is waiting for an event to move on to the next task; however, it means that if the modeller wants to synthesize a “reminder” of the “same command”, the spoken message must be changed slightly to be processed. Messages or expressions to be synthesized from IPME are queued and processed in order of presentation.

3.5 IPME HTA Task List Tool

The IPME printing capabilities are limited and do not support traditional human factors engineering task information reporting methods. A preliminary task data extraction utility was developed to support HTA reporting. This tool allows the user to extract task details from the IPME project file (.PRX) and save the results as a CSV file for subsequent analysis in a spreadsheet or reporting (in a table). The HTA export software for the tool will be provided by CAE with the IPME LAV crew model and the executable file is called “DRDCLAVTask.exe”.

The main dialogue window of this tool is shown in Figure 3-27. The dialogue window provides the user with the capability to navigate to and select an IPME project fie that contains the model network. The user can then select specific network and task data elements for extraction. The extracted data may be saved as a CSV text file, as shown in Figure 3-28, that can be imported into a spreadsheet or word processing program, as required. The utility does not change the IPME PRX project file.

The utility was used with each of the component models to create a set of task lists; these lists are provided in Appendix D.



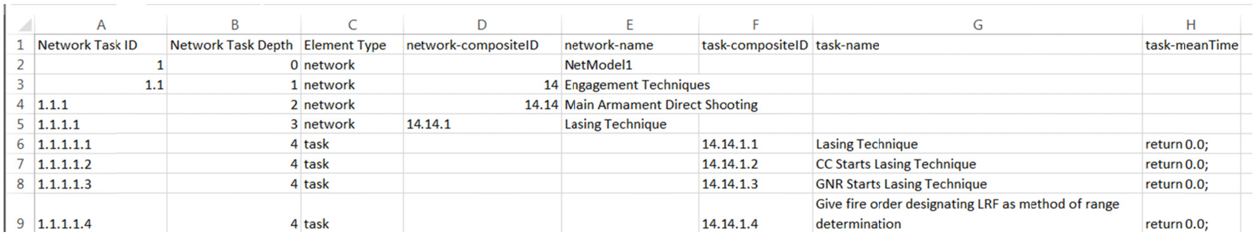
File: C:\Users\bcain\Desktop\Sample.prx

- project
  - network
    - compositeID
    - name
    - notes
    - lastNum
    - location
    - width
    - height
    - disProps
  - task
    - vacpData
    - taskNotes
    - hgaData
    - consequencesOfFailure
    - SchedulerData
    - taskCriticality
    - functionAllocation
    - assignedOperatorInfo
    - decision
    - compositeID

network-compositeID
network-name
task-compositeID
task-name

Output File: GeneratedFile.csv

Figure 3-27: Main Dialogue Window for the HTA Task Information Extraction Tool



	A	B	C	D	E	F	G	H
1	Network Task ID	Network Task Depth	Element Type	network-compositeID	network-name	task-compositeID	task-name	task-meanTime
2	1	0	network		NetModel1			
3	1.1	1	network	14	Engagement Techniques			
4	1.1.1	2	network	14.14	Main Armament Direct Shooting			
5	1.1.1.1	3	network	14.14.1	Lasing Technique			
6	1.1.1.1.1	4	task			14.14.1.1	Lasing Technique	return 0.0;
7	1.1.1.1.2	4	task			14.14.1.2	CC Starts Lasing Technique	return 0.0;
8	1.1.1.1.3	4	task			14.14.1.3	GNR Starts Lasing Technique	return 0.0;
9	1.1.1.1.4	4	task			14.14.1.4	Give fire order designating LRF as method of range determination	return 0.0;

Figure 3-28: Sample Output of Task and Network Information Obtained from IPME HTA Extraction Tool

4 SOFTWARE INSTALLATION

4.1 IPME

The IPME installation file may be obtained from DRDC Toronto, which is the IPME point of contact for the Department of National Defence[6]. This IPME installation file "IPME Setup 4.6.3.1007.msi" is included with the laptop deliverable in the directory Installation\IPME. It is typical and sufficient to accept the default installation options, which will store IPME in the directory "C:\Program Files (x86)\MAAD" in a sub-directory corresponding to the IPME version. The IPME installation application provides all of the necessary files to run IPME on Windows; however, IPME relies on having the runtime version of .NET 4.0 or later.

Once installed, IPME requires a license file to operate. Launching IPME allows the user to generate an information file that can be used to generate a license file using the "Generate Licence Info File…" button from the Help/Licensing… menu item. The information file is computer and user specific, so it should be generated from the user account that will be used for IPME modelling. The resulting electronic IPME information file should be sent to DRDC Toronto, attention Dr. W. Wang[6], who can generate the license file. The license file may be assigned to the user by launching IPME again, navigating to the license location, and selecting the file using the "Locate License File…" button also on the IPME Help/Licensing… menu item.

4.2 Speech Recognition and Production

The Microsoft Speech Platform 11 speech recognition software can be installed by running the setup program "SpeechPlatformRunTime.msi" that is located in the directory Installation\Microsoft Speech Platform 11\. The default options are sufficient to operate with the LAV crew model. Next, install the MSSpeech_SR_en-US_TELE.msi, again accepting all default options.

The speech production software can be installed by running "AT&T runtime.exe" as well as the voice installation files that are included in "Installation\AT&T\". The default options are sufficient for the LAV crew model.

4.3 IPME Speech Plugin

To use the speech capability, copy the files listed below from "Installation\ConfigFiles\" into the IPME directory "C:\Program Files (x86)\MAAD\IPME4_5_7\Console\":

- GunnerCommands.grxml;
- Number.grxml;
- HBR2Plugin.dll;

[6] IPME POC for DND/DRDC is Dr. W. Wang, +1 (416) 635-2000, Wenbi.Wang@drdc-rddc.gc.ca or Wenbi.Wang@forces.gc.ca.

- HBR2Plugin.dll.config;
- IPME.REST.dll;
- IPME.VariableManager.dll;
- SpeechCommandRecognizer.dll; and
- SpeechGenerator.dll.

Before starting the IPME model, it is good practice to restart the simulation engine to ensure that the speech DLL files are loaded properly. Restarting the simulation engine may be performed from the IPME Tools/User Preferences/Execution tab or using Ctrl+Shift+P. Then, click on the Stop Engine button, then click the Restart Engine button. This is necessary to properly load the speech plugin after loading a new model or making changes to the speech files, but it is not required when a model is loaded in a new instance of IPME. Restarting the simulation engine is also required if the microphone is disconnected or connected after IPME has been launched.

The speech software modules rely on an IPME variable named UseHBR2Plugin and this variable must be set as a Boolean “true” value for the speech modules to work.

The microphone also needs to be configured properly before the speech plugin will work. This may be accomplished through the Microsoft Control Panel under Speech Recognition using the “Set up microphone” application. Position the microphone as directed in the setup application.

4.4 Speech Files Source Code

To set up the source code for the speech recognition and production (e.g., in case changes are required), the following environment variables are needed:

- XSDPath = C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools\xsd.exe[7]; and
- IPMEPath = C:\Program Files (x86)\MAAD\IPME4_5_7\Console\:
  - The IPME directory in the path must be the current IPME version.

The “Installation\IPME.Extent\IPME.Extent.sln” file can then be executed to build the project. The configuration files (except the grammar files) are automatically copied to the IPME Console folder if the project is built successfully.

[7] NETFX is provided by https://www.microsoft.com/en-ca/download/details.aspx?id=17718

5 MODEL EXECUTION

Instructions for executing each of the component models may be found in Appendix A; execution of the composite model is performed similarly. Appendix A should be consulted for more detailed explanation of model execution, including variable initialization.

When the composite model is loaded, it may require specification of contextual variables to execute as desired, although default values are set for all relevant variables. As in the component model, execution of the composite model may be started from the IPME model execution interface without immediately executing the LAV crew tasks. A spinner task allows the user to initialize model variables to desired values through the IPME Watch Variables dialogue window, which may be opened from the Display tab at the bottom of the Execution Control dialogue window.

Execution of the LAV crew networks begin once the user sets the IPME “BEGIN” variable to “true”; this may be accomplished in the Watch Variables dialogue window by typing “t” in the BEGIN data value field, or by saying the word “begin” into the speech recognition microphone connected to the computer.

6 COMPUTER REQUIREMENTS

Preliminary, informal testing of the IPME environment with the speech recognition software was conducted on a HP Elitebook laptop computer with an i5 processor and 8 GB of RAM, and an integrated graphics processor unit (GPU). The results of this testing indicated that a laptop solution was feasible for execution in near real time, but to reduce observed software latencies, a multicore, multithreading processor was suggested (e.g., an Intel Core i7-7820HQ). However, the most important factor for timely execution performance appeared to be the amount of system memory available, and it is suggested the laptop be equipped with 16 GB if feasible. In keeping with this observation, it is suggested that the laptop also have a discrete GPU that provides dedicated memory that is separate from system memory (e.g., NVIDIA GeForce GTX 7 to 10 series or AMD Radeon. R5 to R9 series).

Disk storage is not a significant factor; however, as execution will involve disk access, it was suggested that the laptop include a fast hard drive (e.g., a solid state drive (SSD), such as a PCIe x4, NVME). Additional features should include typical advanced office computer characteristics, such as a 1920 x 1080 display, for convenience. IPME officially is intended for use on Windows 7 or various Linux operating systems. For compatibility with other systems, the Windows operating system is recommended; however, Windows 7 is nearing its end of life. Anecdotal reports indicate that IPME runs without problem on Windows 10, so this operating system is suggested provided it is the Professional version. Windows 10 Professional provides access to the BitLocker system encryption software that CAE recommends for laptops containing controlled goods, such as elements of the LAV crew model project deliverables.

The laptop computer should be equipped with a microphone headset suitable for speech recognition, providing good microphone noise reduction, and over-ear headphones, both to reduce extraneous noise and to more closely resemble the headset worn by LAV crew members.

Several options were considered and an “MSI GS43VR PHANTOM PRO” laptop computer[8] with the following characteristics was selected for implementing the LAV crew model:

- Intel Core I7 7700HQ central processing unit (CPU);
- NVIDIA GeForce GTX 1060 6GB GDDR5 GPU;
- 16 GB RAM;
- 128 GB SSD;
- 1 TB hard disk drive (HDD); and
- 14-inch high-definition display with a resolution of 1920 x 1080.

[8] MSI Laptop: https://www.msi.com/Laptop/GS43VR-6RE-Phantom-Pro.html#hero-overview

The “MSI GS43VR PHANTOM PRO” laptop was supplied with Windows 10 Home operating system, which was subsequently upgraded to Windows 10 Professional to provide access to the BitLocker software. The computer hard drive was encrypted using the BitLocker software to better protect the sensitive information that would be stored on the drive.

Decryption information and account credentials will be provided to DRDC separate from the laptop deliverable.

A Plantronics SHR 2083-01 over-the-ear headset[9] was selected to provide speech input and output for the user. This headset incorporated noise cancelling to improve speech recognition input. This headset provides a quick-disconnect plug that connects to the computer through a USB Plantronics quick disconnect adapter (DA80) and digital signal processing device.

[9] Plantronics headset: http://www.plantronics.com/ca/product/shr2083-01

7 RECOMMENDATIONS

The following are recommendations for improvements to the LAV crew model solution developed in the current contract and will assist in the preparation for integration with the ICGS simulator and subsequent study. None of these recommendations seem high risk, although their successful completion may require technical expertise not resident within CAE.

- The speech recognition system used has notable deficiencies that may affect an experimental study. While recognition rates are good for discrete phrases, speaking rapidly results in poor recognition, particularly when discrete phrases are verbalized in rapid succession. Also, there is a noticeable latency in the speech recognition response setting associated values in IPME.
    - CAE recommends exploring alternative speech recognition solutions and engaging speech recognition experts. Tentative discussions with Dr. G. Penn at the University of Toronto suggest that performance improvements may be achieved using custom speech recognition software that exploits more modern techniques than those used in the Microsoft SAPI. Further talks involving both DRDC and CAE should explore how to effectively engage Dr. Penn and his laboratory to enhance the HBR LAV crew speech capabilities.
- The speech production (synthesis) system is notably artificial. While the speech produced is usually intelligible, DRDC has noted that they would prefer a more natural-sounding solution.
    - CAE recommends exploring alternative speech synthesis solutions. Discussions with Dr. Penn also suggest that significant improvements in speech synthesis have been made in recent years that could be readily adapted to the current project. Speech synthesis improvements should be explored with Dr. Penn along with the speech recognition improvements.
- Speech recognition is currently limited to simple phrases that are part of the procedural doctrine; however, interaction amongst the crew often involves natural language communications.
    - CAE recommends that a robust natural language processing capability should be developed to accommodate non-doctrinal expressions that may be required for the training effectiveness study. Examples of non-doctrinal verbal expressions include directions to the DRV during navigation, specification of visual arcs of responsibility for the GNR, etc.

- There are two CC verbal expressions that the speech recognition system development failed to recognize, which have a significant detrimental effect on the model's usefulness.
    - CAE recommends that further development effort should be devoted to correcting the existing speech recognition system in the event that discussions with Dr. Penn fail to identify a feasible way forward for an alternative speech recognition system.
- The LAV crew model reproduces several of the most frequently used drills and procedures; however, not all of the procedures that an operator must know are included in the model.
    - CAE recommends defining the training-effectiveness study scope in sufficient detail to determine which drills and procedures are required for the study before incorporating any additional procedures or elaborating capabilities currently in the LAV Crew model. This study planning could be conducted as a joint DRDC-CAE task during a subsequent callup, building on the approach intended for the VLeo2 study (Cain, 2013).
- The current LAV crew model contains several incomplete tasks that depend significantly upon interaction with the ICGS simulation environment. These tasks do not interfere with the LAV crew model execution in the IPME environment; however, neither do they perform any actions, being simply placeholders for future modelling.
    - CAE recommends that these tasks be elaborated, developing computational expressions will be required to control continuous processes such as target tracking for moving targets or target selection when multiple targets exist, during the integration of the LAV crew model and the ICGS simulation. This strategy will provide the data description for interaction with the ICGS while also providing a verification method for the resulting task executable code.
- The HTA reporting tool significantly reduces the time required to catalogue all operator tasks; however, the utility is rudimentary and substantial manual formatting is required to organize the information into a table suitable for reports (e.g., hierarchical structures and distinguishing network control tasks distinct from operator tasks).
    - CAE recommends further development of the HTA reporting tool to include automated formatting and organizing of the extracted HTA databased on network data to reduce the level of effort for HTA reporting from IPME task analysis representations.

8 CONCLUSION

The development of the LAV crew model has proved challenging and required a greater level of effort than was originally anticipated. The interaction amongst the LAV turret crew is complicated, particularly while operating the weapon system, requiring numerous branching conditions in the model logic to represent the crew procedures accurately.

Some of the verbal expressions required from the speech recognition system, notably uncommon words, lead to recognition failures that adversely affected the model execution. Further, the speech synthesis solution, while intelligible, is decidedly artificial. Each of these developments would benefit from collaboration with experts in speech modelling, a subset of artificial intelligence.

Nevertheless, the IPME representation of the major drills and procedures was largely successful as was its integration with the speech recognition and synthesis software. The LAV crew behavioural models allow the user to simulate either the GNR or the CC although the focus is on the GNR; the CC representation only models the procedural tasks, not more ill-constrained, abstract tasks such as crew supervision, decision making that would require more versatile artificial intelligence representations.

Several recommendations for improving the model have been made that could be incorporated in a subsequent contract, such as during the integration process with the ICGS LAV simulator virtual environment.

9 REFERENCES

Annett, J., & Duncan, K. D. (1967). Task analysis and training design. *Occupational Psychology*, *41*, 211–221.

Annett, J., Duncan, K. D., Stammers, R. B., & Gray, M. J. (1971). *Task Analysis*. London: Her Majesty's Stationery Office.

Annett, J., & Stanton, N. A. (2000). *Task analysis*. London: Taylor & Francis.

Cain, B. (2013). *Validation of a Virtual Leopard 2 virtual reality training device: Proposed assessment approach* (Technical Memorandum No. DRDC Toronto TM 2013-030). (D. R. and D. Canada, Trans.) (p. 45). Toronto: Defence Research and Development Canada.

Cain, B. (2017). *Comparison of Armour Crew drills: Potential for a common crew model* (Contract Report No. W7719–135217/001/TOR). Toronto, ON, Canada: Defence Research and Development Canada.

Cain, B., Magee, L., & Belyavin, A. (2011). Validating an HBR simulation for training within a virtual environment (p. 8). Presented at the Behavioral Representation In Modeling and Simulation (BRIMS 2011), Sundance Resort, Utah.

Cain, B., Magee, L., & Kersten, C. (2011). *Validation of a virtual environment incorporating virtual operators for procedural learning* (No. DRDC Toronto TR 2011-132) (p. 47). Toronto: Defence Research and Development Canada.

DND. (2004). *Theory of armoured gunnery: 25mm Application of fire* (No. B-GL-305-014/FP-001) (p. 69). CTC Gagetown, NB: Chief of the Defence Staff, ARMOUR SCHOOL.

DND. (2014). *DP3 Armour Crew Commander* (Qualification Standard and Training Plan No. A-P3-002-ACC/PG-B01) (p. 325). Department of National Defence.

DND. (2015). *LAV 6.0 Gunnery handbook*. CTC Gagetown, NB: Chief of the Defence Staff, Department of National Defence, Army IG team.

DND. (2016a). *25mm Turret drill manual* (No. B-GL-393-014/FP-001) (p. 82). CTC Gagetown, NB: Chief of the Defence Staff, Department of National Defence, Army IG team.

DND. (2016b). *LAV 6.0 Application of fire* (No. B-GL-305-014/FP-001) (p. 78). CTC Gagetown, NB: Chief of the Defence Staff, Royal Canadian Armour Corps School.

Dubreuil, A. (2015). *Human Behaviour Representation: Task 2 Final Report* (Contract Report No. CAE Document Number 5746-004 (V1)) (p. 36). Toronto, ON, Canada: DRDC.

MacQuarrie, D. (2017). *ICGS (Interim Crew Gunnery System): System user guide* (p. 104). Kingston, ON: Department of National Defence.

MA&D. (2015). Integrated Performance Modelling Environment (IPME) User Guide, Version 4.6. Alion Science and Technology.

Truong, M., Cain, B., & Lawrynczyk, A. (2017). *Strategy for integrating the LAV crew IPME model and the Interim Crew Gunnery System* (Contract Report No. W7719–135217/001/TOR). Toronto, ON, Canada: Defence Research and Development Canada.

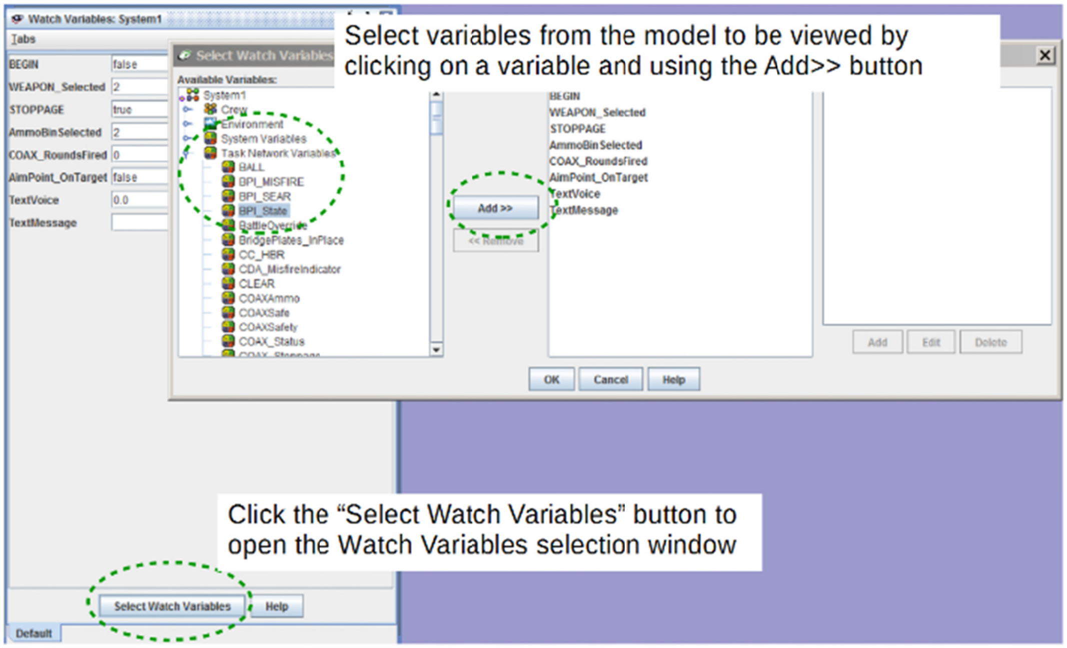
APPENDIX A LAV CREW MODULE EXECUTION

A.1 Setup for Drill Testing

The following is the setup for drill testing:

1. Start the simulation running. The simulation should idle, waiting for the appropriate initiating conditions for the test condition variable values set by the user. Usually the drills wait for the user to set the variable BEGIN to “true” in the Watch Variables dialogue window.

2. Assign the desired test condition variable values in the Watch Variables dialogue window. The window can be enabled from the Display tab in the Execution Control dialogue window. Selection of the variables is performed using the Select Watch Variables button at the bottom of the Watch Variables window, then successively selecting the desired variables (usually from the Task Network Variables category in the system tree of the available variables) as indicated in Figure A-1.



