



Human Factors Research Task 2006-111: AIMSsim Feature Development II

Final Report

*Oliver Schoenborn
CAE Professional Services*

*CAE Professional Services
1135 Innovation Drive, Suite 300
Ottawa, Ontario K2K 3G7*

Project Manager: O. Schoenborn, 613-247-0342 x217

Contract Number: W7711-047904/TOR/001

Contract Scientific Authority: J. Crebolder, 902-426-3100 x296

Defence R&D Canada – Atlantic

Contract Report
DRDC Atlantic CR 2007-302
January 2008

This page intentionally left blank.

Human Factors Research Task 2006-111: AIMSsim Feature Development II

Final Report

Oliver Schoenborn
CAE Professional Services

CAE Professional Services
1135 Innovation Dr., Suite 300
Ottawa, ON
K2K 3G7

Project Manager: O. Schoenborn, 613-247-0342 x217

Contract number: W771-047904/TOR/001

Contract Scientific Authority: J. Crebolder, 902-426-3100 x296

Defence R&D Canada – Atlantic

Contract Report

DRDC Atlantic CR 2007-302

January 2008

Author

Oliver Schoenborn

Approved by

Original signed by Jacquelyn Crebolder

Jacquelyn Crebolder

Approved for release by

Original signed by James L. Kennedy

James L. Kennedy

© Her Majesty the Queen as represented by the Minister of National Defence, 2008

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2008

Abstract

This document is the final report for Human Factors Research Task 2006-111 – AIMS Feature Development which consisted of software development for a research platform designed to simulate the operator-machine interface and controls for an airborne multi-sensor surveillance system. The research platform is used to evaluate design concepts and operator performance issues and is modelled on the Advanced Integrated Multi-sensor Surveillance (**AIMS**) system (formerly known as the Enhanced Low-Light Level Visible and Infrared Surveillance System – **ELVISS**) which is being developed at DRDC Valcartier. With up to five sensors, the AIMS will enhance the capability of search and rescue (**SAR**) by allowing crews to operate effectively at night and in degraded weather conditions. The report summarizes the work performed in the project and makes recommendations for the next phase. Core tasks included adding functionality to over-ride the default sensor controller on the control box and replace it with a universal serial bus (USB) controller, and adding the capability to rearrange the sensor windows and other features on the interface display and other areas on the interface. Both capabilities were added for the purpose of evaluating tools and interface design concepts.

Résumé

Le présent document est le rapport final sur la tâche de recherche sur les facteurs humains 2006-111 – Développement des caractéristiques du système perfectionné de surveillance multi-capteurs intégré (**AIMS**) – qui consistait à élaborer le logiciel pour une plate-forme de recherche conçue pour simuler l'interface opérateur-machine et les commandes d'un système de surveillance multi-capteurs aéroporté. La plate-forme de recherche sert à évaluer les principes de conception et le rendement de l'opérateur et est dérivée du système AIMS (anciennement appelé Système perfectionné de surveillance à intensification de lumière visible et à infrarouge ou **ELVISS**) qui est en cours de développement à RDDC Valcartier. Équipé de deux à cinq capteurs, le système AIMS améliorera la capacité de recherche et sauvetage (**SAR**) en permettant aux équipages de travailler avec efficacité la nuit et par intempérie. Le rapport résume le travail effectué dans le cadre du projet et fait des recommandations pour la prochaine phase. Les tâches de base comprenaient l'ajout de fonctions pour surpasser le contrôleur des capteurs par défaut sur le boîtier de commande et le remplacer par un contrôleur à bus série universel (USB) et l'ajout de la capacité à réaménager les fenêtres de capteur et d'autres caractéristiques sur l'affichage d'interface et d'autres zones de l'interface. Les deux capacités ont été ajoutées dans le but d'évaluer des outils et les principes de conception de l'interface.

This page intentionally left blank

Executive summary

Introduction

To enhance the capability of airborne search and rescue (**SAR**) and general surveillance a multi-sensor electro-optical imaging system, the AIMS (Advanced Integrated Multi-sensor Surveillance) system is being developed by Defence Research & Development Canada (**DRDC**).

The system consists of several sensors including electro-optical (EO) sensors, an Active-Gated TV (**AGTV**), and Forward-Looking Infra-Red (**FLIR**). When combined in one system the suite will effectively extend the capability of search and rescue and surveillance patrol missions beyond the limitations of conventional daylight operation. Sensors are slaved together and view the same scene and are operated from the cockpit of an airplane using a joystick and controls.

To ensure optimal performance the AIMS system requires an appropriate interface and controls, the design of which must realize the interaction between technological capability and operator performance. A research platform has been developed to provide a means for studying operator issues using the system. This report documents software modifications that have been implemented to upgrade the research platform.

Results

The main deliverables from this tasking are updated software executable, source code, and associated system and user manuals for the *AIMSsim* software, and this report.