Select variables from the model to be viewed by clicking on a variable and using the Add>> button

Click the “Select Watch Variables” button to open the Watch Variables selection window

Watch Variables: System1

BEGIN	false
WEAPON_Selected	2
STOPPAGE	true
AmmoBinSelected	2
COAX_RoundsFired	0
AimPoint_OnTarget	false
TextVoice	0.0
TextMessage	

Select Watch Variables

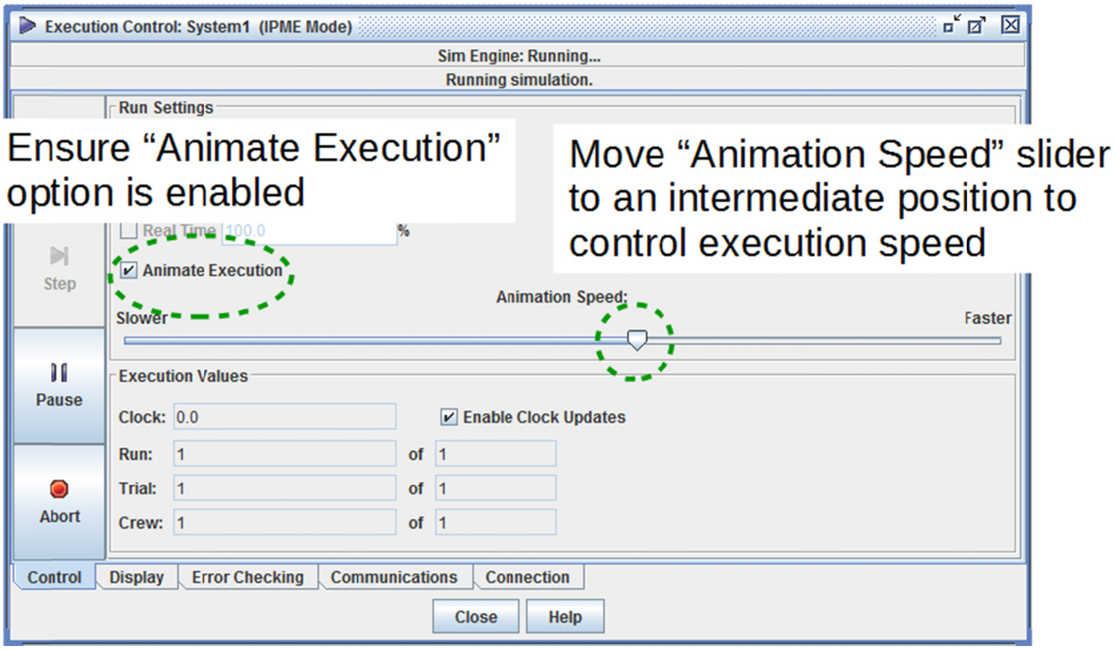
Available Variables:

BEGIN
WEAPON_Selected
STOPPAGE
AmmoBinSelected
COAX_RoundsFired
AimPoint_OnTarget
TextVoice
TextMessage

Figure A-1: Selecting Task Network Variables to be Observed during Execution

3. The variable BEGIN is only used to allow the simulation to continue after the test condition variable values have been set. BEGIN serves no purpose after the simulation continues and should be removed from the networks when they are integrated into a common IPME project and connected to the ICGS simulator.

4. It is recommended to set the execution animation speed to some intermediate value on the Execution Control dialogue window, as shown in Figure A-2. If the animation speed is left at the default setting and the model execution may happen too quickly to observe effectively.



Execution Control: System1 (IPME Mode)

Sim Engine: Running...

Running simulation.

Run Settings

☐ Real Time 100.0 %

☑ Animate Execution

Animation Speed:

Execution Values

Clock: 0.0

☑ Enable Clock Updates

Run: 1 of 1

Trial: 1 of 1

Crew: 1 of 1

Ensure “Animate Execution” option is enabled

Move “Animation Speed” slider to an intermediate position to control execution speed

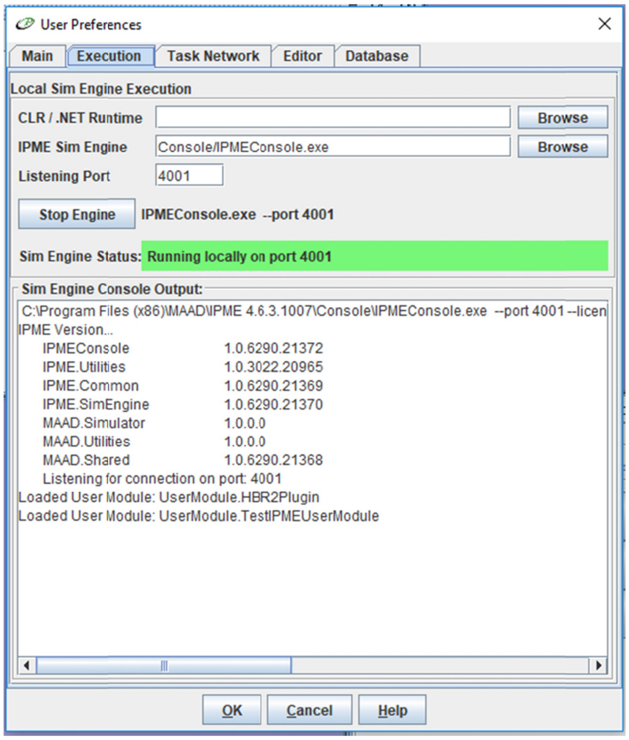
Figure A-2: Execution and Animation Settings for Task Network Testing

A.2 IPME HBR2 Plugin Debugging

Users can determine if the HBR2Plugin is being loaded by examining the IPME Sim Engine Console dialog. To access this dialog in IPME, use the IPME menu command:

1. **Tools->User Preferences->Execution**, or use the keyboard short cut, **Ctrl->Shift->P** to bring up the dialog window (Figure A-3).

The console should list the loaded user modules.



Local Sim Engine Execution

CLR / .NET Runtime:

IPME Sim Engine: Console/IPMEConsole.exe

Listening Port: 4001

IPMEConsole.exe --port 4001

Sim Engine Status: Running locally on port 4001

Sim Engine Console Output:

```
C:\Program Files (x86)\MAAD\IPME 4.6.3.1007\Console\IPMEConsole.exe  --port 4001 --licen
IPME Version...
    IPMEConsole          1.0.6290.21372
    IPME.Utilities       1.0.3022.20965
    IPME.Common          1.0.6290.21369
    IPME.SimEngine       1.0.6290.21370
    MAAD.Simulator       1.0.0.0
    MAAD.Utilities       1.0.0.0
    MAAD.Shared          1.0.6290.21368
    Listening for connection on port: 4001
Loaded User Module: UserModule.HBR2Plugin
Loaded User Module: UserModule.TestIPMEUserModule
```

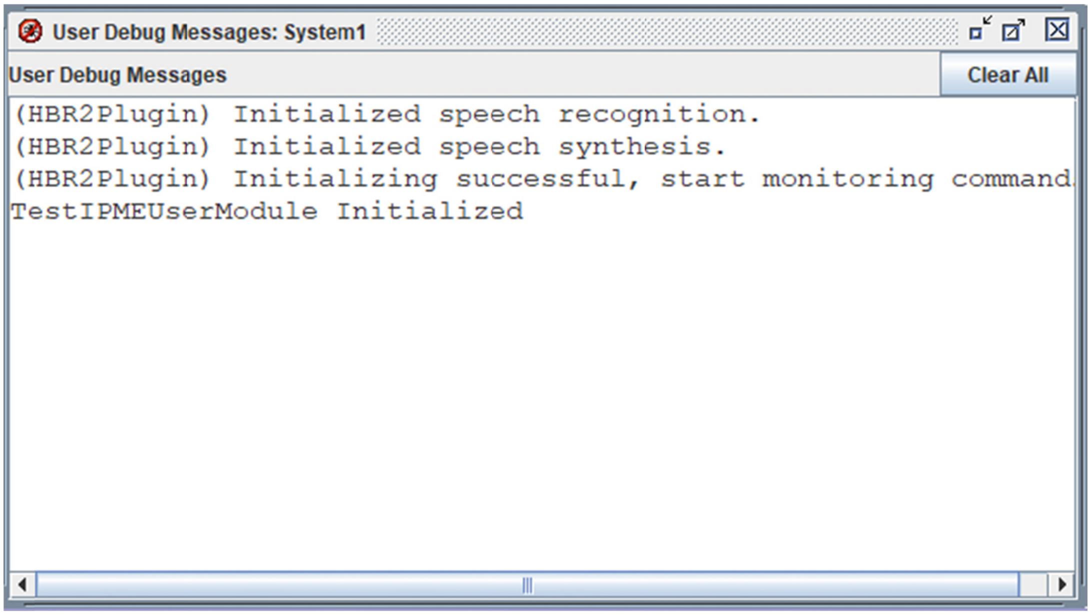
Figure A-3: IPME Sim Engine Console

Debugging statements were added to the plugins that output to the IPME User Debug Message dialog. These debugging statements indicate if the plugins are being initialized correctly. Due to the nature of IPME user module plugins, these debugging statements will only appear when the model is put into run state. The dialog will pop up and display a message if the plugin initializes correctly. This feature is activated by setting DebugEnabled to true in the plugin configuration file. The configuration file is named **HBR2Pligin.dll.config** and located in the following directory: C:\ Program Files (x86)\MAAD\IPME4.6.3.1007\Console\

The following line in the configuration file enables or disables the debugging messages:

```
<DebugEnabled>true</DebugEnabled>
```

Setting the value to false will disable the output of messages. Successful initialization of the plugin should result in the messages shown in Figure A-4 to appear in the IPME User Debug Message dialog window.



```
(HBR2Plugin) Initialized speech recognition.
(HBR2Plugin) Initialized speech synthesis.
(HBR2Plugin) Initializing successful, start monitoring command
TestIPMEUserModule Initialized
```

Figure A-4: User Debug Window

A.3 25mm Misfire Drill, Immediate Action

A.3.1 Misfire Drill, BPI in SEAR: Drill a1/a1A (LAV III) or Ser #1 (LAV 6)

1. BEGIN = f;
2. LAV_Type = 3 (for LAV III; 6 for LAV 6);
3. WEAPON_Selected = 1 (25mm Cannon);
4. MISFIRE = t;
5. BPI_State = 1 (sear);
6. Context = 1 (Simulated range = 0; Simulated combat = 1);
7. CDA_MisfireIndicator = true (No fault condition: false or OFF);
8. LAV_Running = false (needed for Low Battery fault; otherwise can be true); and
9. CannonManualSafety = false (No fault condition).

Select one or more of the following to test the response:

1. GunArmed_IndicatorLight = false (No fault condition: true or ON);

2. SEAR_IndicatorLight = true (No fault condition: false or OFF);

3. GunArmed_IndicatorLight = false (No fault condition: true or ON);

4. Sear_IndicatorLight = true (No fault condition: false or OFF);

5. LowAmmo_IndicatorLight = true (No fault condition: false or OFF);

6. WPN_IndicatorLight = true (No fault condition: false or OFF);

7. LowBattery_IndicatorLight = true (No fault condition: false or OFF); and

8. TempFault_IndicatorLight = true (No fault condition: false or OFF).

Then initiate execution:

1. BEGIN = t.

A.3.2 Misfire Drill, BPI in SEAR, Drill a1B (LAV III) or Ser #2 (LAV 6)

1. BEGIN = f;

2. WEAPON_Selected = 1 (25mm Cannon);

3. MISFIRE = t;

4. BPI_State = 1 (sear);

5. Context = 1 (Simulated range = 0; Simulated combat = 1);

6. CDA_MisfireIndicator = true (No fault condition: false or OFF);

7. LAV_Running = false (needed for Low Battery fault; otherwise can be true); and

8. CannonManualSafety = true.

Select one or more of the following to test the response:

- GunArmed_IndicatorLight = false (No fault condition: true or ON);
- SEAR_IndicatorLight = true (No fault condition: false or OFF);
- GunArmed_IndicatorLight = false (No fault condition: true or ON);
- Sear_IndicatorLight = true (No fault condition: false or OFF);
- LowAmmo_IndicatorLight = true (No fault condition: false or OFF);

- WPN_IndicatorLight = true (No fault condition: false or OFF);
- LowBattery_IndicatorLight = true (No fault condition: false or OFF); and
- TempFault_IndicatorLight = true (No fault condition: false or OFF).

Then initiate execution:

1. BEGIN = t.

A.3.3 Misfire Drill, BPI in MISFIRE, Cannon Manual Safety ON: Drill a1B (LAV III) or Ser #2 (LAV 6)

1. BEGIN = f;
2. WEAPON_Selected = 1 (25mm Cannon);
3. MISFIRE = t;
4. BPI_State = 2;
5. CannonManualSafety = t; and
6. CDA_MisfireIndicator = t.

Set misfire to start simulation executing:

1. BEGIN = t.

Pause simulation when CC orders Single Shot:

1. MISFIRE = f;
2. BPI_State = 1; and
3. Canon_Status = 0.

A.3.4 Misfire Drill, BPI in MISFIRE, Cannon Manual Safety OFF: Drill a1B (LAV III) or Ser #3 (LAV 6)

1. BEGIN = f;
2. WEAPON_Selected = 1 (25mm Cannon);
3. MISFIRE = t;
4. BPI_State = 2;

5. CannonManualSafety = f; and

6. CDA_MisfireIndicator = t.

Set misfire to start simulation executing:

1. BEGIN = t

Pause simulation when CC orders Single Shot:

1. MISFIRE = f;

2. BPI_State = 1; and

3. Canon_Status = 0.

A.4 COAX Immediate Action Drill (Stoppage)

A.4.1 Gun Fails to Fire on Initial Burst – Ammo Select Wrong; Drill a1 (LAV III) or SER #1 (LAV 6)

1. BEGIN = f;

2. CONTEXT = 1 (simulated combat);

3. WEAPON_Selected = 2 (COAX MG);

4. COAX_Stoppage = t;

5. COAX_RoundsFired = 0 (value from 0 to 4 indicates initial burst; more than 4 indicates subsequent burst); and

6. AmmoBinSelected = 2 (secondary bin setting).

Begin the simulation:

1. BEGIN = t.

Pause the simulation once the drills start and reset the stoppage flag:

1. COAX_Stoppage = f.

A.4.2 Gun Fails to fire on Initial Burst – Ammo Select Correct; COAX Safety Engaged; Drill a2 (LAV III) or SER #1 (LAV 6)

1. BEGIN = f;

2. CONTEXT = 1 (simulated combat);

3. WEAPON_Selected = 2 (COAX MG);

4. COAX_Stoppage = t;

5. COAX_RoundsFired = 0 (value from 0 to 4 indicates initial burst; more than 4 indicates subsequent burst);

6. AmmoBinSelected = 1 (primary bin setting); and

7. COAX_Safety = t (to test network when CDA check passes without finding fault).

Begin the simulation:

1. BEGIN = t.

Pause the simulation once the drills start and reset the stoppage flag:

1. COAX_Stoppage = f.

A.4.3 Gun Fails to Fire on Initial Burst – Ammo Select Correct; COAX Safety Set to FIRE; Drill a2 (LAV III) or SER #1 (LAV 6)

1. BEGIN = f;

2. CONTEXT = 1 (simulated combat);

3. WEAPON_Selected = 2 (COAX MG);

4. COAX_Stoppage = t;

5. COAX_RoundsFired = 0 (value from 0 to 4 indicates initial burst; more than 4 indicates subsequent burst);

6. AmmoBinSelected = 1 (primary bin setting); and

7. COAX_Safety = f.

Select any of the following four faults as shown to test the response to each of those tasks:

1. GunArm_IndicatorLight = f (gun is armed if the indicator is on, i.e., t);

2. LowAmmo_IndicatorLight = t;

3. LowBattery_IndicatorLight = t; and

4. TempFault_IndicatorLight = t.

Begin the simulation:

1. BEGIN = t.

Pause the simulation once the drills start and reset the stoppage flag:

1. COAX_Stoppage = f.

If testing the low battery warning, “start” the LAV when instructed by the CC:

1. LAV_Running = t.

A.4.4 Gun Fails to Fire on Initial Burst – Ammo Select Correct – Working Parts Fully to the Rear; Drill a3/4/5 (LAV III) or SER #1 (LAV 6)

1. BEGIN = f;
2. CONTEXT = 1 (simulated combat);
3. WEAPON_Selected = 2 (COAX MG);
4. COAX_Safety = f;
5. STOPPAGE = t; and
6. AmmoBinSelected = 1 (primary bin setting).

Do not select any of the following faults:

- GunArm_IndicatorLight = t;
- LowAmmo_IndicatorLight = f;
- LowBattery_IndicatorLight = f; and
- TempFault_IndicatorLight = f.

Select the position of the COAX “working parts” (i.e., the “action”):

1. COAX_ActionPosition = 2:
    a. The other COAX working parts checks follow a similar setup but use different values for the COAX_ActionPosition according to the following values:

i. 0 OR greater than 3: Assignment error (working parts position not assigned; this indicates a programming logic error);

ii. 1: Working parts fully forward;

iii. 2: Working parts fully to the rear; and

iv. 3: Working parts partially forward.

Begin the simulation:

1. BEGIN = t.

If you want the COAX to fire normally, pause the simulation when the CC reports “Firing Circuit” and sent reset the COAX_Stoppage variable as the ICGS would indicate (false for operating normally or true for stoppage):

1. COAX_Stoppage = f.

A.5 Fire Order and Engagement Drills

1. open the Execution Control window:
   a. press the checkmark button on the IPME banner, which will check for errors and launch the Execution Control window.
2. select the Display tab at the bottom of the Execution Control window:
   a. select the Show Watch Variables from the left-hand side of the Display tab.
3. select the Control tab:
   a. ensure Animate Execution checkbox is enabled; and
   b. set Animation Speed slider to about two-thirds (adjust as necessary).
4. press Run:
   a. the model will start but not much will happen until certain parameters are entered; and
   b. if you want more control over execution, you can hit Pause, then use the Step button to step through the execution manually.
5. start entering execution parameters in the Watch Variables window to match the example drill conditions:
   a. set the LAV_Type to either 3 or 6;

b. set the TargetType:

   i. BMP = 1;

   ii. APC = 2;

   iii. Tank = 3;

   iv. Transport = 4; and

   v. Men = 5.

c. set the TargetQTY:

   i. for now, only use 1.

d. set a nominal TargetRange according to the example scenario:

   i. BATTLE <= ~800;

   ii. Estimated <= 1500; and

   iii. Lase > 1500.

e. if you want to test a DOUBLE laser range finder return set LRF_Double to true, otherwise leave as false.

f. decide who is going to see the target first. While both are false, the model will idle and you can make further changes, but once one is set to true, the model starts:

   i. if CC, set CC_TargetInFOV to true;

   ii. if GNR, set GNR_TargetInFOV to true; and

   iii. do not set both to true. This condition has not yet been tested, but detection is probabilistic. When that operator sees the target, the corresponding detection variable will be set to true, as will the flags that note who initiated the engagement.

g. when the CC and GNR are on the target and ready to shoot, there is not much feedback. However, after they are both ON and the GNR reports the range, the CC has a few options that the user can pass to the simulation through the RangeConfirmation variable:

   i. WAITING: default state;

   ii. ACCEPTABLE: continue with the engagement (i.e., shoot);

   iii. LAST: select the last LRF value;

   iv. FIRST: select the first LRF value (which is the default selection); and

   v. RELASE: tell the GNR to release the target.

h. the communications between the turret crew will be displayed in the Watch Variables list:

   i. the speaker value is in TextVoice; and

   ii. The message value is in TextMessage.

Execution should run uninterrupted to completion, but you can Pause and Step if you wish. If execution happens very rapidly, make sure that Animate Execution is enabled and slow down the animation speed to between one-half and three-fourths.

A.6 Action Drill

1. open the Execution Control window:

   a. press the checkmark button on the IPME banner, which will check for errors and launch the Execution Control window.

2. select the Display tab at the bottom of the Execution Control window:

   a. select the Show Watch Variables from the left-hand side of the Display tab.

3. select the Control tab:

   a. ensure Animate Execution checkbox is enabled;

   b. set Animation Speed slider to about two-thirds (adjust as necessary); and

   c. press Run:

      i. the model will start but not much will happen until certain parameters are entered; and

      ii. if you want more control over execution, you can hit Pause, then use the Step button to step through the execution manually.

4. start entering execution parameters in the Watch Variables window to match the example drill conditions:

   a. LAV_Type: 3 or 6 (for LAV III or LAV 6 as desired);

   b. BPI_State: 1 (SEAR indicated; will be controlled by ICGS);

   c. Set other conditions that will be controlled by the ICGS as desired:

   i. MBGD_InUse: true/false; and

   ii. pintleMG_Present: true/false (whether the pintle mount machine gun is present).

5. start the simulation:

   a. BEGIN: true.

Pause execution after the CC reports the BPI is in SEAR and set to misfire:

1. set conditions for the CC and GNR to test the weapon system:

   a. MISFIRE: true (will be controlled by ICGS); and

   b. BPI_State: 2 (misfire indicated; will be controlled by ICGS).

Pause execution after the CC reports RESET to change the BPI_State:

1. MISFIRE: false (will be controlled by ICGS); and

2. BPI_State: 1 (will be controlled by ICGS).

A.7 Start Mode

1. LAV_Type: 3 or 6 (to select the LAV variant); and

2. BEGIN: true.

A.8 Turret Power Up or Down

These two drills are in the same IPME project, but they are separate drills in the composite model. They should be tested one at a time and this is controlled by setting the variable TEMP_TurretPower to TRUE for Turret Power UP or FALSE for Turret Power Down:

1. TEMP_TurretPower: true/false (to select the desired drill); and

2. BEGIN: true.

A.9 Procedures without Crew Interaction

There is no interaction amongst the crew during the following procedures. Each crew member performs their own duties:

1. opening up; and

2. closing down.

APPENDIX B CONSOLIDATED MODEL VARIABLE LIST

This is a list of variables used within the composite and component models, describing the variable, its type, and initial value. The list also cites the component model that the variable is used in, whether the variable corresponds to a speech recognition expression and an expectation of whether the variable will be used during integration with the LAV vehicle simulation (ICGS).

NOTE: The following table is an extract, provided here as it was originally prepared by the modeller.

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
AimPoint	This is the logical assessment of whether the point of aim is close enough to the target centre of mass to be considered on target. Values can be: • OFF==false • ON==true. This is different from the LAV_AimPoint array that has the coordinates of the aim point.	Boolean	False	• Fire Order and Engagement • Misfire	No	Yes
AmmoBinSelected_ CC	An integer variable that indicates the ammo bin that the CC has selected on the CC PDA. There are two ammo bins for the cannon but there is also the case where there is no ammo loaded. The COAX MG has only a single bin of ammo as does the Pintle Mount MG; typically, this is ball ammo. DO NOT CHANGE Valid values are: • 0==UNLOADED • 1==PRIMARY_BIN • 2==SECONDARY_BIN	Integer	0	• Action • Fire Order and Engagement • Misfire	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
AmmoBinSelected_GNR	An integer variable that indicates the ammo bin that either the GNR or the CC has selected on the PDA. There are two ammo bins for the cannon but there is also the case where there is no ammo loaded. The COAX MG has only a single bin of ammo as does the Pintle Mount MG; typically, this is ball ammo. DO NOT CHANGE Valid values are: • 0==UNLOADED • 1==PRIMARY_BIN • 2==SECONDARY_BIN	Integer	0	• Action • Fire Order and Engagement • Misfire	No	Yes
APC	A target type	Integer	2	Fire Order and Engagement	No	No
ARMED	A flag that indicates the state of a weapon; opposite to SAFE; may be equivalent to READY	Integer	1	• Action • Turret Make Safe	No	No
AUTO	Variable used with the GCP_SightMode to indicate the targeting mode of the sight	Integer	1			
BALL	A variable indicating that the Coax or the MG is loaded with ball ammo	Integer	1			
BATTLE	A range estimate of less than 800m. The BATTLE range actually depends on the ammo type but for now, just use a single value	Integer	800	Fire Order and Engagement	Yes	No
BattleOverride	State of the Battle Override safety switch on the CDA: • ON = true	Boolean	False	• Action • Misfire	Yes	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	- OFF = false NOTE: This is not the Commander's Override switch			- Opening Up		
BattleOverride_Switch	CC switch to override selected safety interrups: - ON = true - OFF = false	Boolean	False			
BEGIN	A temporary logical variable used to start execution of the network, allowing the user to set variable states for debugging purposes. Should be removed from the network when made operational	Boolean	False	ALL	No	No
BEGIN_CDAcheck	A logical variable indicating that the CC has started checking the CDA	Boolean	False	Misfire	No	No
BMP	A target type	Integer	1	Fire Order and Engagement	No	No
BPI_MISFIRE	A state variable used with the BPI_State variable	Integer	2	- Action - Misfire - Fire Order and Engagement - Prove weapons	No	Yes
BPI_SEAR	A state flag used with the BPI_State variable	Integer	0	- Action - Fire Order and Engagement - Misfire - Prove weapons	No	Yes
BPI_State	This is a status flag that indicates the state of the:	Integer	0	- Action	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	Bolt Position Indicator • BPI_SEAR == 1 • BPI_MISFIRE == 2			• Misfire • Fire Order and Engagement • Prove Weapons		
BridgePlates_InPlace	A logical flag that indicates the gun action is present and the covering plates are in position InPlace == true Removed == false part of the cannon inspection during prove weapons drill	Boolean	False	Prove weapons	No	Yes
Cannon_Proven	A logical flag that indicates the crew have verified whether the 25mm gun is loaded or clear	Boolean	False	Prove weapons	No	No
Cannon_RoundsFired	A count of the number of rounds fired from the cannon in the current engagement. It is used to control the response in the event of a misfire. It should be reset to 0 after an engagement	Integer	0	Misfire	No	Yes
Cannon_Status	A status flag for the 25mm cannon. Values are INT: 0 == operational 1 == misfire on initial burst 2 == misfire on subsequent burst	Integer	0	Misfire	No	Yes
CannonAmmo	This is an array of strings that describe the type of ammunition available to the main gun. Options are by index (where the index can be thought of as the ammo bin): 0 == CLEAR (unloaded) 1==SABOT (primary)	String	CLEAR	Fire Order and Engagement	No	Maybe

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	2==FRANG (secondary) Could also add HE as a third type					
CannonBurst	This is a variable to index the number of bursts fired in any given engagement. It is intended for use with the misfire drill. It should be reset to 0 at the end of an engagement, probably after a TARGET STOP order or a GUN MAKE SAFE order	Integer	0	Misfire	No	Yes
CannonElevated	A logical flag that indicates the cannon has been elevated for inspection	Boolean	False	Prove weapons	No	Yes
CannonElevated_Manually	The GNR manually elevates the cannon with the power off so that the CC can inspect the weapon during proving	Boolean	False	• Action • Misfire	No	No
CannonElevation	A variable indicating the angle of elevation of the cannon	FloatingPoint	0	Misfire	No	Yes
CannonFiring	Logical to indicate that the cannon is firing the selected ammunition/modes	Boolean	False	Action	No	Yes
CannonManualSafety	This is the status of the manual safety lever on the cannon block: ON == true (set to SAFE) OFF == false (set to FIRE)	Boolean	False	• Action • Misfire • Opening Up • Prove Weapons • Turret Make Safe	No	Yes
CannonMuzzleCover	In place == ON == true Removed == OFF == false	Boolean	False	• Close Down • Opening Up	No	Maybe
CannonRoundsPres	State of ammunition in the 25mm cannon feeder:	Boolean	False	Prove weapons	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
ent	Present == true Not Present == false					
CC_CommandRange	A flag indicating that the CC has specified that an estimated range value will be used in the firing solution	Boolean	False	- Fire Order and Engagement	Yes	No
CC_EngagementTag	A flag to control the engagement drill for the CC	Integer	101	- Fire Order and Engagement - Misfire	No	No
CC_FO_AmmoType	Designation of ammunition type	String	0	Fire Order and Engagement	Yes	No
CC_FO_construct	A variable to collect the CC's fire order commands into a single string	String	0	Fire Order and Engagement	No	No
CC_FO_Range	Options are LASE, BATTLE, number LASE == 0 BATTLE == 1500 m number == estimated range in metres	String	0	Fire Order and Engagement	Yes	No
CC_FO_TargetType		String	0	Fire Order and Engagement	Yes	No
CC_FreeToTraverse	CC's indication that his side of the turret is clear and ready to traverse	Boolean	False	Action	Yes	Yes
CC_HatchClosed	A logical to indicate the state of the CC Hatch: Open == OFF == false Closed == ON == true	Boolean	False	Turret Power Down	No	Maybe
CC_InitiatedEngagement	A flag that indicates it is the CC who has initiated the engagement	Boolean	False	Fire Order and Engagement	No	No
CC_LaserReadout	A logical var that indicates the state of the CC's	Boolean	False	Turret Power Up	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	laser display					
CC_LaserReadoutPower	Crew Commanders laser readout display power ON = true OFF = false	Boolean	False	Action	No	Yes
CC_LWDPower	Crew Commander's laser warning display ON = true Off = false	Boolean	False	Action	No	Yes
CC_MisfireTag	A flag to control the misfire/stoppage drills for the CC	Integer	102	• Fire Order and Engagement • Misfire	No	No
CC_OnTarget	A boolean flag to indicate the CC believes the vehicle in the sight is the desired target	Boolean	False	Fire Order and Engagement	No	Yes
CC_ORDER_200Ready	An order from the CC at the end of clearing a misfire fault, indicating that the GNR should select 200 RPM and begin firing	Boolean	False	Misfire	Yes	No
CC_ORDER_CannonElevate	A logical var indicating that the CC has ordered the GNR to elevate the cannon for inspection	Boolean	False	Prove weapons	Yes	No
CC_ORDER_DriverStartVehicle	A CC order for the driver to start the vehicle	Boolean	False	Misfire	Yes	No
CC_ORDER_Fire	A logical flag to indicate that the CC has given the executive order to fire	Boolean	True	Fire Order and Engagement	Yes	No
CC_ORDER_FirstLast	A flag to indicate which of the LRF values the CC is accepting: Values are: FIRST == 1 LAST == 2	Integer	0	Fire Order and Engagement	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
CC_ORDER_Lase	Crew commander orders a LRF LASE	Boolean	False	Fire Order and Engagement	Yes	No
CC_ORDER_Laser ON	A flag that indicates the CC has ordered the GNR to turn the laser on	Boolean	False	Misfire	Yes	No
CC_ORDER_Misfire Wait	A logical var indicating that the CC has recognized the MISFIRE report and instructed the GNR to wait till the problem is found	Boolean	False	Misfire	Yes	No
CC_ORDER_Range Change	A flag indicating the CC is changing the range to target	Boolean	False	Fire Order and Engagement	Yes	No
CC_ORDER_Ready	A CC order that the weapon is ready and should be fired	Boolean	False	• Action • Misfire	Yes	No
CC_ORDER_ReLase	A flag that indicates that the CC has ordered the GNR to relase or relay on the target	Boolean	False	Fire Order and Engagement	Yes	No
CC_ORDER_Reset	An order from the CC indicating that the gun misfire has been fixed and is ready to fire	Boolean	False	• Action • Misfire	Yes	No
CC_ORDER_Single Shot	A CC order for the gunner to fire the cannon BUT after the CC has ordered single shot or the GNR has selected single shot because of a misfire procedure	Boolean	False	Misfire	Yes	No
CC_ORDER_Single Shot	An order from the CC indicating that the GNR should select the single shot mode of the 25mm cannon. Can be used normally as a ranging burst or during misfire troubleshooting	Boolean	False	Misfire	Yes	No
CC_ORDER_Target Stop	Logical flag to indicate that the CC has ordered the GNR to stop beating up the target	Boolean	False	Fire Order and Engagement	Yes	No
CC_ORDER_Turret MakeSafe	An order from the CC to secure the weapons system	Boolean	False	• Misfire • Turret Make	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
				Safe		
CC_ORDER_Turret PowerDown	A logical indicating the CC has ordered the turret to be powered down	Boolean	False	Turret Power Down	Yes	No
CC_ORDER_Turret PowerDown	A logical indicating the CC has ordered the turret to be powered down	Boolean	False	Turret Power Down	Yes	No
CC_ORDER_Turret PowerUp	A logical variable indicating that the CC has ordered that the turret be powered up	Boolean	False	Turret Power Up	Yes	No
CC_Override	The state of the Commander’s override switch for the guns ON = true OFF = false This is different from the Battle Override switch	Boolean	False	Action	Yes	Yes
CC_RangeEstimate	A value of the CC's judgement of the range to the tgt to the nearest 100 metres	Integer	0	Fire Order and Engagement	Yes	No
CC_REPORT_Cann onClear	State of the 25 mm cannon CLEAR == true (unloaded) not clear == false (rounds in the chamber/feeder)	Boolean	False	Prove weapons	Yes	No
CC_REPORT_Cann onSafe	A flag that indicates the CC has made the cannon manual safety and reported the cannon safe	Boolean	False	Prove weapons	Yes	No
CC_REPORT_Cann onSafe	A flag that indicates the CC has made the cannon manual safety and reported the cannon safe	Boolean	False	Prove weapons	Yes	No
CC_REPORT_Coax Clear	A logical flag indicating the CC has proven the COAX MG and reported	Boolean	False	Prove weapons	Yes	No
CC_REPORT_Com dOverrideOff	A logical indicating the CC has turned the commander's override switch off and reported	Boolean	False	• Action • Turret Power Down	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
CC_REPORT_CommandOverrideOff	A logical indicating the CC has turned the commander's override switch off and reported	Boolean	False	Turret Power Down	Yes	No
CC_REPORT_CSAMOff	A logical indicating the CC has turned the CSAM off and reported	Boolean	False	Turret Power Down	Yes	No
CC_REPORT_FiringNow	A logical flag indicating that the CC is firing the weapon instead of the GNR	Boolean	False	Misfire	Yes	No
CC_REPORT_IIOff	A logical indicating the CC has turned the image intensifier off and reported	Boolean	False	Turret Power Down	Yes	No
CC_REPORT_LWDOff	A logical indicating the CC has turned their laser display off and reported	Boolean	False	Turret Power Down	Yes	No
CC_REPORT_MGproven	A logical flag indicating that the CC has proven the pintle mount MG and reported	Boolean	False	Prove weapons	Yes	No
CC_REPORT_MGproven	A logical flag indicating that the CC has proven the pintle mount MG and reported	Boolean	False	Prove weapons	Yes	No
CC_REPORT_ProlongedStoppage	A logical variable indicating that someone has declared a prolonged stoppage for the COAX MG and that secondary actions are required to get the COAX to fire	Boolean	False	Misfire	Yes	No
CC_REPORT_RadioOff	A logical var indicating the CC has turned the radios off and reported	Boolean	False	Turret Power Down	Yes	No
CC_REPORT_RIGHTMBGDproven	A logical flag indicating the CC has proven the right MBGD state and reported	Boolean	False	Prove weapons	Yes	No
CC_REPORT_SEAR	A logical flag to indicate that the CC has observed the BPI, noted that the cannon is in SEAR, and reported	Boolean	False	Action	Yes	No
CC_REPORT_ThermalOff	A logical indicating the CC has turned the CC thermal sight off and reported	Boolean	False	Turret Power Down	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
CC_REPORTED_LWSOff	A logical indicating the CC has turned the Laser warning system off and reported	Boolean	False	Turret Power Down	Yes	No
CC_ReportTurretLockOn	A logical indicating the CC has locked the turret and reported	Boolean	False	Turret Power UP	Yes	No
CC_StartMode	Logical variable that indicates whether the CC has configured the CC's side of the turret to the default Start Mode	Boolean	False	Start	Yes	No
CC_Tag	A tag variable to control the network execution of the CC Value is 1 DO NOT CHANGE	Integer	1	All	No	No
CC_TargetDestroyed	A logical flag to indicate that the CC thinks that the target has taken enough abuse and should leave it alone	Boolean	False	Fire Order and Engagement	Yes	No
CC_TargetDetected	A logical variable that indicates the CC has detected a target; used to initiate the fire order and engagement drill	Boolean	False	Fire Order and Engagement	No	No
CC_TargetInFOV	A logical variable that indicates there is an active target in the Field of View of the crew commanders' sights	Boolean	False	Fire Order and Engagement	No	Yes
CC_TargetQty	An integer that indicates the number of targets the CC observes	Integer	0	Fire Order and Engagement	No	Yes
CC_ThermalSightMode	Crew Commander's thermal sight operating mode WIDE = 0 NARROW = 1	Integer	0	Action	No	Yes
CC_ThermalSightPower	ON = true OFF = false	Boolean	False	Action	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
CC_VERIFY_COAX action	A logical variable indicating that the CC is checking the COAX action to determine the MG state during a stoppage	Boolean	False	Misfire	No	No
CCTD_VideoSwitch		Integer	0	Start Mode	Yes	Yes
CCTD_VideoSwitch		Integer	0	Start Mode	No	Yes
CDA_FaultFound	Logical flag that indicates a CDA fault that has caused a misfire or stoppage has been found and corrected	Boolean	False	Misfire	No	No
CDA_MisfireIndicato r	An indicator on the CDA that shows whether the gun in in misfire ON == true OFF == false	Boolean	False	Misfire	No	Yes
CDAPower	Flag indicating the state of power to the CDA.	Boolean	False	Turret Power Down	No	Yes
CLEAR	A variable indicating the weapon is unloaded	Integer	0	Misfire	No	Maybe
CLOSED	A variable used to set the status of any of the hatches	Integer	1	Action	No	No
COAX CLEAR	A report by the CC indicating that the COAX is unloaded			Prove weapons	Yes	No
COAX_ActionPositio n	A variable used to indicate the position of the working parts of the COAX MG Error in assignment == 0 Fully Forward = 1 Fully Back = 2 Partially Forward = 3	Integer	0	Misfire	No	Maybe
COAX_ActionPositio n_Back	A variable used to indicate the position of the working parts of the COAX MG	Integer	2	Misfire	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	Error in assignment == 0 Fully Forward = 1 Fully Back = 2 Partially Forward = 3					
COAX_ActionPosition_Forward	A variable used to indicate the position of the working parts of the COAX MG Error in assignment == 0 Fully Forward = 1 Fully Back = 2 Partially Forward = 3	Integer	1	Misfire	No	Yes
COAX_ActionPosition_Partial	A variable used to indicate the position of the working parts of the COAX MG Error in assignment == 0 Fully Forward = 1 Fully Back = 2 Partially Forward = 3	Integer	3	Misfire	No	Yes
COAX_Clear	A logical variable indicating that the COAX MG is unloaded	Boolean	False	Prove weapons	Yes	Maybe
COAX_RoundsFired	A count of the number of rounds fired during a fire order/engagement	Integer	0	Misfire	No	Yes
COAX_Safety	A logical inticating the state of the COAX safety switch Safe == ON == true Fire == OFF == false	Boolean	False	Misfire	No	Yes
COAX_Status	A status flag for the 25mm cannon. Values are INT: 0 == operational	Integer	0	Misfire	No	Maybe

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	1 == misfire on initial burst (STOPPAGE_InitialBurst) 2 == misfire on subsequent burst (STOPPAGE_SubsequentBurst)					
COAX_Stoppage	A logical variable indicating the COAX weapon has misfired. It should be set by the ICGS and COAX_Status should be used in the model to reflect the ICGS state. If the COAX is operational, this value is false. If the COAX has a fault and a stoppage has occurred, this value is true	Boolean	False	Misfire	No	Yes
COAX_Stoppage_InitialBurst	An int variable that indicates the COAX MG has failed during an engagement on the initial burst. Used with COAX_Status	Integer	1	Misfire	No	Yes
COAX_Stoppage_SubsequentBurst	A int variable that indicates the COAX MG has failed during an engagement sometime after the initial burst. Used with COAX_Status	Integer	2	Misfire	No	Yes
COAXAmmo	This is an array of strings that describes whether COAX MG is loaded or not. There is only one type of ammo and one bin available for the COAX. Options are by index (where the index can be thought of as the ammo bin): 0==CLEAR (unloaded) 1==BALL	String	CLEAR	Fire Order and Engagement	No	Maybe
CoaxMounted	The COAX MG is present: Mounted == true Not Mounted == false	Boolean	False			

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
COAXMuzzleCover	In place == ON == true Removed == OFF == false	Boolean	False	- Close Down - Opening Up	No	Maybe
ComdOverride	Commander's override switch on the GCP. This switch allows the CC to take control of the weapon system to manually perform the shoot ON == true OFF == false NOTE: This is NOT the Battle Override switch	Boolean	True	- Start Mode - Misfire - Turret Power Up - Turret Make SafeSHOULD THIS ALSO BE IN THE POWER DOWN PROC?	No	Yes

CONTEXT	A state flag for the type of training scenario assumed:
	Training == 0 // simulating on the range
	Combat == 1 // simulating war
		Integer	0	Misfire	No	No

CONTEXT_Combat	A flag that indicates the simulation represents action in combat; use with variable CONTEXT to control network execution	Integer	1	Misfire	No	No
CONTEXT_Training	A flag that indicates the simulation represents action on the training range; use with variable CONTEXT to control network execution	Integer	0	Misfire	No	No
CSAMCover	In place == ON == true Removed == OFF == false	Boolean	False	Close DownSHOULD THIS BE IN THE ACTION DRILL/OPENING UP DRILL AS WELL?	No	Maybe