Work related to the Scenario Generation Environment (SGE) and simulation management were combined in one open-ended, more generalized activity of defining and prototyping a next generation of the SGE that would integrate the *AIMSsim* and a Graphical User Interface (GUI) -based scenario manipulation environment called the AIMS Simulation Experimentation Station (ASES).

The “world hierarchy” feature was replaced by a temporary change in the terrain clamping of targets, based on discussions with client. A future phase may look at a better solution to the issue, but the temporary measure should be sufficient in the medium term.

The following capability was added:

The capability to support USB joysticks to override only part of the FlyPanel control box;

- Configurability of the GUI via text dialog screens and sensor display layout;
- Configurable thumbnail selection bar;
- Several new types of sensors;
- A high-level specification of ASES.

Due to time constraints the open-ended ASES prototyping and lower-priority items such as “setup development environment” were not implemented as fully as originally desired but brought to a satisfactory conclusion nonetheless.

Significance

The continued development and upgrade of the *AIMSsim* research platform provides the AIMS experimental team with an appropriate level of simulation detail to conduct human performance analyses which in turn delivers up-to-date knowledge and advice on the design of sensor surveillance systems to the military stakeholder.

Future Plans

Recommendations for the next phase include implementing a visualization of the sensor path's history on the moving map display, providing a debug tool, upgrading the scripting language, and facilitating the copying and use of old scripts.

Schoenborn O., 2007; AIMSsim Feature Development II. DRDC Atlantic CR 2007-302. Defence R&D Canada – Atlantic.

Sommaire

Introduction

Pour améliorer la capacité de recherche et sauvetage (**SAR**) par aéronef et la surveillance générale, un système d'imagerie multi-capteurs électro-optique, le système AIMS (système perfectionné de surveillance multi-capteurs intégré), est en cours de développement à Recherche et développement pour la défense Canada (**RDDC**).

Le système comprend plusieurs capteurs, y compris des capteurs électro-optiques, un capteur de télévision commandée par portes actives (**AGTV**) et une caméra infrarouge à balayage frontal (**FLIR**). L'ensemble de ce système augmente la capacité de recherche et sauvetage et des missions de surveillance au-delà des limitations des opérations classiques sous la lumière du jour. Les capteurs sont asservis les uns aux autres et observent une même scène; ils sont commandés à partir du poste de pilotage d'un avion au moyen d'une manette et d'autres commandes.

Pour assurer des performances optimales, le système AIMS exige une interface et des commandes appropriées, dont la conception doit réaliser l'interaction entre la capacité technologique et le rendement de l'opérateur. On a développé une plate-forme de recherche qui permet d'étudier les questions liées à l'opérateur du système. Le présent rapport décrit les modifications logicielles qui ont été mises en œuvre pour mettre à niveau la plate-forme de recherche.

Résultats

Les principaux produits à livrer de cette attribution de tâches sont un logiciel mis à jour, un code source exécutable, les manuels de système et de l'utilisateur connexes pour le logiciel *AIMSsim* et le présent report.

Le travail en rapport avec l'environnement de génération de scénario (SGE) et la gestion de la simulation ont été combinés en une seule activité évolutive plus généralisée de définition et de prototypage d'une prochaine génération de SGE qui intégrerait le logiciel *AIMSsim* et un environnement de manipulation de scénario basé sur interface graphique et appelé poste d'expérimentation de simulation AIMS (ASES).

La caractéristique de l'« hiérarchie des mondes » a été remplacée par un changement temporaire du verrouillage des cibles au terrain, en fonction de discussions avec le client. Une phase future peut consister à étudier une meilleure solution du problème, mais la mesure temporaire devrait suffire à moyen terme.

On a ajouté les capacités suivantes :

- Prise en charge des manettes USB pour surpasser une partie seulement du boîtier de commande FlyPanel;
- Configurabilité de l'interface graphique au moyen d'écrans de dialogue et de la présentation de l'affichage des capteurs;
- Barre configurable de sélection de vignettes;
- Plusieurs nouveaux types de capteurs;

- Spécification de haut niveau du poste ASES.

À cause des délais, le prototypage évolutif du poste ASES et des articles de priorité inférieure tels que l'environnement de développement de montage n'ont pas été mis en œuvre dans la mesure souhaitée à l'origine, mais ont quand même été menés à une conclusion satisfaisante.

Portée

Le développement et la mise à niveau continus de la plate-forme de recherche *AIMSsim* fournit à l'équipe de l'expérience AIMS un niveau de simulation assez détaillé pour mener des analyses du rendement humain, qui fournissent à leur tour à l'intervenant militaire des connaissances à jour et des conseils au sujet de la conception de systèmes de surveillance à capteurs.

Recherches futures

Les recommandations pour la prochaine phase comprennent la mise en œuvre d'une visualisation de l'historique du déplacement des capteurs sur l'affichage cartographique défilant, la fourniture d'un outil de débogage, la mise à niveau du langage de script et la facilitation de la copie et de l'utilisation d'anciens scripts.