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
CSAMPower	ON == true OFF == false	Boolean	False	• Turret Power Up • Turret Power Down	No	Yes
DAGRPower	Power state of the DAGR ON == true OFF == false	Boolean	False	• Action • Turret Power Up • Turret Power Down	No	Yes
DaySightMode	Off = 0 on = 1 BIT = 2 Note that this is an integer so do not use ON/OFF, which are Boolean vars	Integer	0	Opening Up	No	Yes
DCB	ON = true OFF = false	Boolean	False	Action	No	Yes
DriftNull	State of the cannon stabilization mechanism Auto == ON == true OFF == False Drifting == OFF == false (i.e. stab has a problem)	Boolean	False	Action	Yes	Yes
DrivingHatchesDown	State of driver while driving (TENTATIVE): Hatches Down == true Hatches Up == false	Boolean	True	Opening Up	No	No
DRV_AutoSwitch	Driver's “auto” power switch state: OFF == false ON == true	Boolean	False	• Action • Close Down • Opening Up	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
DRV_AuxPower	Driver's auxiliary power switch state: OFF == false ON == true	Boolean	False	• Action • Close Down • Opening Up	No	Yes
DRV_DCB_Power	Logical state of the driver's DCB: ON == true OFF == false	Boolean	False	• Close Down • Opening Up	No	Yes
DRV_DVA_Power	Logical state of the driver's DVA: ON == true OFF == false	Boolean	False	• Close Down • Opening Up	No	Yes
DRV_Hatch	OPEN = 0 CLOSED = 1 LOCKED = 2	Integer	0	• Action • Opening Up	No	Yes
DRV_LWDPower	Power status of the Driver's Laser Warning Receiver Display: ON = true OFF = false	Boolean	False	• Action • Close down	No	Yes
ELEVATING	A report by the GNR that they are elevating the cannon for inspection			Prove weapons	Yes	No
FinishedArmingWeapon	A logical flag used to indicate when the GNR has finished arming the weapon	Boolean	False	Fire Order and Engagement	No	No
FIRST	A int variable corresponding to the datum returned by the LRF Alternative to LAST Used as an index for the LRF_Range array to select the first distance return from the LRF and to set the LRF_FirstLastSwitch to the corresponding value	Integer	1	• Action • Fire Order and Engagement • Start	No	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
Flag	This is a physical flag that the CC puts up during training on the range. to indicate the state of the turret in some way RED YELLOW GREEN	Integer	0	Misfire	No	Yes
FlatTire_LHS		Boolean	False	Opening Up	No	Yes
FlatTire_RHS		Boolean	False	Opening Up	No	Yes
FO_Issued		Boolean	False	Fire Order and Engagement	Yes	No
FRANG	An ammunition type used in the fire order	String	FRANG	Fire Order and Engagement	Yes	No
GCP_SightMode	Gunner's targeting reticle modes. Options are: AUTO == 1	Integer	0			
GNR_AuthorizedToLase	A flag that indicates whether the GNR has been authorized to lase targets without direction from the CC. This is mainly for the LAV 3 but could be set for the LAV 6 as well.	Boolean	False	Fire Order and Engagement	Maybe	No
GNR_Contact_construct	A string variable to build the GNR's announcement of a detected contact	String	0	Fire Order and Engagement	No	No
GNR_ContactReported	Logical flag used to indicate the GNR has initiated an engagement and reported it to the CC	Boolean	False	Fire Order and Engagement	Yes	No
GNR_EngagementTag	A flag to control the engagement drill for the GNR	Integer	201	• Fire Order and Engagement • Misfire	No	No
GNR_FireWeapon	A logical expression indicating that the GNR is pulling the trigger to fire the selected weapon.	Boolean	False	• Fire Order and Engagement	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	The expected use is that the GNR will pull and release after the desired burst length is obtained			• Misfire		
GNR_FreeToTraverse	GNR's indication that their side of the turret is clear and ready to traverse	Boolean	False	• Action	Yes	No
GNR_Hatch	A logical to indicate the state of the GNR Hatch Open == OFF == false Closed == ON == true	Boolean	False	• Close Down • Turret Power DownSHOULD THIS BE IN THE ACTION DRILL/OPENING UP DRILL AS WELL?	No	Maybe
GNR_InitiatedEngagement	A flag that indicates the GNR has seen a tgt and no FO has been issued so the GNR initiates a contact report	Boolean	False	Fire Order and Engagement	Yes	No
GNR_MaintainingAimPoint	A logical flag indicating that the GNR is monitoring their aim point; used to control the GNR engagement procedure	Boolean	False	Fire Order and Engagement	No	Yes
GNR_MeasuredRange	A string variable that records what the GNR reports as the LRF FIRST return	String		Fire Order and Engagement	Maybe	Yes
GNR_MisfireTag	A flag to control the misfire/stoppage drills for the GNR	Integer	202	• Fire Order and Engagement • Misfire	No	No
GNR_MonitoringLOS	A logical flag indicating the GNR is monitoring the line of sight to the target used to control the GNR engagement task flow	Boolean	False	Fire Order and Engagement	No	Yes
GNR_ObserveThroughSight	A logical variable that indicates the GNR is observing through the weapon system sight	Boolean	False	Action	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
GNR_Observing	Flag to indicate that the GNR is observing downrange through one of the sights	Boolean	False	Fire Order and Engagement	No	Maybe
GNR_RangeConfirmation	A string containing the value that the GNR reports back to the CC after receiving a range order	String		Fire Order and Engagement	Yes	No
GNR_RangeEstimate	GNRs estimate of the target range to the nearest 100m	Integer	0	Fire Order and Engagement	Yes	Yes
GNR_RangeReported	A flag that indicates the GNR has reported the range to tgt	Boolean	False	Fire Order and Engagement	Yes	No
GNR_REPORT_AmmoSelectCorrect	A logical flag indicating that the GNR has corrected the ammo select button on the power controller Used with REPORT_AmmoSelectWrong to trigger CC actions	Boolean	False	Misfire	Yes	No
GNR_REPORT_AmmoSelectWrong	A logical flag indicating that the GNR has reported that the ammo select button on the power controller was set to the wrong bin Used with REPORT_AmmoSelectCorrect to trigger CC actions	Boolean	False	Misfire	Yes	No
GNR_REPORT_AuxPowerOff	A logical that indicates the GNR has turned the turret auxiliary power off and reported	Boolean	False	Turret Power Down	Yes	No
GNR_REPORT_CoaxStoppage	A logical variable indicating that the GNR has reported they have tried to fire the COAX MG but that it has failed to fire	Boolean	False	Misfire	Yes	No
GNR_REPORT_FiringNow	A logical indicating that the GNR is firing. NOTE: THE CC WILL ALSO REPORT "FIRING NOW" WHEN THE CC TAKES CONTROL OF THE SHOOT AND FIRES THE WEAPON	Boolean	False	• Fire Order and Engagement • Misfire	Yes	No
GNR_REPORT_Gu	A flag that indicates the GNR has made the gun	Boolean	False	Turret Make Safe	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
nOff	safe and reported					
GNR_REPORT_LaserOff	A report from the GNR indicating a change in the laser power setting	Boolean	False	Turret Power Down	Yes	No
GNR_REPORT_LaserOff	A logical flag indicating the GNR has turned the LRF off and reported	Boolean	False	Prove weapons	Yes	No
GNR_REPORT_LaserOn	A report from the GNR indicating a change in the laser power setting	Boolean	False	Turret Power UP	Yes	No
GNR_REPORT_LaserOn	A report from the CC in the Turret Make Safe drill	Boolean	False	Misfire	Yes	No
GNR_REPORT_LaserPowerOff	A flag indicating that the GNR has turned the laser off and reported	Boolean	False	Turret Make Safe	Yes	No
GNR_REPORT_LEFTMBGDproven	A logical variable indicating the GNR has proven the state of the left bank of MBGDs and reported	Boolean	False	Prove weapons	Yes	No
GNR_REPORT_Misfire	A variable used to indicate the GNR has reported a 25mm cannon misfire	Boolean	False	- Action - Misfire	Yes	No
GNR_REPORT_SingleShot	A logical flag to indicate that the GNR has selected SINGLE	Boolean	False	Misfire	Yes	No
GNR_REPORT_StabOn	A logical indicating the GNR has turned the STAB On and reported			Turret Power UP	Yes	No
GNR_REPORT_ThermalOff	A logical indicating that the GNR has turned the thermal sight off and reported	Boolean	False	Turret Power Down	Yes	No
GNR_REPORT_ThermalOff	A logical indicating the GNR has turned their termal sight Off and reported	Boolean	False	Turret Power Down	Yes	No
GNR_REPORT_ThermalOn	A logical indicating the GNR has turned their termal sight On and reported	Boolean	False	Turret Power UP	Yes	No
GNR_REPORT_Tur	A logical flag indicating the GNR has turned the	Boolean	False	Prove weapons	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
retOff	turret power off and reported					
GNR_REPORT_TurretOn	A logical flag that indicates the GNR has made the turret power switch ON and reported	Boolean	False	Turret Power UP	Yes	No
GNR_REPORT_WeaponOff	A logical flag indicating the GNR has turned the weapon system power off and reported	Boolean	False	Prove weapons	Yes	No
GNR_REPORT_WeaponOn	A logical flag indicating that the GNR has made the weapon power ON and reported	Boolean	False	Turret Power UP	Yes	No
GNR_REPORT_WeaponPowerOff	A flag indicating that the GNR has turned the weapon system power off and reported	Boolean	False	Turret Make Safe	Yes	No
GNR_REPORTED_AuxPowerOn	A logical flag indicating that the GNR has made the turret auxiliary power switch ON and reported	Boolean	False	Turret Power UP	Yes	No
GNR_ReportedFirst	A logical flag to indicate the CC has asked for and the GNR has reported the FIRST LRF value	Boolean	False	Fire Order and Engagement	Yes	No
GNR_ReportedLast	A logical flag to indicate the CC has asked for and the GNR has reported the LAST LRF value	Boolean	False	Fire Order and Engagement	Yes	No
GNR_ReportedON	Logical flag that indicates the GNR has reported having the target in the sight image	Boolean	False	Fire Order and Engagement	Yes	No
GNR_ReportedRange	A string variable containing the distance that the GNR has reported to the CC	String		Fire Order and Engagement	Yes	No
GNR_ReportShotFall	A string variable that gives the GNR report of the last burst assessment INCOMPLETE. REQUIRES INPUT FROM ICGS AND LOGIC DESCRIBING THE FALL OF SHOT	String		Fire Order and Engagement	Yes	Yes
GNR_StartMode	Logical variable that indicates whether the GNR has configured their side of the turret to the default Start Mode	Boolean	False	Action	Yes	No
GNR_Tag	A tag variable to control the network execution	Integer	2	All	No	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	Value is 2 DO NOT CHANGE					
GNR_Tag_Lase	A special tag value to control the TargetLased activity of the GNR	Integer	21	Fire Order and Engagement	No	No
GNR_TargetDetecte d	A logical flag indicating that the GNR has detected a target	Boolean	False	Fire Order and Engagement	No	No
GNR_TargetInFOV	A logical indicating that an active target is in the GNRs sight FOV	Boolean	False	Fire Order and Engagement	No	Yes
GNR_TargetType	A string variable that records the type of vehicle the GNR thinks they have detected in a GNR initiated engagement	String		Fire Order and Engagement	Maybe	Maybe
GNR_TgtLOS	Logical variable indicating that the GNR has an unobstructed line of sight to the target for the engagement	Boolean	False	Fire Order and Engagement	No	Yes
GNR_ThermalSight Mode	Sight FOV setting Generic for now but adapt to different sights Values are: • WIDE • NARROW • 2xNARROW	Integer	0	• Action • Fire Order and Engagement	No	Yes
GNR_ThermalSight Power	• ON == true • OFF == false	Boolean	False	• Action • Turret Power Up • Turret Power Down	No	Yes
GREEN	An index for the flag variable. Alternatives are RED	Integer	1	Misfire	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
GunArmed_IndicatorLight	When the electronic control panel is armed, the GunArmed_IndicatorLight value should be true. If the electronic control panel is not armed, this is an electronic lockout for the weapon system and it cannot fire; GunArmed_IndicatorLight should be false	Boolean	True	Misfire	No	Yes
GunArmed_Switch	This is the electronic safety switch on the CDA that toggles the weapon system ON/OFF. It is integrated with the GunArmed_IndicatorLight, which lights up when the weapon system is armed	Boolean	False	• Action • Fire Order and Engagement • Misfire	No	Yes
GunFiringRate_CC	This is the state of the CC's firing mode switch on the CC's PDA: • SINGLE = 1 • 100 • 200	Integer	0	• Start Mode • Misfire	No	Yes
GunFiringRate_GNR	This is the state of the GNR's firing mode switch on the GNR's PDA: • SINGLE = 1 • 100 • 200	Integer	0	• Action • Misfire • Start Mode	No	Yes
HBR_CC	A flag that indicates whether the CC is represented by the model or is a person: • Simulated set to 'true' • Person set to 'false'	Boolean	True	All	No	No
HBR_DRV	A flag that indicates whether the DRV is represented by the model or is a person • simulated set to 'true'	Boolean	True	All	No	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	- person set to 'false'					
HBR_GNR	A flag to indicate that this crewmember is not present and is to be represented by a HBR operator - If a human crewmember present, set to false - If using the model for the operator, set to true	Boolean	True	All	No	No
HBR_Sentry	A flag that indicates whether the Sentry is represented by the model or is a person: - Simulated set to 'true' - Person set to 'false'	Boolean	True	All	No	No
ImageIntensifier	The CC has a low light (night vision goggle) sight that they can use; not the GNR? - ON == true - OFF == false	Boolean	False	- Turret Power Up - Turret Power Down	No	Yes
IR	A mode for the CCTD_Video display corresponding to IRTAS	Integer	1	Start Mode	Yes	No
IR	A mode for the CCTD_Video display corresponding to IRTAS	Integer	1	Start Mode	No	Yes
KitSecure	A logical variable used to indicate that the crew's kit is secure	Boolean	True	Opening Up	No	No
LaserPower	A flag indicating whether the laser power is: - ON == true - OFF == false	Boolean	False	- Action - Opening Up - Prove weapons - Turret Power Up - Turret Power	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
				Down		
LaseTgt	A network control flag that indicates the GNR should lase the target	Boolean	False	- Fire Order and Engagement	No	No
LAST	An int variable corresponding to the second or farthest range returned by the LRF when a DOUBLE return occurs Used as an index for the LRF_Range array to select the LAST (second) distance return from the LRF and to set the corresponding value of the LRF_FirstLastSwitch	Integer	2	- Action - Fire Order and Engagement - Start	No	
LAV_6	Switch to indicate VehicleType. DO NOT CHANGE THE VALUE	Integer	6	- Action - Fire Order and Engagement	No	No
LAV_AimPoint	This is the point of aim of the LAV weapons system. It must correspond to the Target_CoM for a valid firing solution. I expect that this will be horizontal and vertical angles of the Target CoM relative to the LAV. This is different from the AimPoint logical variable that indicates whether the aim point is on target	FloatingPoint	0	- Fire Order and Engagement - Misfire	No	Yes
LAV_III	Switch to indicate VehicleType. DO NOT CHANGE THE VALUE	Integer	3	- Action - Fire Order and Engagement	No	No
LAV_Running	A variable that indicates the LAV engine is running and power is available to run the systems and charge the batteries. If the LAV engine is running, this var should be true; if the engine is not running (or the electrical power circuit breakers are open???) this var	Boolean	False	- Misfire	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	should be false					
LAV_Type	An integer flag indicating the type of weapon system. Valid entries are: • LAV III == 3 (LAV_III) • LAV 6.0 == 6 (LAV_6)	Integer	3	• Action • Fire Order and Engagement • Turret Make Safe	No	Yes
LOCKED	A var used to set the state of a hatch opposite to OPEN, CLOSED	Integer	2	Action	No	No
LowAmmo_IndicatorLight	Indicator light on the CDA that shows the number of rounds left for the selected weapon is low	Boolean	False	Misfire	Maybe	Yes
LowBattery_IndicatorLight	The battery indicator on the CDA lights up if the battery is getting low and needs to be recharged from the engine generator	Boolean	False	Misfire	Maybe	Yes
LRF_Attempts	A counter to record how many lases of the target have been made to establish the range	Integer	0	Fire Order and Engagement	No	No
LRF_Double	A logical flag that indicated the LRF has received a DOUBLE return (i.e., two values)	Boolean	False	Fire Order and Engagement	No	Yes
LRF_FirstLastSwitch	This is the position of the LRF First/Last switch that selects the return that will be used by the computer or the GNR: • FIRST = 1 • LAST = 2	Integer	0	• Action • Fire Order and Engagement • Start Mode	No	Yes
LRF_Range	The range to the target returned by the LRF: • UNDEFINED == 0 • FIRST == 1 • LAST == 2. This contains the range values returned from a	Integer	0	Fire Order and Engagement	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	LASE and each value is selected when the FirstLastSwitch is toggled between FIRST and LAST					
LRF_Return	A logical variable indicating someone has used the LRF and measured the distance to the tgt	Boolean	False	Fire Order and Engagement	No	Yes
LWRPower	LWR power status: • ON == true • OFF == false	Boolean	False	• Turret Power Up • Turret Power Down	No	Yes
M B G Ds LOADED	Reports by the CC and GNR indicating there are MBGDs loaded in the grenade launchers					
MBGD_InUse	Logical expression to control loading of the MBGDs	Boolean	False	Action	No	Maybe
MBGD_LEFT	Flag indicating that the MBGDs are loaded on the LHS of the turret	Boolean	False	• Action • Prove weapons	No	Yes
MBGD_LEFT_Proven	A logical variable indicating the state of the left-hand side MBGDs was established	Boolean	False	• Prove weapons	No	Yes
MBGD_RIGHT	Flag indicating that the MBGDs are loaded on the RHS of the turret	Boolean	False	• Action • Prove weapons	No	Yes
MBGD_RIGHT_Proven	A logical variable indicating that the state of the RHS grenade launchers has been established	Boolean	False	Prove weapons	No	No
MBGD_Status	An integer indicating the state of the MBGD readiness to fire: • SAFE == 0 • Ready to fire Left == 1 • Ready to fire Right == 2	Integer	0	• Action • Opening Up • Misfire	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	• Ready to fire Both == 3					
MEN	A target type	Integer	5	Fire Order and Engagement	Yes	No
MEN_IN_OPEN	A target type	Integer	6	Fire Order and Engagement	Yes	No
MEN_IN_TRENCHES	A target type	Integer	7	Fire Order and Engagement	Yes	No
MG_Proven	A logical flag indicating the state of the pintle mount MG has been established (usually CLEAR?)	Boolean	False	Prove weapons	No	No
MG_Safety	Status of the pintle-mount MG safety: • ON == true • OFF == false (fire)	Boolean	False	Turret Make Safe	No	Yes
MISFIRE	A logical flag that indicates the weapon system has misfired. May be triggered by either the cannon or the COAX. Used to initiate the misfire drill	Boolean	False	• Action • Misfire	No	Yes
Misfire_InitialBurst	Used with CannonMisfire to indicate status of weapon DO NOT CHANGE VALUE FROM 1	Integer	1	Misfire	No	Maybe
Misfire_SubsequentBurst	Used with CannonMisfire to indicate status of misfire DO NOT CHANGE VALUE FROM 2	Integer	2	Misfire	No	Maybe
MisfireButton	Status of the Misfire button: • Clear = OFF = 0	Boolean	False	Action	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	• Misfire = ON = 1					
MisfireDrillActive	A logical variable used to indicate that the Misfire Drill is already active	Boolean	False	Misfire	No	No
MisfireDrillCount	Integer variable used to count how many times the misfire drill has been performed in the current engagement sequence. When the gun fires normally, the MisfireDrillCount should be reset to 0	Integer	0	Misfire	No	No
MoveTime	A variable that temporarily holds the calculated time for movement. Used in the meantime field	FloatingPoint	0	All	No	No
NARROW	FOV for sight settings. Options are: • WIDE • NARROW • NARROWx2	Integer	1	Fire Order and Engagement	No	No
NARROWx2	FOV for sight settings options are: • WIDE • NARROW • NARROWx2	Integer	0	• Fire Order and Engagement • Start Mode	Maybe	Maybe
NO_SOLUTION	GNR reports that the LRF has not returned a viable firing solution			Fire Order and Engagement	Yes	Maybe
OFF	A logical state of a device. Opposite of ON	Boolean	False	All		
ON	A logical state of a device. Opposite of OFF	Boolean	True	All	Yes	No
OPEN	A variable used to set the status of a hatch. Opposite to CLOSED, LOCKED	Integer	0	Action	No	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
OPERATIONAL	An integer used with flag that indicates the state of a weapon. Set to 0. DO NOT CHANGE	Integer	0	Misfire	No	No
ORDER_CrewAction	CC's order to perform the crew action drill	Boolean	False	Action	Yes	No
PerformStartDrill	A logical expression that indicates an engagement has finished and the turret crew should perform the start drill	Boolean	False	Fire Order and Engagement	No	No
pintleMG_Present	Logical: is the pintle mount MG (present (true) or is there none mounted (false)	Boolean	False	• Action • Misfire • Prove weapons	No	Yes
pintleMG_Status	Ready state for the pintle-mount MG: • READY = true • SAFE = false	Boolean	False	Misfire	No	Yes
PowerModeSwitch	A switch on the GNR's panel that ...	Boolean	False	Opening Up	No	Yes
PreviousAmmoBin	An integer variable that corresponds to the previous ammo bin that was used during an engagement. That will determine what type of round is in the gun and whether the ballistics need to be ignored. Valid values for ammo bins are: • PRIMARY (1) • SECONDARY (2) A value of UNLOADED (0) might be used for a downloaded gun.	Integer	0	Fire Order and Engagement	No	No
PRIMARY_BIN	An index into the ammunition arrays that	Integer	1	• Action	No	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	corresponds to the Primary ammo bin. DO NOT CHANGE			• Fire Order and Engagement • Misfire		
RadioPower	• ON == true • OFF == false	Boolean	False	• Turret Power Up • Turret Power Down	No	Yes
RangeConfirmation	A string variable that corresponds to the CC's assessment of whether the GNR's reported LRF return is good or requires further action. Used in the CC's Confirm Range network to control execution flow	String	Waiting	Fire Order and Engagement	Maybe	No
READY	Logical constant indicating status of equipment. Opposite is SAFE	Boolean	True	Misfire	Yes	No
RED	Flag state. Alternatives are GREEN	Integer	0	Misfire	No	Yes
REPORT_CDAPowerOff	A flag to control CC task execution, waiting for the GNR to report that they have turned the CDA power off	Boolean	False	Turret Power Down	Yes	Yes
REPORT_DAGROff	A report by the CC indicating that they have turned the DAGR off SHOULD THERE BE A COMPLEMENTARY REPORT ABOUT TURNING THE DAGR ON?	Boolean	False	Turret Power Down	Yes	No
REPORT_DriftNull	A logical variable indicating the GNR has turned on STAB and nulled turret drift; used as a control flag for the CC to do the next portion of the drill	Boolean	False	Turret Power Up	Yes	No
REPORT_LaserPower	A report from the GNR indicating a change in the laser power setting	Boolean	False	• Turret Power Up • Turret Power	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
				Down		
RightMBGD_Proven						
RoofHatchesLocked		Boolean	True			
SAFE	A flag that indicates the state of a weapon and used to set the weapon's status variable: • Opposite to ARMED • Opposite to READY	Integer	0	• Action • Misfire • Turret Make Safe	Yes	No
SEAR_IndicatorLight	This is a light on the CDA that indicates that the BPI is in SEAR. It is not the mechanical BPI up on the breech block, but presumably reflects that state. If this indicator is lit, its value is TRUE; if it is not lit, its value is FALSE	Boolean	False	Misfire	No	Yes
SECONDARY_BIN	An index into the ammunition arrays that corresponds to the Secondary ammo bin DO NOT CHANGE	Integer	2	• Action • Fire Order and Engagement • Misfire	No	No
SecondRound	A logical variable used to flag when the ordered ammo is different from the loaded ammo in the gun and that the CC should ignore the first-round ballistics as the ammo belts switch over	Boolean	False	Fire Order and Engagement	Yes	No
Sentry_Hatch	The rear sentry hatch status	Boolean	False	Action	No	Yes
SINGLE	A variable used to indicate single shot mode for the 25mm cannon. Set to 1. Alternatives are 100 and 200	Integer	1	• Action • Misfire	Maybe	No
SpeechTime	A variable that temporarily holds the calculated time for a spoken report or order; used in the	FloatingPoint	0	All	No	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	meantime field					
StabPower	Gun stabilizer power state: • ON == true • OFF == false	Boolean	False	• Action • Opening Up • Turret Power Up	No	Yes
TacNavPower	• ON = true • OFF = false	Boolean	False	Action	No	Yes
TANK	A target type	Integer	3	Fire Order and Engagement	Yes	No
Target_CoM	This is the location of the Centre of Mass relative to the LAV and it must correspond to the Aim Point for a firing solution to be accepted. Note: Probably not appropriate for area firing! I'm expecting the values to be angles of horizontal and vertical	FloatingPoint	0	Fire Order and Engagement	No	Yes
TargetDestroyed	A logical variable that describes the actual status of the current target	Boolean	False	Fire Order and Engagement	No	Yes
TargetInSightBox	Target is in the WFOV sight targeting box; indicator from ICGS	Boolean	False	Fire Order and Engagement	No	Yes
TargetLased	Generic flag to control branching; indicates that someone is TargetLased the target	Boolean	False	Fire Order and Engagement	No	Yes
TargetQty	A count of the number of targets that have been detected	Integer	0	Fire Order and Engagement	No	Yes
TargetRange	Range to the target obtained from the ICGS	Integer	0	Fire Order and Engagement	No	Yes
TargetType	Initially just set to NUL but figure out what values the ICGS uses	Integer	0	Fire Order and Engagement	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	Options are set temporarily here: (get more types from the Gunnery Manual) • APC == 1 • BMP == 2 • TANK == 3 • TRANSPORT = 4 • MEN == 5 • MEN IN OPEN == 6 //no logic for this case • MEN IN TRENCHES == 7 //no logic for this case					
TempFault_Indicator Light	The state of the temperature fault indicator on the CDA	Boolean	False	Misfire	No	Yes
TextMessage	A string variable used to indicate a verbal message that the speech production system speaks. it is accessed through the SpeechProduction function. Used in association with TextName	String		All	No	No
TextVoice	A string variable that is used to hold the operator voice file name for the speech utility. Accessed through the SpeechProduction function	String		All	No	No
ThermalImager	• ON == true • OFF == false NEED TO CREATE TWO VERSIONS OF THIS: ONE FOR THE CC AND ONE FOR THE GNR	Boolean	False	• Opening Up • Turret Power Up • Turret Power Down	No	Yes
ThermalSight	Operating mode of the GNRs thermal sight:	Integer	0	Action	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
	• WIDE = 0 //default • NARROW = 2					
ThermalSightPower	• ON == true • OFF == false	Boolean	False	Action	No	Yes
TrackAndBolt_Mounted	A flag that indicates the cannon action is present	Boolean	False	Prove weapons	No	Yes
TRANSPORT	A target type	Integer	4	Fire Order and Engagement	Yes	No
TURRET_AuxPower	GNRs control of the auxiliary power: • ON = true • OFF = false	Boolean	False	• Action • Turret Power Up • Turret Power Down	No	Yes
TurretLock	State of the turret lock: • ON == true • OFF == false	Boolean	False	• Action • Opening Up • Turret Power Down	No	Yes
TurretPower	Logical status of turret (i.e., Power ON/OFF). Assume power is off by default as standard operating procedure states: Turret power will be OFF when entering or exiting the vehicle. • ON == true • OFF == false	Boolean	False	• Action • Opening Up • Prove weapons • Turret Power Up • Turret Power Down	No	Yes
UNKNOWN	A target type	Integer	0	Fire Order and Engagement	Yes	No

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
UNLOADED	An index for the ammunition array corresponding to the first entry or no ammunition loaded. DO NOT CHANGE	Integer	0	• Fire Order and Engagement • Misfire	No	No
UseHBR2Plugin	A logical flag that enables the speech recognition and production plugin to work with the MS SAPI. This should be set to true to enable speech. Not sure what the implications are if it is set to false (other than not making use of MS SAPI)	Boolean	True	All	Yes	No
UseHBR2Plugin	A logical flag that enables the speech recognition and production plugin to work with the MS SAPI. This should be set to true to enable speech	Boolean	True	All	Yes	No
WEAPON_Cannon	A variable used to set WEAPON_Selected indicating the 25mm cannon is the active weapon. DO NOT CHANGE. Value is 1	Integer	1	• Action • Fire Order and Engagement	No	No
WEAPON_Coax	A variable used with WEAPON_Selected to indicate the 7.62mm COAX MG has been selected. DO NOT CHANGE Value is 2	Integer	2	• Action • Fire Order and Engagement	No	No
WEAPON_None	A variable used with WEAPON_Selected, indicating no weapon has been selected. (Probably indicating an error) DO NOT CHANGE	Integer	0	• Action • Fire Order and Engagement	No	No
WEAPON_Power	Weapon power: • ON == true • OFF == false	Boolean	False	• Action • Prove weapons • Turret Make	No	Yes

Variable Name	Description	Type	Initial Value	Source Model	Speech Model	Link to ICGS
				Safe • Turret Power Up • Turret Power Down		
WEAPON_Selected	A flag that indicates which weapon the GNR has selected on the palm controller: • None_Selected == 0 (probably an error) • CANNON_Selected == 1 (typical weapon) • COAX_Selected == 2	Integer	0	• Action • Fire Order and Engagement • Misfire	Maybe	Yes
WIDE	FOV for sight settings options are: • WIDE • NARROW • NARROWx2	Integer	0	• Fire Order and Engagement • Start	Maybe	Maybe
WPN_IndicatorLight	A variable that indicates the status of the weapon indicator light on the CDA: ON == true OFF == false	Boolean	False	Misfire	No	Yes

APPENDIX C SPEECH RECOGNITION FILES

This appendix provides a listing of the contents of the three speech recognition files used with the LAV crew model to implement the CC’s verbal interactions with the model. Sections of code beginning with “<!—“ and ending with “-->” are comments.

C.1.1 HBR2Blugin.dll.config

The following listing provides the association amongst the speech recognition variables and the corresponding IPME LAV crew model variables. In most cases, the variable names are the same. The association also defines the IPME variable type.

```
<?xml version="1.0" encoding="utf-8" ?>

<HBR2PluginSection xmlns="http://CAE.HBR2/HBR2Plugin.xsd"

          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

          xsi:schemaLocation="http://CAE.HBR2/HBR2Plugin.xsd HBR2Plugin.dll.xsd"

          LoadPropName="UseHBR2Plugin">

 <MapperConfig>

  <Incoming>

<!-- BEGIN is a temporary variable used to allow the user to adjust initial conditions before
executing the model directly -->

      <InMap InVariable="BEGIN" IPMEVariable="BEGIN" Type="bool" />

<!-- Miscillaneous variables; some may be unnecessary to keep in speech files -->

      <InMap InVariable="FO_Issued" IPMEVariable="FO_Issued" Type="bool" />

      <InMap InVariable="CCTD_VideoSwitch" IPMEVariable="CCTD_VideoSwitch"
Type="int" />

      <InMap InVariable="MG_Proven" IPMEVariable="MG_Proven" Type="bool" />

      <InMap InVariable="MBGD_RIGHT_Proven" IPMEVariable="MBGD_RIGHT_Proven"
Type="bool" />

      <InMap InVariable="CC_StartMode" IPMEVariable="CC_StartMode" Type="bool" />

      <InMap InVariable="CC_ORDER_RangeChange"
IPMEVariable="CC_ORDER_RangeChange" Type="bool" />
```

```
<!-- Crew Commander verbal orders and reports -->

    <!-- Fire order variables -->

        <InMap InVariable="CC_FO_AmmoType" IPMEVariable="CC_FO_AmmoType"
Type="string" />

        <InMap InVariable="CC_FO_Range" IPMEVariable="CC_FO_Range" Type="string" />

        <InMap InVariable="CC_FO_TargetType" IPMEVariable="CC_FO_TargetType"
Type="string" />



    <!-- Orders -->

        <InMap InVariable="CC_ORDER_200Ready" IPMEVariable="CC_ORDER_200Ready"
Type="bool" />

        <InMap InVariable="CC_ORDER_AuxiliaryOn"
IPMEVariable="CC_ORDER_AuxiliaryOn" Type="bool" />

        <InMap InVariable="CC_ORDER_CannonDepress"
IPMEVariable="CC_ORDER_CannonDepress" Type="bool" />

        <InMap InVariable="CC_ORDER_CannonElevate"
IPMEVariable="CC_ORDER_CannonElevate" Type="bool" />

        <InMap InVariable="CC_ORDER_CrewAction"
IPMEVariable="CC_ORDER_CrewAction" Type="bool" />

        <InMap InVariable="CC_ORDER_DriverStartVehicle"
IPMEVariable="CC_ORDER_DriverStartVehicle" Type="bool" />

        <InMap InVariable="CC_ORDER_Fire" IPMEVariable="CC_ORDER_Fire" Type="bool"
/>

        <InMap InVariable="CC_ORDER_FirstLast" IPMEVariable="CC_ORDER_FirstLast"
Type="int" />

        <InMap InVariable="CC_ORDER_GoOn" IPMEVariable="CC_ORDER_GoOn"
Type="bool" />

        <InMap InVariable="CC_ORDER_Lase" IPMEVariable="CC_ORDER_Lase"
Type="bool" />
```

```
        <InMap InVariable="CC_ORDER_LaserOn" IPMEVariable="CC_ORDER_LaserOn"
Type="bool" />

        <InMap InVariable="CC_ORDER_MisfireWait"
IPMEVariable="CC_ORDER_MisfireWait" Type="bool" />

        <InMap InVariable="CC_ORDER_NextTarget"
IPMEVariable="CC_ORDER_NextTarget" Type="bool" />

        <InMap InVariable="CC_ORDER_Ready" IPMEVariable="CC_ORDER_Ready"
Type="bool" />

        <InMap InVariable="CC_ORDER_Relase" IPMEVariable="CC_ORDER_Relase"
Type="bool" />

        <InMap InVariable="CC_ORDER_Reset" IPMEVariable="CC_ORDER_Reset"
Type="bool" />

        <InMap InVariable="CC_ORDER_SingleShot" IPMEVariable="CC_ORDER_SingleShot"
Type="bool" />

        <InMap InVariable="CC_ORDER_Stop" IPMEVariable="CC_ORDER_Stop" Type="bool"
/>

        <InMap InVariable="CC_ORDER_TargetStop"
IPMEVariable="CC_ORDER_TargetStop" Type="bool" />

        <InMap InVariable="CC_ORDER_TurretMakeSafe"
IPMEVariable="CC_ORDER_TurretMakeSafe" Type="bool" />

        <InMap InVariable="CC_ORDER_TurretOn" IPMEVariable="CC_ORDER_TurretOn"
Type="bool" />

        <InMap InVariable="CC_ORDER_TurretPowerDown"
IPMEVariable="CC_ORDER_TurretPowerDown" Type="bool" />

        <InMap InVariable="CC_ORDER_TurretPowerUp"
IPMEVariable="CC_ORDER_TurretPowerUp" Type="bool" />

        <InMap InVariable="CC_ORDER_WeaponsOn"
IPMEVariable="CC_ORDER_WeaponsOn" Type="bool" />

        <!-- Reports -->

        <InMap InVariable="CC_REPORT_CannonClear"
IPMEVariable="CC_REPORT_CannonClear" Type="bool" />
```

```
        <InMap InVariable="CC_REPORT_CannonSafe"
IPMEVariable="CC_REPORT_CannonSafe" Type="bool" />

        <InMap InVariable="CC_REPORT_CoaxClear"
IPMEVariable="CC_REPORT_CoaxClear" Type="bool" />

        <InMap InVariable="CC_REPORT_ComdOverrideOff"
IPMEVariable="CC_REPORT_ComdOverrideOff" Type="bool" />

        <InMap InVariable="CC_REPORT_CSAMOff" IPMEVariable="CC_REPORT_CSAMOff"
Type="bool" />

        <InMap InVariable="CC_REPORT_DAGROff" IPMEVariable="CC_REPORT_DAGROff"
Type="bool" />

        <InMap InVariable="CC_REPORT_FiringNow"
IPMEVariable="CC_REPORT_FiringNow" Type="bool" />

        <InMap InVariable="CC_REPORT_FreeToTraverse"
IPMEVariable="CC_REPORT_FreeToTraverse" Type="bool" />

        <InMap InVariable="CC_REPORT_IIOff" IPMEVariable="CC_REPORT_IIOff"
Type="bool" />

        <InMap InVariable="CC_REPORT_LWDOff" IPMEVariable="CC_REPORT_LWDOff"
Type="bool" />

        <InMap InVariable="CC_REPORT_LWSOff" IPMEVariable="CC_REPORT_LWSOff"
Type="bool" />

        <InMap InVariable="CC_REPORT_MGproven"
IPMEVariable="CC_REPORT_MGproven" Type="bool" />

        <InMap InVariable="CC_REPORT_ProlongedStoppage"
IPMEVariable="CC_REPORT_ProlongedStoppage" Type="bool" />

        <InMap InVariable="CC_REPORT_RadioOff" IPMEVariable="CC_REPORT_RadioOff"
Type="bool" />

        <InMap InVariable="CC_REPORT_RIGHTMBGDproven"
IPMEVariable="CC_REPORT_RIGHTMBGDproven" Type="bool" />

        <InMap InVariable="CC_REPORT_SEAR" IPMEVariable="CC_REPORT_SEAR"
Type="bool" />

        <InMap InVariable="CC_REPORT_SwitchesUp"
IPMEVariable="CC_REPORT_SwitchesUp" Type="bool" />
```

```
      <InMap InVariable="CC_REPORT_ThermalOff"
IPMEVariable="CC_REPORT_ThermalOff" Type="bool" />


   </Incoming>


   <IncomingMulti>

   <InMultiMap>

    <InVars>

     <Variable InVariable="CC_RightMBGDloaded" />

     <Variable InVariable="CC_LeftMBGDloaded" />

    </InVars>

    <OutVars>

     <Variable IPMEVariable="MBGD_Right" IPMEVariableValue="$.CC_RightMBGDloaded"
Type="bool"/>

     <Variable IPMEVariable="MBGD_LEFT" IPMEVariableValue="$.CC_LeftMBGDloaded"
Type="bool"/>

    </OutVars>

   </InMultiMap>

  </IncomingMulti>


 </MapperConfig>


 <SpeechCommandConfig  Enable="true" LangLocale="en-US">

  <Grammars>

   <Grammar>GunnerCommands.grxml</Grammar>

  </Grammars>
  </SpeechCommandConfig>


 <SpeechSynthesisConfig Enable="true" DriverVoiceName="Brian" GunnerVoiceName="Salli"
CommanderVoiceName="Joey">

 <!-- <SpeechSynthesisConfig Enable="true" DriverVoiceName="ray"
GunnerVoiceName="mike" CommanderVoiceName="crystal"> -->

    <Phrases>

      <Phrase Name="Speech text message" Voice="gunnerVoice">

        <Trigger>

          <SingleCondition>

            <VariableCondition Variable="TextMessage" Condition="Equal" Value="true"/>

          </SingleCondition>

        </Trigger>

        <Action>

          <ActionTemplate></ActionTemplate>

        </Action>

      </Phrase>

    </Phrases>

  </SpeechSynthesisConfig>

</HBR2PluginSection>
```

C.1.2 GunnerCommands.grxml

This section provides a listing of the main grammar file used for speech recognition. It is organized similarly to HBR2Plugin.dll.config (Section C.1.1); however, there are two sections to the file. The first section defines relationships used by the speech recognition system to assign values to variables. The second section defines the verbal expression and some phonetic variations of those expressions that are used to identify relevant phrases of speech.

```
<?xml version="1.0" encoding="utf-8" ?>

<grammar version="1.0" xml:lang="en-US"

      tag-format="semantics-ms/1.0"

      root="GunnerAndLoaderCommands"

      mode="voice"

      xmlns="http://www.w3.org/2001/06/grammar"

      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

      xsi:schemaLocation="http://www.w3.org/2001/06/grammar

                  http://www.w3.org/TR/speech-grammar/grammar.xsd"

      xmlns:sapi="http://schemas.microsoft.com/Speech/2002/06/SRGSExtensions">


 <rule id="GunnerAndLoaderCommands" scope="public">

  <one-of>

 <!-- BEGIN is a temporary variable used to allow the user to adjust initial conditions before
executing the model directly -->

       <item>

           <ruleref uri="#BEGIN" />

           <tag> $.Command = 'CC.BEGIN'; </tag>

           <tag> $.BEGIN = 'true'; </tag>

       </item>


       <!-- Miscillaneous variables; some may be unnecessary to keep in speech files -->


       <item>

           <ruleref uri="#FO_Issued" />
```

```
        <tag> $.Command = 'CC.FO_Issued'; </tag>
        <tag> $.FO_Issued = 'true'; </tag>
    </item>

    <item>
        <ruleref uri="#CCTD_VideoSwitch" />
        <tag> $.Command = 'CC.CCTD_VideoSwitch'; </tag>
        <tag> $.CCTD_VideoSwitch = '1'; </tag>
    </item>

    <item>
        <ruleref uri="#GNR_StartMode" />
        <tag> $.Command = 'GNR.GNR_StartMode'; </tag>
        <tag> $.GNR_StartMode = 'true'; </tag>
    </item>

    <item>
        <ruleref uri="#MG_Proven" />
        <tag> $.Command = 'CC.MG_Proven'; </tag>
        <tag> $.MG_Proven = $$; </tag>
    </item>

    <item>
        <ruleref uri="#MBGD_RIGHT_Proven" />
        <tag> $.Command = 'CC.MBGD_RIGHT_Proven'; </tag>

        <tag> $.MBGD_RIGHT_Proven = $$; </tag>

    </item>


    <item>

        <ruleref uri="#CC_StartMode" />

        <tag> $.Command = 'CC.CC_StartMode'; </tag>

        <tag> $.CC_StartMode = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_RangeChange" />

        <tag> $.Command = 'CC.CC_ORDER_RangeChange'; </tag>

        <tag> $.CC_ORDER_RangeChange = $$; </tag>

    </item>


    <!-- Crew Commander verbal orders and reports -->


    <!-- Fire order variables -->

    <item>

        <ruleref uri="#CC_FO_AmmoType" />

        <tag> $.Command = 'CC.CC_FO_AmmoType'; </tag>

        <tag> $.CC_FO_AmmoType = $$.ammoType; </tag>

    </item>
```

```
<item>
    <ruleref uri="#CC_FO_Range" />
    <tag> $.Command = 'CC.CC_FO_Range'; </tag>
    <tag> $.CC_FO_Range = $$; </tag>
</item>


<item>
    <ruleref uri="#CC_FO_TargetType" />
    <tag> $.Command = 'CC.CC_FO_TargetType'; </tag>
    <tag> $.CC_FO_TargetType = $$; </tag>
</item>



<!-- Orders -->
<item>
    <ruleref uri="#CC_ORDER_200Ready" />
    <tag> $.Command = 'CC.CC_ORDER_200Ready'; </tag>
    <tag> $.CC_ORDER_200Ready = 'true'; </tag>
</item>


<item>
    <ruleref uri="#CC_ORDER_AuxiliaryOn" />
    <tag> $.Command = 'CC.CC_ORDER_AuxiliaryOn'; </tag>
```

```
        <tag> $.CC_ORDER_AuxiliaryOn = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_CannonDepress" />

        <tag> $.Command = 'CC.CC_ORDER_CannonDepress'; </tag>

        <tag> $.CC_ORDER_CannonDepress = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_CannonElevate" />

        <tag> $.Command = 'CC.CC_ORDER_CannonElevate'; </tag>

        <tag> $.CC_ORDER_CannonElevate = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_CrewAction" />

        <tag> $.Command = 'CC.CC_ORDER_CrewAction'; </tag>

        <tag> $.CC_ORDER_CrewAction = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_DriverStartVehicle" />

        <tag> $.Command = 'CC.CC_ORDER_DriverStartVehicle'; </tag>
        <tag> $.CC_ORDER_DriverStartVehicle = $$; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_FirstLast" />

        <tag> $.Command = 'CC.CC_ORDER_FirstLast'; </tag>

        <tag> $.CC_ORDER_FirstLast = $$; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_Fire" />

        <tag> $.Command = 'CC.CC_ORDER_Fire'; </tag>

        <tag> $.CC_ORDER_Fire = $$; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_GoOn" />

        <tag> $.Command = 'CC.CC_ORDER_GoOn'; </tag>

        <tag> $.CC_ORDER_GoOn = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_Lase" />

        <tag> $.Command = 'CC.CC_ORDER_Lase'; </tag>
        <tag> $.CC_ORDER_Lase = $$; </tag>
    </item>

    <item>
        <ruleref uri="#CC_ORDER_LaserOn" />
        <tag> $.Command = 'CC.CC_ORDER_LaserOn'; </tag>
        <tag> $.CC_ORDER_LaserOn = $$; </tag>
    </item>

    <item>
        <ruleref uri="#CC_ORDER_MisfireWait" />
        <tag> $.Command = 'CC.CC_ORDER_MisfireWait'; </tag>
        <tag> $.CC_ORDER_MisfireWait = 'true'; </tag>
    </item>

    <item>
        <ruleref uri="#CC_ORDER_NextTarget" />
        <tag> $.Command = 'CC.CC_ORDER_NextTarget'; </tag>
        <tag> $.CC_ORDER_NextTarget = 'true'; </tag>
    </item>

    <item>
        <ruleref uri="#CC_ORDER_Ready" />
        <tag> $.Command = 'CC.CC_ORDER_Ready'; </tag>
        <tag> $.CC_ORDER_Ready = $$; </tag>
    </item>

    <item>
        <ruleref uri="#CC_ORDER_Relase" />
        <tag> $.Command = 'CC.CC_ORDER_Relase'; </tag>
        <tag> $.CC_ORDER_Relase = $$; </tag>
    </item>

    <item>
        <ruleref uri="#CC_ORDER_Reset" />
        <tag> $.Command = 'CC.CC_ORDER_Reset'; </tag>
        <tag> $.CC_ORDER_Reset = 'true'; </tag>
    </item>

    <item>
        <ruleref uri="#CC_ORDER_SingleShot" />
        <tag> $.Command = 'CC.CC_ORDER_SingleShot'; </tag>
        <tag> $.CC_ORDER_SingleShot = $$; </tag>
    </item>

    <item>
        <ruleref uri="#CC_ORDER_Stop" />
        <tag> $.Command = 'CC.CC_ORDER_Stop'; </tag>
        <tag> $.CC_ORDER_Stop = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_TargetStop" />

        <tag> $.Command = 'CC.CC_ORDER_TargetStop'; </tag>

        <tag> $.CC_ORDER_TargetStop = $$; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_TurretMakeSafe" />

        <tag> $.Command = 'CC.CC_ORDER_TurretMakeSafe'; </tag>

        <tag> $.CC_ORDER_TurretMakeSafe = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_TurretOn" />

        <tag> $.Command = 'CC.CC_ORDER_TurretOn'; </tag>

        <tag> $.CC_ORDER_TurretOn = 'true'; </tag>

    </item>


    <item>

        <ruleref uri="#CC_ORDER_TurretPowerDown" />

        <tag> $.Command = 'CC.CC_ORDER_TurretPowerDown'; </tag>
        <tag> $.CC_ORDER_TurretPowerDown = $$; </tag>
    </item>


    <item>
        <ruleref uri="#CC_ORDER_TurretPowerUp" />
        <tag> $.Command = 'CC.CC_ORDER_TurretPowerUp'; </tag>
        <tag> $.CC_ORDER_TurretPowerUp = $$; </tag>
    </item>


    <item>
        <ruleref uri="#CC_ORDER_WeaponsOn" />
        <tag> $.Command = 'CC.CC_ORDER_WeaponsOn'; </tag>
        <tag> $.CC_ORDER_WeaponsOn = 'true'; </tag>
    </item>



    <!-- Reports -->


    <item>
        <ruleref uri="#CC_REPORT_CannonClear" />
        <tag> $.Command = 'CC.CC_REPORT_CannonClear'; </tag>
        <tag> $.CC_REPORT_CannonClear = $$; </tag>
    </item>
```

```
<item>
    <ruleref uri="#CC_REPORT_CannonSafe" />
    <tag> $.Command = 'CC.CC_REPORT_CannonSafe'; </tag>
    <tag> $.CC_REPORT_CannonSafe = 'true'; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_CoaxClear" />
    <tag> $.Command = 'CC.CC_REPORT_CoaxClear'; </tag>
    <tag> $.CC_REPORT_CoaxClear = $$; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_ComdOverrideOff" />
    <tag> $.Command = 'CC.CC_REPORT_ComdOverrideOff'; </tag>
    <tag> $.CC_REPORT_ComdOverrideOff = $$; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_CSAMOff" />
    <tag> $.Command = 'CC.CC_REPORT_CSAMOff'; </tag>
    <tag> $.CC_REPORT_CSAMOff = $$; </tag>
</item>
```

```
<item>
    <ruleref uri="#CC_REPORT_DAGROff" />
    <tag> $.Command = 'CC.CC_REPORT_DAGROff'; </tag>
    <tag> $.CC_REPORT_DAGROff = $$; </tag>
</item>


<item>
    <ruleref uri="#CC_REPORT_FiringNow" />
    <tag> $.Command = 'CC.CC_REPORT_FiringNow'; </tag>
    <tag> $.CC_REPORT_FiringNow = $$; </tag>
</item>


<item>
    <ruleref uri="#CC_REPORT_FreeToTraverse" />
    <tag> $.Command = 'CC.CC_REPORT_FreeToTraverse'; </tag>
    <tag> $.CC_REPORT_FreeToTraverse = $$; </tag>
</item>


<item>
    <ruleref uri="#CC_REPORT_IIOff" />
    <tag> $.Command = 'CC.CC_REPORT_IIOff'; </tag>
    <tag> $.CC_REPORT_IIOff = $$; </tag>
</item>
```

```
<item>
    <ruleref uri="#CC_REPORT_LWDOff" />
    <tag> $.Command = 'CC.CC_REPORT_LWDOff'; </tag>
    <tag> $.CC_REPORT_LWDOff = $$; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_LWSOff" />
    <tag> $.Command = 'CC.CC_REPORT_LWSOff'; </tag>
    <tag> $.CC_REPORT_LWSOff = $$; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_MGproven" />
    <tag> $.Command = 'CC.CC_REPORT_MGproven'; </tag>
    <tag> $.CC_REPORT_MGproven = $$; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_ProlongedStoppage" />
    <tag> $.Command = 'CC.CC_REPORT_ProlongedStoppage'; </tag>
    <tag> $.CC_REPORT_ProlongedStoppage = 'true'; </tag>
</item>
```

```
<item>
    <ruleref uri="#CC_REPORT_RadioOff" />
    <tag> $.Command = 'CC.CC_REPORT_RadioOff'; </tag>
    <tag> $.CC_REPORT_RadioOff = $$; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_RIGHTMBGDproven" />
    <tag> $.Command = 'CC.CC_REPORT_RIGHTMBGDproven'; </tag>
    <tag> $.CC_REPORT_RIGHTMBGDproven = $$; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_SEAR" />
    <tag> $.Command = 'CC.CC_REPORT_SEAR'; </tag>
    <tag> $.CC_REPORT_SEAR = 'true'; </tag>
</item>

<item>
    <ruleref uri="#CC_REPORT_SwitchesUp" />
    <tag> $.Command = 'CC.CC_REPORT_SwitchesUp'; </tag>
    <tag> $.CC_REPORT_SwitchesUp = 'true'; </tag>
</item>
```

```
        <item>
            <ruleref uri="#CC_REPORT_ThermalOff" />
            <tag> $.Command = 'CC.CC_REPORT_ThermalOff'; </tag>
            <tag> $.CC_REPORT_ThermalOff = $$; </tag>
        </item>


    </one-of>
  </rule>



<!-- Recognition rule logic -->


        <!-- BEGIN is a temporary variable used to allow the user to adjust initial conditions
before executing the model directly -->
    <rule id="BEGIN" scope="private">
        <example> BEGIN </example>
        <item> begin </item>
    </rule>


        <!-- Miscillaneous variables; some may be unnecessary to keep in speech files -->


<!-- FO_Issued is temporarily linked to the CC's request for GNR feedback that the target is in
the sight box -->
    <rule id="FO_Issued" scope="private">
        <example> on </example>
    <item> on </item>
</rule>

<rule id="GNR_StartMode" scope="private">
    <example> START MODE </example>
    <item> start </item>
    <item> mode </item>
</rule>

<rule id="CCTD_VideoSwitch" scope="private">
    <example> IR ?</example>
    <item> I </item>
    <item> R </item>
</rule>

<rule id="MG_Proven" scope="private">
    <one-of>
        <item> M G clear <tag> $="true"; </tag> </item>
        <item> M gee clear <tag> $="true"; </tag> </item>
        <item> M G not mounted <tag> $="true"; </tag> </item>
        <item> M gee not mounted <tag> $="true"; </tag> </item>
    </one-of>
</rule>
```

```
    <rule id="MBGD_RIGHT_Proven" scope="private">
        <one-of>
            <item> right M B G D clear <tag> $="true"; </tag> </item>
            <item> right M bee gee dee clear <tag> $="true"; </tag> </item>
            <item> right M B G D loaded <tag> $="true"; </tag> </item>
            <item> right M bee gee dee loaded <tag> $="true"; </tag> </item>
        </one-of>
    </rule>

    <rule id="CC_StartMode" scope="private">
        <example> START MODE </example>
        <item> start </item>
        <item> mode </item>
    </rule>

<!-- I think that CC_ORDER_RangeChange is not needed but I need to check the model -->
    <rule id="CC_ORDER_RangeChange" scope="private">
        <item> range change <tag> $="true"; </tag> </item>
    </rule>

<!-- Crew Commander verbal orders and reports -->

    <!-- Fire order variables -->
    <rule id="CC_FO_AmmoType" scope="private">
```

```
        <one-of>

            <item> coax <tag> $.ammoType="COAX"; </tag> </item>

            <item> co axe <tag> $.ammoType="COAX"; </tag> </item>

            <item> frang <tag> $.ammoType="FRANG"; </tag> </item>

            <item> frange <tag> $.ammoType="FRANG"; </tag> </item>

            <item> H E <tag> $.ammoType="HE"; </tag> </item>

            <item> sabot <tag> $.ammoType="SABOT"; </tag> </item>

            <item> saw bow <tag> $.ammoType="SABOT"; </tag> </item>

        </one-of>

    </rule>


<!-- The CC_FO_Range rule does not currently work; this is needed for the fire order and
engagement drill -->

        <rule id="CC_FO_Range" scope="private">

        <example> BATTLE; 1 400 for 1400; 2000 for 2 thousand; 4 300 for 4300 </example>

        <one-of>

        <item> battle <tag> $="BATTLE";</tag> </item>

        <item>

            <ruleref uri="Number.grxml#DIGITS" />

            <tag>$=$$;</tag>

        </item>

        </one-of>

    </rule>


    <rule id="CC_FO_TargetType" scope="private">
```

```
        <one-of>
            <item> A P C <tag> $="APC"; </tag> </item>
            <item> B M P <tag> $="APC"; </tag> </item> <!-- a BMP is now officially designated
as an APC -->
            <item> BMP <tag> $="APC"; </tag> </item>
            <item> tank <tag> $="TANK"; </tag> </item>
            <item> transport <tag> $="TRANSPORT"; </tag> </item>
            <item> men <tag> $="MEN"; </tag> </item>
            <item> men in open<tag> $="MEN IN OPEN"; </tag> </item>
            <item> men in trenches<tag> $="MEN IN TRENCHES"; </tag> </item>
            <item> unknown <tag> $="UNKNOWN"; </tag> </item>
        </one-of>
    </rule>


        <!-- Orders -->


    <rule id="CC_ORDER_200Ready" scope="private">
        <example> 200 READY</example>
        <item> Two </item>
        <item> Hundred </item>
        <item> Ready </item>
    </rule>


    <rule id="CC_ORDER_AuxiliaryOn" scope="private">
        <example> auxiliary on </example>
        <item> auxiliary </item>
        <item> on </item>
    </rule>

    <rule id="CC_ORDER_CannonDepress" scope="private">
        <example> Depress </example>
        <one-of>
            <item> depress </item>
            <item> cannon depress </item>
            <item> depress gun </item>
        </one-of>
    </rule>

    <rule id="CC_ORDER_CannonElevate" scope="private">
        <example> ELEVATE </example>
        <item> elevate </item>
    </rule>

    <rule id="CC_ORDER_CrewAction" scope="private">
        <example> CREW ACTION </example>
        <item> crew </item>
        <item> action </item>
    </rule>
```

```
<rule id="CC_ORDER_DriverStartVehicle" scope="private">

    <one-of>

        <item> driver start vehicle <tag>$="true";</tag> </item>

        <item> driver start your vehicle <tag>$="true";</tag> </item>

        <item> low battery start vehicle <tag>$="true";</tag> </item>

        <item> low battery driver start vehicle <tag>$="true";</tag> </item>

        <item> low battery driver start your vehicle <tag>$="true";</tag> </item>

    </one-of>

</rule>


<rule id="CC_ORDER_Fire" scope="private">

    <item> fire <tag> $="true"; </tag> </item>

</rule>


<rule id="CC_ORDER_FirstLast" scope="private">

    <example> LAST </example>

    <one-of>

      <item> first <tag> $="1"; </tag> </item>

      <item> last <tag> $="2"; </tag> </item>

    </one-of>

</rule>


<rule id="CC_ORDER_GoOn" scope="private">

    <example> go on </example>
    <item> go </item>

    <item> on </item>

</rule>


<rule id="CC_ORDER_Lase" scope="private">

    <example> LASE (lays) </example>

    <one-of>

      <item> lase <tag> $="true"; </tag> </item>

      <item> lays <tag> $="true"; </tag> </item>

    </one-of>

</rule>


<rule id="CC_ORDER_LaserOn" scope="private">

    <example> LASER ON </example>

    <one-of>

        <item> laser on <tag> $="true"; </tag> </item>

        <item> lays her on <tag> $="true"; </tag> </item>

    </one-of>

</rule>


<rule id="CC_ORDER_MisfireWait" scope="private">

    <example> MISFIRE WAIT </example>

    <item> misfire </item>

    <item> wait </item>
```

```
</rule>


<rule id="CC_ORDER_NextTarget" scope="private">
    <one-of>
        <item> next target <tag> $="true"; </tag> </item>
        <item> stop next target <tag> $="true"; </tag> </item>
    </one-of>
</rule>


<rule id="CC_ORDER_Ready" scope="private">
    <one-of>
        <item> ready <tag> $="true"; </tag> </item>
        <item> ready cannon <tag> $="true"; </tag> </item>
    </one-of>
</rule>


<rule id="CC_ORDER_Relase" scope="private">
    <one-of>
        <item> relase <tag> $="true"; </tag> </item>
        <item> re lase <tag> $="true"; </tag> </item>
        <item> relays <tag> $="true"; </tag> </item>
    </one-of>
</rule>
```

```
<rule id="CC_ORDER_Reset" scope="private">

    <example> RESET </example>

    <item> reset </item>

</rule>


<rule id="CC_ORDER_SingleShot" scope="private">

    <one-of>

        <item> single <tag> $="true"; </tag> </item>

        <item> single shot <tag> $="true"; </tag> </item>

    </one-of>

</rule>


<rule id="CC_ORDER_Stop" scope="private">

    <example> stop </example>

    <item> stop </item>

</rule>


<rule id="CC_ORDER_TargetStop" scope="private">

    <item> target stop <tag> $="true"; </tag> </item>

</rule>


<rule id="CC_ORDER_TurretMakeSafe" scope="private">

    <example> TURRET MAKE SAFE  </example>

    <item> turret </item>
    <item> make </item>
    <item> safe </item>
</rule>

<rule id="CC_ORDER_TurretOn" scope="private">
    <example> turret on </example>
    <item> turret </item>
    <item> on </item>
</rule>

<rule id="CC_ORDER_TurretPowerDown" scope="private">
    <item> turret power down <tag> $="true"; </tag> </item>
</rule>

<rule id="CC_ORDER_TurretPowerUp" scope="private">
    <item> turret power up <tag> $="true"; </tag> </item>
</rule>

<rule id="CC_ORDER_WeaponsOn" scope="private">
    <example> weapons on </example>
    <item> weapons </item>
    <item> on </item>
</rule>
```

```
        <!-- Reports -->

    <rule id="CC_REPORT_CannonClear" scope="private">
        <one-of>
            <item> cannon clear <tag> $="true"; </tag> </item>
            <item> track and bolt not mounted <tag> $="true"; </tag> </item>
        </one-of>
    </rule>

    <rule id="CC_REPORT_CannonSafe" scope="private">
        <example> CANNON SAFE </example>
        <item> cannon safe </item>
    </rule>

    <rule id="CC_REPORT_CoaxClear" scope="private">
        <example> COAX CLEAR </example>
        <one-of>
        <item> co axe clear <tag>$="true";    </tag> </item>
        <item> coax clear <tag>$="true";</tag> </item>
        </one-of>
    </rule>

    <rule id="CC_REPORT_ComdOverrideOff" scope="private">
        <one-of>
        <item> override off <tag> $="true"; </tag> </item>

        <item> over ride off <tag> $="true"; </tag> </item>

        </one-of>

    </rule>


    <rule id="CC_REPORT_CSAMOff" scope="private">

        <one-of>

            <item> C SAM off <tag> $="true"; </tag> </item>

            <item> SEE SAM off <tag> $="true"; </tag> </item>

            <item> SEA SAM off <tag> $="true"; </tag> </item>

            <item> C SAME off <tag> $="true"; </tag> </item>

            <item> SEE SAME off <tag> $="true"; </tag> </item>

            <item> SEA SAME off <tag> $="true"; </tag> </item>

        </one-of>

    </rule>


    <rule id="CC_REPORT_DAGROff" scope="private">

        <example> dagr off or dagger off </example>

        <one-of>

        <item> dagr off <tag> $="true"; </tag> </item>

        <item> dagger off <tag> $="true"; </tag>  </item>

        </one-of>

    </rule>
```

```
<rule id="CC_REPORT_FiringNow" scope="private">
    <example> firing now </example>
    <item> firing now <tag> $="true"; </tag> </item>
</rule>

<rule id="CC_REPORT_FreeToTraverse" scope="private">
    <one-of>
        <item> my side free to traverse <tag> $="true"; </tag> </item>
        <item> free to traverse <tag> $="true"; </tag> </item>
    </one-of>
</rule>

<rule id="CC_REPORT_IIOff" scope="private">
    <one-of>
        <item> eye eye off <tag> $="true"; </tag> </item>
        <item> I I off <tag> $="true"; </tag> </item>
    </one-of>
</rule>

<rule id="CC_REPORT_LWDOff" scope="private">
    <one-of>
        <item> readout off <tag> $="true"; </tag> </item>
        <item> read out off <tag> $="true"; </tag> </item>
    </one-of>
```

```
</rule>

<rule id="CC_REPORT_LWSOff" scope="private">
    <item> display off <tag> $="true"; </tag> </item>
</rule>

<rule id="CC_REPORT_MGproven" scope="private">
    <one-of>
        <item> M G clear <tag> $="true"; </tag> </item>
        <item> M G not mounted <tag> $="true"; </tag> </item>
    </one-of>
</rule>

<rule id="CC_REPORT_ProlongedStoppage" scope="private">
    <example> PROLONGED STOPPAGE </example>
    <item> prolonged </item>
    <item> stoppage </item>
</rule>

<rule id="CC_REPORT_RadioOff" scope="private">
    <item> radio off <tag> $="true"; </tag> </item>
</rule>

<rule id="CC_REPORT_RIGHTMBGDproven" scope="private">
```

```
    <one-of>
        <item> right M B G Ds clear <tag> $="true"; </tag> </item>
        <item> right M bee gee dees clear <tag> $="true"; </tag> </item>
        <item> right M B G Ds loaded <tag> $="true"; </tag> </item>
        <item> right M bee gee dees loaded <tag> $="true"; </tag> </item>
    </one-of>
</rule>

<rule id="CC_REPORT_SEAR" scope="private">
    <example> SEAR </example>
    <item> sear </item>
</rule>

<rule id="CC_REPORT_SwitchesUp" scope="private">
    <example> SWITCHES UP  </example>
    <item> switches </item>
    <item> up </item>
</rule>

<rule id="CC_REPORT_ThermalOff" scope="private">
    <item> thermal off <tag> $="true"; </tag> </item>
</rule>
```

```
</grammar>
```

C.1.3 Number.grxml

This section provides a listing of the grammar file used to recognize numbers spoken by the CC.

```
<?xml version="1.0" encoding="UTF-8" ?>
<grammar version="1.0" xml:lang="en-US"
    tag-format="semantics-ms/1.0"
    root="DIGITS"
    mode="voice"
    xmlns="http://www.w3.org/2001/06/grammar"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2001/06/grammar
                        http://www.w3.org/TR/speech-grammar/grammar.xsd"
    xmlns:sapi="http://schemas.microsoft.com/Speech/2002/06/SRGSExtensions">

 <rule id="DIGITS" scope="public">
  <tag> $.Number="; </tag>
  <!--
  This allows digit strings of any length using number='1+'
  The script concatentates all the digits into the variable V
 -->
  <item repeat="1-">
   <ruleref uri="#DIGIT"/>
   <tag> $.Number += $$; </tag>
  </item>
</rule>

<rule id="DIGIT" scope="public">
  <one-of>
    <item>
      <ruleref uri="#ZERO"/>
      <tag> $ = $$; </tag>
    </item>

    <item>
      <ruleref uri="#HUNDRED"/>
      <tag> $ = '00'; </tag>
    </item>

      <item>
      <ruleref uri="#THOUSAND"/>
      <tag> $ = '000'; </tag>
    </item>

    <item>
      one <tag> $ ='1'; </tag>
    </item>
```

```
<item>
  two <tag> $='2'; </tag>
</item>

<item>
  three <tag> $='3'; </tag>
</item>

<item>
  four <tag> $='4'; </tag>
</item>

<item>
  five <tag> $='5'; </tag>
</item>

<item>
  six <tag> $='6'; </tag>
</item>

<item>
  seven <tag> $='7'; </tag>
</item>
```

```
  <item>
   eight <tag> $='8'; </tag>
  </item>

   <item>
   <ruleref uri="#NINE"/>
   <tag> $ = $$; </tag>
  </item>

 </one-of>
</rule>

<rule id="ZERO">
 <one-of>
  <item>
   oh <tag> $ = '0'; </tag>
  </item>

  <item>
   zero<tag> $ = '0'; </tag>
  </item>
 </one-of>
</rule>
```

```
<rule id="NINE">
    <one-of>
        <item>
           nine <tag> $='9'; </tag>
        </item>

        <item>
           niner <tag> $='9'; </tag>
        </item>
    </one-of>
</rule>

<rule id="HUNDRED" scope="private">
  <one-of>
    <item> hundred </item>
    <item> hundreds </item>
  </one-of>
</rule>

<rule id="THOUSAND" scope="private">
  <one-of>
    <item> thousand </item>
    <item> thousands </item>
  </one-of>
  </rule>
</grammar>
```

APPENDIX D COMPONENT MODEL TASK LISTS

The following task lists were derived from the IPME component project file model. The HTA reporting tool was used to extract the raw information, which was subsequently formatted and organized in Microsoft Excel. IPME assigns identification numbers to objects sequentially as they are created. The identification numbers are not renumbered when objects are deleted as object numbers are used to reference those tasks in executable statements; changing the identification numbers would require extending IPME to ensure all references to those numbers would be updated. Where feasible, the task lists have been organized to reflect increasing identification number; however, occasionally the tasks are presented out of sequence so that logical grouping of the tasks is preserved.

NOTE: In the following task lists, objects in the lists without an assigned operator are either networks, sub-networks, or network control tasks.

D.1 Prove Weapons Drill Task List

Table D-1: Prove Weapons Drill Task List

Identification Number	Name	Operator
1	Prove weapons	
3	CC starts proving weapons drill	CC
4	GNR wit for CC to finish proving MG	GNR
5	Prove left MBGDs	
5.1	Report Left MBGDs Loaded	GNR
5.2	Report Left MBGDs CLEAR	GNR
5.3	Prove left MBGDs	GNR
5.4	Left MGBDs proven	
6	Prove the pintle mount MG	
6.1	Prove the pintle mount MG	CC
6.3	Pintle MG state proven	
6.4	Verify pintle MG state	CC
6.5	Report MG not mounted	CC
8	Ensure turret power Off	
8.2	Make turret power off	GNR
8.3	Make Weapons power off	GNR
8.4	Make LASER power off	GNR

Identification Number	Name	Operator
8.5	Turret power off	
8.7	Wait for CC to report MBGD state	GNR
8.8	Ensure turret power off	
9	Prove the cannon	
9.2	Ensure BPI is in BPI_SEAR	CC
9.3	Open weapons enclosure bag	CC
9.4	Order GNR to elevate	CC
9.5	Wait for GNR to report turret power status	CC
9.6	Report Track and Bolt not mounted	CC
9.7	Cycle cannon feeder	CC
9.8	Inspect feeder sprockets	CC
9.9	Report Cannon clear	CC
9.1	Cannon Proven	
9.11	Make cannon manual safety ON	CC
9.12	Wait for GNR to elevate gun	
9.13	Prove the cannon	
11	Prove right MBGDs	
11.1	Wait for GNR to report Left MBGD status	CC
11.2	Report right MBGDs Loaded	CC
11.3	Report right MBGDs CLEAR	CC
11.4	Right MBGDs proven	
11.5	Prove right MBGDs	
12	Prepare cannon for inspection	
12.1	Remove upper & lower bridge plates	GNR
12.2	Wait for CC's order to elevate	GNR
12.3	Elevate cannon	GNR
12.4	Replace upper/lower bridge plates	GNR
12.5	Wait while CC inspects breach	GNR
12.7	Track and Bolt missing!	CC
12.8	Cannon proven	
12.9	Prepare cannon for inspection	

Identification Number	Name	Operator
14	Prove the COAX	CC
15	Weapons proven	
16	GNR starts proving weapons drill	

D.2 Opening Up Drill Task List

Table D-2: Opening Up Drill Task List

Identification Number	Name	Operator
2	Opening up procedure	
2.3	Opening up procedure	
2.7	CC interior opening up procedure	
2.7.19	Ensure turret electronics off	
2.7.19.1	Ensure turret electronics off	CC
2.7.19.2	Ensure radio power off	CC
2.7.19.3	All electronics power off	CC
2.7.19.4	Ensure II Sight power off	CC
2.7.19.5	Ensure LWD power off	CC
2.7.19.6	Ensure thermal imager power off	CC
2.7.19.7	Ensure LRF power off	CC
2.7.19.8	Ensure DAGR power off	CC
2.7.19.9	Ensure CCTD is OFF	CC
2.7.19.10	Ensure CSS is OFF	CC
2.7.19.11	Ensure CCP is OFF	CC
2.7.19.12	Ensure spotlight control unit is OFF	CC
2.7.27	Ensure weapons serviceable	
2.7.27.1	Ensure BPI in SEAR	CC
2.7.27.2	Open weapons enclosure bag on CC side	CC
2.7.27.3	Ensure ejection chutes unobstructed	CC
2.7.27.4	Ensure cannon serviceable	CC
2.7.27.5	Ensure COAX serviceable	CC

Identification Number	Name	Operator
2.7.27.6	Ensure weapons serviceable	
2.7.27.8	Ensure cannon SAFE	CC
2.7.1	CC interior opening up procedure	CC
2.7.3	Secure components and electrical cables	CC
2.7.7	Ensure kit secure	CC
2.7.8	Unlock turret	CC
2.7.9	Adjust seat	CC
2.7.10	Adjust brow pad	CC
2.7.20	Ensure turret is LOCKED	CC
2.8	GNR Ground Inspect LAV Hull LHS	
2.8.1	GNR Ground Inspect LAV Hull LHS	
2.8.2	Secure kit	GNR
2.8.3	Clear turret obstructions	GNR
2.8.4	Check for flat tires	GNR
2.8.5	Secure winch	GNR
2.8.6	Secure side hatch	GNR
2.8.7	Secure power entry panel	GNR
2.8.8	GNR Hull inspection complete	
2.9	GNR inspect top of LAV hull	
2.9.1	GNR inspect top of LAV hull	GNR
2.9.2	Raise turret wire cutter	GNR
2.9.3	Secure MBGDs	GNR
2.9.4	Clear sight head LOS	GNR
2.9.5	Secure GNR hatch open	GNR
2.9.6	GNR Hull top secure	
2.12	DRV Inspect top of LAV hull	
2.12.1	DRV Inspect top of LAV hull	
2.12.2	Secure drivers wire cutter	DRV
2.12.3	Secure barbed wire	DRV
2.12.4	Remove DVA cover	DRV
2.12.5	Clear driver periscopes	DRV

Identification Number	Name	Operator
2.12.9	DRV Hull top check complete	
2.16	CC Inspect Top of LAV Hull	
2.16.1	CC Inspect top of LAV Hull	
2.16.2	Ensure 25mm and Coax muzzle covers are removed	CC
2.16.3	Ensure barrels locked and secured	CC
2.16.4	Clear turret obstructions	CC
2.16.5	Clear casing ejection port	CC
2.16.14	LAV Hull Secure	
2.16.15	Clear link ejection opening	CC
2.16.17	Turret weapons unobstructed	
2.16.18	Inspect MBGDs for serviceability	CC
2.16.19	Inspect Link ejection port for fouling	CC
2.16.20	Ensure CC sight head unobstructed	CC
2.16.21	Open CC sight head cover	CC
2.16.22	Secure Platt mount	CC
2.16.23	Mount MG	CC
2.16.24	Secure turret wind screens	CC
2.16.25	Ensure antennas are tied down/secure	CC
2.16.26	Ensure external load distributed in RHS bin	CC
2.16.27	Remove LWR cover and ensure unobstructed	CC
2.16.28	Check for damage to sight windows	CC
2.16.29	Secure GPS antenna	CC
2.16.30	Ensure external load distributed in rear bustle	CC
2.16.31	Check security of Sentry hatches	CC
2.16.32	Check security of ECPK	
2.16.33	Check security of centre hatch	CC
2.16.34	Ensure CC hatch is secured OPEN	CC
2.17	CC Ground inspect LAV Hull RHS	
2.17.1	CC Ground Inspect LAV Hull RHS	
2.17.2	Secure kit	CC
2.17.3	Clear turret obstructions	CC

Identification Number	Name	Operator
2.17.4	Check for flat tires	CC
2.17.5	Ensure roof hatches closed and locked	CC
2.17.6	Close signal entry panel	CC
2.17.7	Ensure ramp up	CC
2.17.8	LAV RHS Secure	
2.17.9	Secure tire chain bin	CC
2.17.10	Lock ramp door	CC
2.17.11	Ensure ramp door closed	CC
2.3	GNR interior opening up procedure	
2.30.1	GNR interior opening up procedure	
2.30.2	Check components and electrical cables	GNR
2.30.3	Ensure Thermal Sight is OFF	GNR
2.30.4	Ensure LASER is OFF	GNR
2.30.5	Check CDA cover operation	GNR
2.30.6	Ensure STAB is OFF	GNR
2.30.7	Ensure Battle Override is OFF	GNR
2.30.8	Ensure MBGDs set to SAFE	GNR
2.30.9	Ensure Gun Arm switch set to SAFE	GNR
2.30.10	Open the weapons enclosure bag	GNR
2.30.11	Ensure feed chutes secure and serviceable	GNR
2.30.12	Ensure Bridge Plates secure	GNR
2.30.13	Ensure primary ammo bin clean and serviceable	GNR
2.30.14	Close weapons enclosure bag	GNR
2.30.15	Adjust GNR seat	GNR
2.30.16	Adjust GNR sight brow pads	GNR
2.30.17	Ensure kit stowed	GNR
2.30.18	Ensure Turret OFF	GNR
2.30.19	Ensure Weapons OFF	GNR
2.30.20	Ensure AUX OFF	GNR
2.30.21	Make Power/Mode switch OFF	GNR
2.30.22	Set WFOV	GNR

Identification Number	Name	Operator
2.30.23	Set GDU brightness	GNR
2.32	DRV Ground Inspect LAV Hull	
2.32.1	DRV Ground Inspect LAV Hull	
2.32.2	Ensure tow cable is secure	DRV
2.32.3	Close/secure front hatch	DRV
2.32.4	Ensure barbed wire strapped down	DRV
2.32.5	Ensure headlights and brackets secure	DRV
2.33	DRV Interior Opening Up procedure	
2.33.1	DRV Interior Opening Up procedure	
2.33.2	Ensure all PDA circuit breakers are OFF	DRV
2.33.3	Verify secondary ammo bin clean and serviceable	DRV
2.33.4	Secure and lock DRV escape hatch	DRV
2.33.5	Make Auto and Aux switches ON	DRV
2.33.6	Close and lock DRV hatch	DRV
2.33.7	Adjust seat and pedals	DRV
2.33.8	Set hatch at desired opening	DRV
2.33.9	Raise hatch safety arm	DRV
2.33.10	Adjust hatch	
2.33.11	Perform BIT on enunciator panel	DRV
2.33.12	Release the Air Brake	DRV
2.33.13	Ensure the external AFESS button is OFF	DRV
2.33.14	Set up DVA	DRV
2.33.15	Ensure TacNav circuit breaker is IN (RUN)	DRV
2.33.16	Observe TacNav BIT	DRV
2.33.17	Verify TacNav display operation	DRV
2.14	CC opening up procedure	
2.15	CC Climb on vehicle	CC
2.18	CC enter turret	CC
2.19	GNR opening up procedure	
2.20	GNR climb on vehicle	GNR
2.21	GNR enter turret	GNR

Identification Number	Name	Operator
2.23	DRV opening up procedure	DRV
2.24	DRV climbs on vehicle	DRV
2.25	DRV enter driver compartment	DRV
2.26	Opening Up procedure complete	
2.27	DRV opening up procedure complete	DRV
2.28	GNR opening up procedure complete	
2.29	CC opening up procedure complete	

D.3 Turret Power Up Procedure Task List

Table D-3: Turret Power Up Procedure Task List

Identification Number	Name	Operator
3	Turret Power Up procedure	
3.1	CC starts turret power up procedure	
3.2	Wait for CC to order Power Up	GNR
3.3	Turret Power Up procedure	
3.4	Order Turret Power Up Drill	CC
3.5	Switch Turret ON	GNR
3.6	Complete CDA lamp test	GNR
3.7	Turn LASER on	GNR
3.8	Turn thermal imager on	GNR
3.9	Turn STAB ON	GNR
3.10	Set drift null switch to auto	GNR
3.11	Turn DAGR on	CC
3.12	Turn thermal image display on	CC
3.13	Ensure Comd Override is off	CC
3.14	Ensure laser readout is on	CC
3.15	Turn LWD on	CC
3.16	Turn II sight on	CC
3.17	Turn CSAM on	CC

Identification Number	Name	Operator
3.18	Turn radios on	CC
3.20	Turn Auxiliary Turret Power ON	GNR
3.21	Turn Weapons ON	GNR
3.22	GNR Power Up procedure complete	
3.23	CC Power Up procedure complete	
3.24	Supervise GNR	CC
3.25	GNR starts turret power up procedure	

D.4 Action Drill Task List

Table D-4: Action Drill Task List

Identification Number	Name	Operator
8	Action Drills	
8.2	CC Exterior Action Drills	
8.2.1	CC Exterior Action Drills	
8.2.2	Remove cannon and muzzle covers	CC
8.2.3	Ensure rear deck hatches closed and secured	CC
8.2.4	Remove turret/sight obstructions	CC
8.2.5	Load MBGDs	CC
8.2.6	Load pintle mount MG	CC
8.2.7	Drop into turret interior	CC
8.2.8	Remove barrel obstructions	CC
8.2.10	DRV and Rear Hatches secured	
8.2.11	Ensure driver hatch closed and secured	CC
8.2.12	Order Hatches Closed	CC
8.2.13	Report MBGDs not used	CC
8.2.14	GNR Load MBGDs	GNR
8.3	CC and GNR interior action drills	
8.3.53	CC prepare for Crew Action	
8.3.53.1	CC prepare for Crew Action	

Identification Number	Name	Operator
8.3.53.2	Unlock turret	CC
8.3.53.3	Make cannon manual safety ON	CC
8.3.53.4	Ensure BPI in Sear	CC
8.3.53.6	Order Crew Action	CC
8.3.53.7	Ensure turret basket clear	CC
8.3.55	GNR prepare for Crew Action	
8.3.55.1	Turn turret power on	GNR
8.3.55.2	Turn weapons power on	GNR
8.3.55.3	Turn auxiliary power on	GNR
8.3.55.4	Ensure turret basket clear	GNR
8.3.55.5	GNR wait for crew action order	GNR
8.3.55.6	Stow personal kit	GNR
8.3.55.8	GNR prepare for Crew Action	
8.3.56	GNR Prepare Weapon System	
8.3.56.1	GNR Prepare Weapon System	
8.3.56.2	Turn laser ON	GNR
8.3.56.3	Select first on first/last switch	GNR
8.3.56.4	Set Thermal Sight to WIDE	GNR
8.3.56.5	Turn thermal sight ON	GNR
8.3.56.7	Turn STAB ON	GNR
8.3.56.8	Enable automatic Drift correction	GNR
8.3.56.9	Make cannon safe	GNR
8.3.56.10	Set power controller to MAIN	GNR
8.3.56.11	Select ?single sho”	GNR
8.3.56.12	Select ammo type ”primary”	GNR
8.3.56.13	Press palm switch on hand controller	GNR
8.3.56.14	GNR Observe through sight	GNR
8.3.56.15	Grasp Hand Controller	GNR
8.3.57	CC Prepare Weapon System	
8.3.57.1	Turn DAGR ON	CC
8.3.57.2	Turn CC thermal sight ON	CC

Identification Number	Name	Operator
8.3.57.3	Select thermal sight FOV to WIDE	CC
8.3.57.4	Make commander's override switch OFF	CC
8.3.57.5	Make laser readout ON	CC
8.3.57.6	Make TacNav ON	CC
8.3.57.7	Make LWS display on	CC
8.3.57.8	Make Battle Override OFF	CC
8.3.57.9	Make MBGDs SAFE	CC
8.3.57.10	Hand controller switches UP	CC
8.3.57.11	Order 'Ready Cannon'	CC
8.3.57.12	Wait for GNR to finish preparing weapon system	CC
8.3.57.13	CC Prepare Weapon System	CC
8.3.1	Complete interior action drills	
8.3.2	CC interior action drill	
8.3.3	GNR interior action drill	
8.3.39	Fire cannon	GNR
8.3.40	Ensure BPI in MISFIRE	CC
8.3.41	Reset misfire button	CC
8.3.43	Verify BPI in BPI_SEAR	CC
8.3.50	Make Cannon ready to fire	CC
8.3.54	Supervise GNR	CC
8.3.59	Report Misfire	GNR
8.3.62	Wait for order to fire cannon	GNR
8.3.63	Wait for GNR to report misfire	CC
8.3.64	Perform Start Mode drill	
8.3.65	Wait for GNR to report Firing Now	CC
8.3.66	GNR wait for CC's SEAR report	GNR
8.3.67	CC interior action drill complete	
8.3.69	GNR interior action drill complete	
8.4	DRV Interior Action Drills	
8.4.2	DRV start up procedure	
8.4.2.1	DRV start up procedure	

Identification Number	Name	Operator
8.4.2.2	Make Auxiliary switch ON	DRV
8.4.2.3	Ensure enunciator panel BIT clear	DRV
8.4.2.4	Test air breaks	DRV
8.4.2.5	Test AFESS	DRV
8.4.2.6	Make Auto switch ON	DRV
8.4.1	DRV interior action drill	
8.4.3	Close and Lock driver's hatch	DRV
8.4.4	Make DCB ON	DRV
8.4.5	Turn on NAV system	DRV
8.4.6	Make LWD ON	DRV
8.4.7	Turn DVA ON	DRV
8.4.8	Make Dome light override ON	DRV
8.4.9	DRV interior action drill complete	
8.5	Action Drills	
8.6	Assign arcs of responsibility INCOMPLETE	CC
8.7	Action Drill complete	

D.5 Fire Order and Engagement Task List

Table D-5: Fire Order and Engagement Task List

Identification Number	Name	Operator
1	Fire order and engagement	
6	GNR Scan for target in WFOV	GNR
15	Give executive order to FIRE	CC
20	Respond to fire order	GNR
21	End engagement	CC
22	GNR Return to START MODE	GNR
23	CC Return to START MODE	CC
28	Issue fire order	
28.1	Issue fire order	

Identification Number	Name	Operator
28.2	Specify Main gun ammo type	CC
28.3	Specify COAX	CC
28.4	Specify target	CC
28.5	Estimate target range	CC
28.6	CC Initially align gun to target	CC
28.7	Direct gunner onto target	CC
28.9	Fire order complete	
28.10	Order Lase	CC
28.11	CC Lase target	CC
28.12	Recognize a hostile target	CC
29	Lase Target	
29.1	Fire Laser Range Finder	GNR
29.2	Report Range	GNR
29.4	Receive null return	GNR
29.6	Report WRONG LAY	GNR
29.7	Report NO SOLUTION	GNR
29.8	Lase target	GNR
29.9	Refine aim point for relase	GNR
29.1	Double return	GNR
29.11	Target lasing complete	
30	Verify firing solution	
30.6	Confirm range	
30.6.1	Order Re-Lase / Re-Lay	CC
30.6.2	Confirm range	CC
30.6.4	Order FIRST	CC
30.6.6	Range confirmed	
30.6.8	Order LAST	CC
30.2	Confirm target	CC
30.3	Monitor aim point	CC
30.4	Verify targeting solution	
30.5	Good firing solution	

Identification Number	Name	Operator
30.8	Stop monitoring aim point	
31	GNR Select ammo type	GNR
36	Make GUN ARMED	
37	CC Scan for target in WFOV or binoculars	CC
38	GNR Select Weapon	GNR
40	Wait for Fire Order	GNR
44	Observe target	CC
48	Initialize GNR engagement variables	
49	Initialize CC engagement variables	
50	Arming weapon finished	
58	Begin	
46	Engage the target	
46.1	Fire the action	GNR
46.2	Observe fall of shot	GNR
46.3	Ensure line of fire is unobstructed	GNR
46.4	Engage the target	
46.5	Maintain aiming point	GNR
46.6	Stop engaging target	
54	GNR initiated engagement -- GNR initial tasks	
54.1	GNR initiated engagement -- GNR initial tasks	
54.2	GNR Estimated range	GNR
54.3	Gunner identify target type	GNR
54.4	Enter target range	GNR
54.5	GNR View target in NFOV (GNR initiated)	GNR
54.7	Fire Laser Range Finder	GNR
54.8	Report Range	GNR
55	CC initiated engagement -- GNR initial tasks	
55.1	CC initiated engagement -- GNR initial tasks	
55.2	GNR View target in NFOV (CC Initiated)	GNR
55.4	Report ON	GNR
55.5	Enter commanded range	GNR

D.6 Misfire and Stoppage Immediate Action Drills Task List

Table D-6: Misfire and Stoppage Immediate Action Drills Task List

Identification Number	Name	Operator
28	Misfire Drills	
35	Weapon selection error	
30	25mm Misfire Drill (Immediate Action)	
30.1	25mm Misfire Drill (Immediate action)	
30.4	CC listens for misfire report	CC
30.5	Retain sight picture	GNR
30.6	GNR Reports Misfire	GNR
30.7	Report 'MISFIRE WAIT'	CC
30.8	Check BPI position	CC
30.16	Go to Secondary Actions	
30.22	GNR Selects SINGLE SHOT	GNR
30.23	CC Order Turret Make Safe	CC
30.24	CC Retime Cannon	
30.29	GNR Respond to Turret Make Safe order	
30.31	Revert to normal Gun Drill	
30.32	GNR begin MISFIRE Drill again	
30.33	GNR starts misfire drill	
30.37	GNR selects firing rate 200	GNR
30.43	Cannon fails to fire	
30.44	UnNamed	
30.59	GNR fires cannon	GNR
30.60	Wait for CC direction	GNR
30.68	Misfire: BPI in SEAR	
30.68.1	Misfire: BPI in SEAR	
30.68.2	CC Make Gun-Arm ON	CC
30.68.3	CC Check SEAR indicator	CC
30.68.4	Check ammo indicator	CC
30.68.5	Check WPN indicator	CC

Identification Number	Name	Operator
30.68.6	Check if low battery indicator is lit	CC
30.68.7	Check if Temp/Fault indicator	CC
30.68.9	CC Check CDA Gun Arm indicator	CC
30.68.10	CC Low Ammo response	CC
30.68.11	Response to LOW BATT indicator	CC
30.68.12	CC Order DRV to Check Hatch	CC
30.68.13	Select Battle Override	CC
30.68.14	CC Report BPI in SEAR	CC
30.68.15	Report gun ARMED	CC
30.68.16	Override Temperature Fault	CC
30.68.17	TEMPORARY	
30.68.18	Wait for DRV to secure hatch	CC
30.68.19	WPN Indicator Light secure	
30.68.20	Wait for DRV to start engine	CC
30.68.21	Immediate Action fault finding complete	CC
30.68.22	TEMPORARY	
30.68.23	TEMPORARY	
30.68.25	Misfire Fault Corrected	CC
30.69	Misfire: BPI in MISFIRE	
30.69.1	Misfire: BPI in MISFIRE	
30.69.2	Check Manual safety	CC
30.69.3	Make CDA Misfire reset button	CC
30.69.4	Observe movement of rounds in feed chutes	CC
30.69.5	Make Manual Safety to FIRE	CC
30.69.6	CC Order SingleShot	CC
30.69.7	CC reports RESET	CC
30.69.8	Command 200 READY	CC
30.69.11	Wait for GNR to respond to Single Shot order	CC
30.69.13	Check CDA Misfire reset indicator	CC
30.69.15	Check BPI position	CC
30.69.16	Report BPI in SEAR	CC

Identification Number	Name	Operator
30.69.17	Report 'MISFIRE WAIT'	CC
30.74	Revert to normal Gun Drill	
30.98	CC starts misfire drill	
34	COAX Stoppage drills (Immediate Action)	
34.1	COAX Stoppage drills (Immediate Action)	
34.12	COAX Stoppage	
34.13	CC starts COAX stoppage drill	CC
34.14	GNR starts COAX stoppage drill	GNR
34.15	Supervise GNR during IA drill	CC
34.16	Release hand controller trigger	GNR
34.17	Check Ammo select	GNR
34.19	Lay back on target	GNR
34.24	Revert to normal Gun Drill	
34.25	Revert to normal Gun Drill	CC
34.26	Report Ammo Select Correct	GNR
34.33	CC Check SEAR indicator -- IS THIS REQUIRED FOR THE COAX???	CC
34.34	Check ammo indicator	CC
34.35	Check WPN indicator light	CC
34.36	Check LOW BATT indicator	CC
34.37	Check Temp/Fault indicator	CC
34.39	Low Ammo response	CC
34.40	Response to LOW BATT indicator	CC
34.41	Order DRV to Check Hatch	CC
34.42	Select Battle Override	CC
34.44	Override Temperature Fault	CC
34.46	Wait for DRV to secure hatch	CC
34.47	WPN Indicator Light secure	
34.48	Wait for DRV to start engine	
34.49	Immediate Action fault finding complete	CC
34.51	TEMPORARY	

Identification Number	Name	Operator
34.53	Report READY	CC
34.55	CC Make Gun-Arm ON	CC
34.56	Check CDA Gun Arm indicator	CC
34.57	Report gun ARMED	CC
34.59	TEMPORARY	
34.63	Check COAX manual safety	CC
34.64	Make Manual Safety to FIRE	CC
34.65	Verify COAX firing	CC
34.66	COAX still fails to fire	
34.67	Verify position of COAX working parts	CC
34.68	COAX working parts fully forward	
34.69	COAX working parts fully to rear	
34.70	COAX working parts partially forward	
34.71	Cock COAX	CC
34.72	Order READY	CC
34.73	Verify COAX firing	GNR
34.74	COAX still fails to fire	
34.75	Inspect COAX ammo feed and bin	GNR
34.76	Report 'Firing Circuit'	CC
34.77	CC manually fire action	CC
34.78	Report 'Prolonged stoppage'	CC
34.79	COAX fires	
34.80	GNR begin COAX secondary actions	
34.82	CC begin COAX secondary actions	

D.7 Start Mode Drill Task List

Table D-7: Start Mode Drill Task List

Identification Number	Name	Operator
37	Start mode drill	

Identification Number	Name	Operator
38	CC Start Mode	CC
39	GNR Start Mode	GNR
42	Set Nav to Turret	CC
43	Report 'Start Mode'	CC
44	Make gun safe	GNR
45	Select LRF to FIRST	GNR
46	Select WIDE FOV	GNR
47	Select Cannon	GNR
48	Select 200 firing rate	GNR
49	Report 'Start Mode'	GNR
50	GNR Start Mode drill complete	
51	Select GCP AUTO tracking	GNR
52	Set CCTD video switch	CC
53	Turn CCP Commanders Override to standby	CC
54	Wait for GNR's report	CC
65	CC Start Mode drill complete	
64	CC Switches up	
64.1	CC Switches up	
64.3	Select Cannon	CC
64.4	Select 200 rnds/min firing rate	CC
64.5	Report switches up	CC
64.6	CC Switches up completed	
64.9	Select Primary ammo bin	CC

D.8 Turret Make Safe Drill Task List

Table D-8: Turret Make Safe Drill Task List

Identification Number	Name	Operator
1	Turret Make Safe	
2	CC Turret Make Safe	

Identification Number	Name	Operator
3	GNR Turret Make Safe	
4	Order Turret Make Safe	CC
6	Make gun safe	GNR
7	Make PDA Weapons power off	GNR
8	Make LASER power off	GNR
9	Wait for GNR to report	CC
10	Make cannon manual safety ON	CC
11	Make COAX manual safety ON	CC
12	Make MBGDs SAFE	GNR
13	Make pintle MG manual safety ON	CC
14	CC Turret make safe completed	
15	GNR Turret make safe completed	

D.9 Close Down Procedure Task List

Table D-9: Close Down Procedure Task List

Identification Number	Name	Operator
6	Closing Down procedure	
6.1	Closing down procedure	
6.2	CC starts closing down procedure	
6.3	GNR starts closing down procedure	
6.4	Driver starts closing down procedure	
6.5	Replace cannon muzzle cover	CC
6.6	Replace CSAM cover	CC
6.7	Release tension on antenna	GNR
6.8	Switch off engine	DRV
6.9	Turn DCB OFF	DRV
6.1	Turn off DVA	DRV
6.11	Turn off Auto and Aux switch	DRV
6.12	Empty air reservoir	DRV

Identification Number	Name	Operator
6.13	Replace DVA cover	DRV
6.14	Exit vehicle	DRV
6.15	Complete last parade	DRV
6.16	Close and lock driver’s hatch	DRV
6.17	Lock Turret	CC
6.18	Close/Lock CC hatch	CC
6.20	Replace COAX muzzle cover	CC
6.21	CC closedown complete	
6.22	Close/Lock GNR hatch	GNR
6.23	GNR closing down procedure complete	
6.24	Turn LWD OFF	GNR
6.25	DRV Close down procedure complete	

D.10 Turret Power Down Procedure Task List

Table D-10: Turret Power Down Procedure Task List

Identification Number	Name	Operator
4	Turret Power Down Drill	
4.1	GNR starts turret power down drill	GNR
4.2	CC starts turret power down drill	CC
4.3	Order turret power down	CC
4.4	Turn off radios	CC
4.5	Turn off CSAM	CC
4.6	Turn off II sight	CC
4.7	Turn off LWS	CC
4.8	Ensure laser readout is off	CC
4.9	Turn off thermal imager sight	CC
4.10	Turn off DAGR	CC
4.11	Turret power down drill	
4.13	Turn off thermal imager	GNR

Identification Number	Name	Operator
4.14	Turn LASER OFF	GNR
4.15	Shut Turret PDA circuit breakers OFF	GNR
4.16	Lock Turret	CC
4.17	CC Leave turret	CC
4.18	Lock CC's hatch	GNR
4.19	GNR Leave turret	GNR
4.20	Close GNR hatch	GNR
4.23	Shut weapon PDA circuit breakers off	GNR
4.24	Shut auxiliary power PDA circuit breakers off	GNR
4.25	Close CC hatch	CC
4.26	Lock GNR Hatch	GNR
4.28	Wait for CC to finish first Power Down tasks	GNR
4.30	Wait for CC to close hatch	GNR
4.31	CC Power Down procedure done	
4.32	GNR Power Down procedure done	
4.34	Wait for GNR to finish first Power Down tasks	GNR
4.36	Turn off Commander's Override	CC
4.37	Turn CDA OFF	GNR

APPENDIX E LAV CREW DIALOGUE

These lists of expressions and keywords are based on doctrine derived from CAF documents (DND, 2004, 2016b, 2016a, 2016b). The expressions are in ALL CAPS to indicate that they are generally spoken loudly to be heard over the noise of the vehicle and without going through the intercom (i.e., open air communications that have to penetrate the headset hearing protection).

NOTE: The tables do not repeat communications that occur more than once in a given drill or procedure.

The focus of these communications is on exchanges between the turret crew; there are few formal communications with the DRV and most require no verbal feedback from the DRV. Most of the formal communications with the DRV concern control of the vehicle under specific conditions that require unambiguous direction. Feedback is evident in the motion of the vehicle. Typically, driver orders are provided as natural language directions.

Acronyms are expressed as a “word” but some manipulation may be required to ensure that the speech production software produces the intended “word”. Abbreviations are expressed as single letters in rapid succession to ensure the speech production software clearly enunciates the intended expression.

E.1 Prove Weapons Drill Verbal Communications

Table E-1: Prove Weapons Drill Verbal Communications

CC	GNR
MG CLEAR	LEFT MBGD CLEAR
MG NOT MOUNTED	TURRET OFF
RIGHT MBGD CLEAR	WEAPONS OFF
CANNON SAFE	AUXILIARY OFF
TRACK AND BOLT NOT MOUNTED	LASER OFF
SEAR	ELEVATING
ELEVATE	DEPRESSING
CANNON CLEAR	
COAX CLEAR	
COAX NOT MOUNTED	
MBGDS SAFE	
ALL WEAPONS CLEAR	

E.2 Opening Up Procedure

There is no formal communication amongst the crew during this procedure.

E.3 Power Up Procedure

The LAV 6 aide memoire (DND, 2015) does not indicate explicit communication amongst the crew as is indicated in the LAV III aide memoire (DND, 2016a). However, the LAV 6 aide memoire suggests that similar verbal communication does occur as it states:

> "*Power Up Procedures are done simultaneously by the crew*." (DND, 2015, p. 9)

Table E-2 summarizes the LAV III Power Up procedure verbal communications, but it includes implied communications from the LAV 6 Power Up procedure.

Table E-2: Power Up Procedure Verbal Communications

CC	GNR
TURRET ON	TURRET ON
WEAPONS ON	WEAPONS ON
AUXCILIARY ON	AUXCILIARY ON
DAGR ON	LASER ON
THERMAL ON	THERMAL ON
OVERRIDE OFF	STAB ON
READOUT ON	DRIFT ELIMINATED
DISPLAY ON	STANDBY
II ON	POWER ON
CSAM ON	WEAPONS OFF
RADIO ON	
MBGDs SAFE	

E.4 Action Drill Verbal Communications

Table E-3: Action Drill Verbal Communications

CC	GNR
MUZZLE ENDS CLEAR	TURRET ON
HATCHES CLOSED	WEAPONS ON
NO OBSTRUCTIONS	AUXILIARY ON

CC	GNR
M B G Ds LOADED	MY SIDE FREE TO TRAVERSE
M B G Ds NOT IN USE	LASER ON
M G LOADED	FIRST
M G NOT IN USE	WIDE
MY SIDE FREE TO TRAVERSE	THERMAL ON
TURRET UNLOCKED	STAB ON
CANNON SAFE	DRIFT ELIMINATED
SEAR	GUN SAFE
CREW ACTION	MAIN
DAGR ON	SINGLE SHOT
THERMAL ON	PRIMARY
WIDE	HANDS ON MY CONTROLS
OVERRIDE OFF	OBSERVING TO MY FRONT
READOUT ON	FIRING NOW
WIDE	MISFIRE
OVERRIDE OFF	200
READOUT ON	START MODE
NAV ON	
DISPLAY ON	
BATTLE OVERRIDE OFF	
MBGDs SAFE	
SWITCHES UP	
READY CANNON	
MISFIRE WAIT	
RESET	
CANNON READY	

E.5 Start Mode Drill Verbal Communications

Table E-4: Start Mode Drill Verbal Communications

CC	GNR
SWITCHES UP	GUN SAFE

CC	GNR
I R	FIRST
STANDBY	AUTO
NAV TO TURRET	WIDE
START MODE	MAIN
	200
	START MODE

E.6 Turret Make Safe Drill

Table E-5: Turret Make Safe Drill Communications

CC	GNR
TURRET MAKE SAFE	GUN SAFE
CANNON SAFE	WEAPONS OFF
COAX SAFE	LASER OFF
M B G Ds SAFE	
MG SAFE	

E.7 Power Down Procedure

As with the Power Up procedure, the LAV 6 aide memoire (DND, 2015) does not indicate explicit communication amongst the crew for the Power Down procedure as is indicated in the LAV III aide memoire (DND, 2016a). Table E-6 summarizes the LAV III Power Up procedure verbal communications, but it includes implied communications from the LAV 6 Power Up procedure.

Table E-6: Power Down Procedure Verbal Communications

CC	GNR
RADIO OFF	CDA OFF
CSAM OFF	THERMAL OFF
II OFF	LASER OFF
DISPLAY OFF	TURRET OFF
READOUT OFF	WEAPONS OFF
OVERRIDE OFF	AUXILIARY OFF
THERMAL OFF	STAB OFF

CC	GNR
DAGR OFF	IRSTAS OFF
TURRET LOCK	

E.8 25mm Misfire Drill Verbal Communications

Table E-7: Misfire Drill Verbal Communications

CC	GNR
MISFIRE WAIT	MISFIRE
SEAR	FIRING NOW
GUN ARMED	SINGLE SHOT
TURRET MAKE SAFE	200
LOW AMMO	
DRIVER CHECK YOUR HATCH	
BATTLE OVERRIDE	
LOW BATTERY	
READY	
SINGLE SHOT	
RESET	
200	
JAMMED CANNON	
CLEAR WEAPONS	

E.9 Coaxial MG Stoppage Drill Verbal Communications

Table E-8: Stoppage Drill Verbal Communications

CC	GNR
ARMED	STOPPAGE
LOW AMMO	AMMO SELECT WRONG
DRVER CHECK YOUR HATCH	AMMO SELECT CORRECT
BATTLE OVERRIDE	FIRING NOW
LOW BATTERY	
COAX ARMED	

CC	GNR
READY	
FIRING CIRCUIT	
FIRING NOW	
PROLONGED STOPPAGE	
TURRET MAKE SAFE	
AMMO SELECT WRONG	

E.10 Firing Orders

Firing orders and reporting requires range reporting in a somewhat standardized format. In some cases, distances are reported as natural language numbers (e.g., 1,500 is reported as fifteen hundred) but they can also be reported in a more stylized format (e.g., one five hundred). For distances that are round numbers of 1,000, natural language is typically used (e.g., 2,000 is reported as two thousand). There are other natural language communications required to direct attention to a target location, to indicate the movement of a target, or to select a specific target when multiple targets exist.

Several expressions are common to both turret crewmembers and these expressions are presented at the bottom of Table E-9. In several cases, expressions may be combined (e.g. 200 READY; AIM LEFT; APC MOVING RIGHT).

Table E-9: Engagement Fire Order Verbal Communications

CC	GNR
STOP	CONTACT
FIRE	DOUBLE
TARGET STOP	NOT OBSERVED
MY SIDE	WRONG LAY
NEXT TARGET	NO SOLUTION
COAX	SECOND ROUND
GO ON	GUN SAFE
REPEAT	WIDE
LASE	MAIN
RE-LASE	200
RE-LAY	
SABOT	

CC	GNR
H E	
FRANG[10]	
TRAVERSING FIRE	
MULTIPLE TARGETS	
Common expressions	
FIRING NOW[11]	
BATTLE	
TARGET	
FIRST	
LAST	
ON	
WAIT	
A P C[12]	
TRANSPORT	
MULTIPLE TARGETS	
MEN IN OPEN	
TRENCHES	
MEN	
LEFT/RIGHT	
TOP/BOTTOM	
DROP/ADD	
LEFT/RIGHT	
LEFT/RIGHT HAND EDGE	
AIM	
MOVING	

[10] Pronounced fran-je, where the g is pronounced je as in ‘je suis’.
[11] Under some conditions, the CC will opt to take control of the engagement and fire the weapon.
[12] BMP is occasionally used as a target type; BMP has been officially replaced by the more general APC.

E.11 Speech Recognition Verification Results

The following list describes the CC variables related to the model execution and driven by speech recognition. The list is organized first by the failure or successful implementation of the speech recognition, then alphabetically.

Table E-10: Speech Recognition Verification Results

IPME Variable	Expression to be Recognized	Verification Outcome
CC_FO_Range	This is supposed to return a string that represents the distance the CC reports as the range; however, it has a stylized format. It can be either "BATTLE" or a number following one of the following templates: • Number HUNDRED (e.g. "100" for ONE HUNDRED) • Number THOUSAND (e.g. "1000" for ONE THOUSAND) • Number number HUNDRED (e.g. "1 200" for ONE TWO HUNDRED representing the range 1200)	Fail
CC_ORDER_Relase	Relase	Fail
BEGIN	Begin	Success
CC_FO_AmmoType	1 of FRANG, SABOT, HE, COAX	Success
CC_FO_TargetType	1 of APC, BMP, MEN, TANK, TRANSPORT, UNKONWN, MEN IN OPEN, MEN IN TRENCHES	Success
CC_FreeToTraverse	My side free to traverse	Success
CC_ORDER_200Ready	200 ready	Success
CC_ORDER_AuxiliaryOn	Auxiliary on	Success
CC_ORDER_CannonDepress	Depress	Success
CC_ORDER_CannonElevate	Elevate	Success
CC_ORDER_CrewAction	Crew action	Success
CC_ORDER_DriverStartVehicle	Driver start your vehicle	Success
CC_ORDER_Fire	Fire	Success
CC_ORDER_FirstLast	1 of FIRST or LAST	Success
CC_ORDER_GoOn	Go on	Success
CC_ORDER_Lase	Lase	Success

IPME Variable	Expression to be Recognized	Verification Outcome
CC_ORDER_LaserOn	Laser On	Success
CC_ORDER_MisfireWait	Misfire Wait	Success
CC_ORDER_NextTarget	"Next target" or "Stop next target"	Success
CC_ORDER_Ready	Ready	Success
CC_ORDER_Reset	Reset	Success
CC_ORDER_SingleShot	"Single" or "Single shot"	Success
CC_ORDER_Stop	Stop	Success
CC_ORDER_TargetStop	Target stop	Success
CC_ORDER_TurretMakeSafe	Turret make safe	Success
CC_ORDER_TurretOn	Turret on	Success
CC_ORDER_TurretPowerDown	Turret power down	Success
CC_ORDER_TurretPowerUp	Turret power up	Success
CC_ORDER_WeaponsOn	Weapons on	Success
CC_REPORT_CannonClear	Cannon Clear	Success
CC_REPORT_CannonSafe	Cannon Safe	Success
CC_REPORT_CoaxClear	Coax clear	Success
CC_REPORT_ComdOverrideOff	Override off	Success
CC_REPORT_CSAMOff	CSAM off	Success
CC_REPORT_DagrOff	Dagr off	Success
CC_REPORT_FiringNow	Firing Now	Success
CC_REPORT_FreeToTraverse	My side free to traverse	Success
CC_REPORT_IIoff	I I off	Success
CC_REPORT_LWDOff	Readout off	Success
CC_REPORT_LWSOff	Display off	Success
CC_REPORT_MGproven	1 of "M G clear" or "M G not mounted"	Success
CC_REPORT_ProlongedStoppage	Prolonged stoppage	Success
CC_REPORT_RadioOff	Radio off	Success
CC_REPORT_RightMBGDproven	1 of "Right M B G Ds loaded" or "Right M B G Ds clear"	Success
CC_REPORT_SEAR	Sear	Success
CC_REPORT_SwitchesUp	switches up	Success

IPME Variable	Expression to be Recognized	Verificatio n Outcome
CC_REPORT_ThermalOff	Thermal off	Success

APPENDIX F IPME PRX FILE XML SCHEMA

Development of the HTA data extraction tool required identifying the associated XML fields and variables within the IPME PRX file. The resulting description is provided here for reference and future development.

```
<project>
    <network>
        <compositeID/>
        <name/>
        <notes/>
        <lastNum/>
        <location/>
        <width/>
        <height/>
        <disProps/>
    </network>
    <task>
        <vacpData>
            <vacp>
                <weight/>
                <selection/>
            </vacp>
            <windex>
                <weight/>
            </windex>
        </vacpData>
        <taskNotes>
            <note>
                <text/>
            </note>
        </taskNotes>
        <hgaData>
            <inputSensations>
                <sensation>
                    <checked/>
                </sensation>
            </inputSensations>
            <outputBehaviors>
                <behavior>
                    <checked/>
            </behavior>
        </outputBehaviors>
        <cognitiveProcesses>
        </cognitiveProcesses>
        <externalVariables>
        </externalVariables>
        <internalVariables>
        </internalVariables>
    </hgaData>
    <consequencesOfFailure>
        <probabilityNoEffect/>
        <probabilityDegrading/>
        <probabilityTaskFollow/>
        <probabilityTaskRepeat/>
        <probabilityModelTerminate/>
        <degradedTaskID/>
        <percentageTimeDegradation/>
        <percentageFailureDegradation/>
        <taskToFollowID/>
    </consequencesOfFailure>
    <SchedulerData>
        <ipData>
            <ipComponents>
                <visionEnabled/>
                <visionExternallyCued/>
                <visionCategory/>
                <visionLocation/>
                <auditoryEnabled/>
                <auditoryExternallyCued/>
                <auditoryCategory/>
                <cognitionEnabled/>
                <cognitionCategory/>
                <multiCognitionCategory/>
                <psychomotorPrimaryEnabled/>
                <psychomotorAlternateEnabled/>
                <miscellaneousEnabled/>
                <miscellaneousCategory/>
                <reportEnabled/>
                <primaryChannelMask/>
                <alternateChannelMask/>
            </ipComponents>
```

```
<ipTPM>
    <speedAccuracyTimeStressEnabled/>
    <speedAccuracyTimeStress/>
    <visualDetectionTimeStressEnabled/>
    <visualDetectionTimeStress/>
    <forgettingEnabled/>
    <forgetting/>
    <forgettingCopiedFromOperator/>
    <taskHistoryEnabled/>
    <taskHistory/>
    <taskPriorityEnabled/>
    <taskPriority/>
    <aptitudeEnabled/>
    <aptitude/>
    <experienceEnabled/>
    <experience/>
    <nonPrefLimbEnabled/>
    <nonPrefLimb/>
    <stressorOneEnabled/>
    <stressorOne/>
    <stressorTwoEnabled/>
    <stressorTwo/>
    <stressorThreeEnabled/>
    <stressorThree/>
    <stressorFourEnabled/>
    <stressorFour/>
    <stressorFiveEnabled/>
    <stressorFive/>
</ipTPM>
<ipScheduling>
    <priorityCategory/>
    <priorityMultiplierType/>
    <priorityMultiplier/>
    <taskPriorityTimePressure/>
    <priorityInterruptible/>
    <priorityResumable/>
    <prioritySheddable/>
    <priorityShedIfLate/>
    <criticalToMissionPerformance/>
    <criticalToMissionAwareness/>
    <criticalToSystemAwareness/>
```

```
                <sheddingConsequences/>
                <sheddingStartTaskID/>
                <sheddingStartTaskIDatShedTime/>
            </ipScheduling>
        </ipData>
        <popData>
            <inputDemand>
                <inputDemandValue/>
                <visualValue/>
                <auditoryValue/>
            </inputDemand>
            <centralDemand>
                <centralDemandValue/>
                <spatialValue/>
                <verbalNumericValue/>
                <otherValue/>
            </centralDemand>
            <outputDemand>
                <outputDemandValue/>
                <manualValue/>
                <vocalValue/>
                <popInterferenceMask/>
            </outputDemand>
            <timePressureDemand>
                <timePressureDemandValue/>
                <paced/>
                <priority/>
            </timePressureDemand>
        </popData>
    </SchedulerData>
    <taskCriticality>
        <safety/>
        <missionEffectiveness/>
        <efficiency/>
        <systemReliability/>
        <cost/>
        <operatorCapability/>
        <taskImprovement/>
        <adjustedCriticality/>
    </taskCriticality>
    <functionAllocation>
```

```
        <criterion>
            <weight/>
            <assignedTo/>
            <note/>
        </criterion>
    </functionAllocation>
    <assignedOperatorInfo>
        <taxonPercentages>
            <vigilanceTaxonPercentage/>
            <visualTaxonPercentage/>
            <auditoryTaxonPercentage/>
            <verbalTaxonPercentage/>
            <spatialTaxonPercentage/>
            <fineDiscreteTaxonPercentage/>
            <fineContinuousTaxonPercentage/>
            <grossMotorTaxonPercentage/>
            <vocalTaxonPercentage/>
        </taxonPercentages>
        <assignmentType/>
        <operatorAssigned/>
        <operatorAssignmentExpression/>
    </assignedOperatorInfo>
    <decision>
        <type/>
        <dispProps/>
    </decision>
    <compositeID/>
    <name/>
    <meanTime/>
    <stdDev/>
    <distType/>
    <criticalTask/>
    <taskType/>
    <osdType/>
    <normalizedTask/>
    <machineTask/>
    <beginningEffects/>
    <endingEffects/>
    <releaseCondition/>
    <schedEffects/>
    <cycleTime/>
```

```
        <cycleTimeStDev/>
        <difficulty/>
        <taskStartEventType/>
        <startEvent/>
        <taskEndEventType/>
        <endEvent/>
        <probabilityOfFailure/>
        <location/>
        <width/>
        <height/>
        <disProps/>
        <microSaintDecisionType/>
        <microSaintQueueOrder/>
    </task>
    <!--<starter>
        <location/>
        <width/>
        <height/>
        <next_id/>
        <points/>
    </starter>-->
</project>
```

DOCUMENT CONTROL DATA

*Security markings for the title, authors, abstract and keywords must be entered when the document is sensitive

1. ORIGINATOR (Name and address of the organization preparing the document. A DRDC Centre sponsoring a contractor's report, or tasking agency, is entered in Section 8.)

DRDC – Toronto Research Centre
Defence Research and Development Canada
1133 Sheppard Avenue West
Toronto, Ontario M3K 2C9
Canada

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

CAN UNCLASSIFIED

2b. CONTROLLED GOODS

NON-CONTROLLED GOODS
DMC A

3. TITLE (The document title and sub-title as indicated on the title page.)

LIGHT ARMOURED VEHICLE CREW MODEL DEVELOPMENT

4. AUTHORS (Last name, followed by initials – ranks, titles, etc., not to be used)

CAE, B. C.; CAE, A.L.

5. DATE OF PUBLICATION (Month and year of publication of document.)

July 2018

6a. NO. OF PAGES (Total pages, including Annexes, excluding DCD, covering and verso pages.)

194

6b. NO. OF REFS (Total references cited.)

16

7. DOCUMENT CATEGORY (e.g., Scientific Report, Contract Report, Scientific Letter.)

Contract Report

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

DRDC – Toronto Research Centre
Defence Research and Development Canada
1133 Sheppard Avenue West
Toronto, Ontario M3K 2C9
Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

114032-003

10a. DRDC PUBLICATION NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC-RDDC-2018-C155

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11a. FUTURE DISTRIBUTION WITHIN CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

Public release

11b. FUTURE DISTRIBUTION OUTSIDE CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

12. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Use semi-colon as a delimiter.)

vehicle crews; agent-based simulations

13. ABSTRACT/RÉSUMÉ (When available in the document, the French version of the abstract must be included here.)