Table of contents

Abstract.....	i
Executive summary	iii
Sommaire.....	v
Table of contents	vii
1. Introduction	1
1.1 Background	1
1.2 Task Objective.....	1
1.3 This Document	1
2. Deliverables.....	2
3. Summary of Work Performed	3
4. Future Work Recommendations.....	8
References	10
Annex A: Specification and Design of ASES	11
Bibliography	22
List of Abbreviations and Acronyms.....	23
Distribution List.....	25

This page intentionally left blank

1. Introduction

1.1 Background

To enhance the capability of airborne search and rescue (**SAR**) and general surveillance a multi-sensor electro-optical imaging system, the AIMS (Advanced Integrated Multi-sensor Surveillance) system is being developed by Defence Research & Development Canada (**DRDC**).

The system consists of several sensors including electro-optical (EO) sensors, an Active-Gated TV (**AGTV**), and Forward-Looking Infra-Red (**FLIR**). When combined in one system the suite will effectively extend the capability of search and rescue and surveillance patrol missions beyond the limitations of conventional daylight operation. Sensors are slaved together and view the same scene and are operated from the cockpit of an airplane using a joystick and controls.

To ensure optimal performance the AIMS system requires an appropriate interface and controls, the design of which must realize the interaction between technological capability and operator performance. A research platform has been developed to provide a means for studying operator issues using the system. Software modifications have been implemented to upgrade the research platform.

1.2 Task Objective

The objective of this Call-Up was to develop and improve the features of the AIMS simulation software.

The main deliverables from this tasking are the updated software executable, source code, and associated system and user manuals for the *AIMSsim* software, and this report, which summarizes the work done in this project.

1.3 This Document

This document is the final report for *HFR Task 2006-111: AIMSsim Feature Development II*. Section 2 of this document gives details on the deliverables of this project, section 3 summarizes the work performed, and section 4 concludes by making some recommendations for the next phase of work.

2. Deliverables

The deliverables for this project consist of

- Updated AIMSSim application package: zipped file containing executables, experiments, and supporting DLL's, data (geometry, textures, shader files, etc), documentation, and experiments
- Updated source code of AIMS HMI prototype (packaged Subversion database file)
- Updated User and System manuals (Microsoft Word and PDF formats)
- Final report (this document), outlining any functionality that was affected and any future requirements

These have been saved onto a CD, and sent along with a hardcopy of this document.

3. Summary of Work Performed

Work for this project commenced January 15, 2007, and was originally scheduled to end in May 2007. The SOW was amended in May to set the target end date to August 31st, 2007, without any change to the budget, thus spreading the work to be performed over more months so as to overlap with summer staff. The actual end date was September 30, 2007.

Each work task identified in the SOW is described below. More details can be found in the User and System manuals.

1. **Multiple USB joysticks:** This consisted in adding the ability for the experimenter to override the FlyPanel's joystick control (including its two triggers and POV hat) with a USB joystick connected to the PC.

The support for USB joystick override was tested only with a popular model of a Logitech 3D Extreme joystick, but should carry over to a large spectrum of USB joysticks thanks to the use of the advanced DirectInput API provided by Microsoft. Support for other joysticks should be straightforward, involving one of 1) no extra work, works "out of the box"; 2) reconfiguring the USB joystick to match the functionality expected by the system, via the vendor's software; 3) improving the AIMSSim software (specifically, *simInputs*) to better support new joystick models incompatible with current mainstream joysticks.

2. **Fillet problem:** This consisted in fixing a minor problem with the path following algorithm, which caused incorrect aircraft or target motion in the joining segment of two path plans (*path following* is discussed in section 4.2.10 – 4.2.12 of the User Manual). The identification of the cause of the issue was more work than the solution. Fixing this problem allowed the removal of a temporary hack introduced at the end of the last phase during the experiment development in which random circular paths were being generated and strung together. Along with the solution to the problem, the fillet radius adapts to path joins more effectively.

3. **Acceleration rate:** This consisted in adding the ability for the experimenter to specify the acceleration rate on a path plan segment, rather than having it hardcoded. Moreover, the rate is a true acceleration rate (i.e. speed per unit time) rather than per time step.

4. **World hierarchy:** The objective of this work item was to identify the source of a problem noticed by the client in the visual system but not reproducible at CAE PS due to the challenging nature of the problem, which made it difficult for the client to capture and demonstrate. Many exchanges with the client helped determine that the visuals were fine but that they had conflicting requirements not yet fully understood.

As a temporary measure, the ground clamping was improved to make it configurable via the experiment scripts by allowing the experiment to choose between highest and lowest clamping. Search scenarios, where an operator

searches for a target of interest in the ground terrain, can use lowest clamping, which will cause the targets to be under the tree canopy. This feature allows for more true to life scenarios where targets might be camouflaged and hidden by the terrain. In target tracking scenarios a target is tracked with the sensor and a tree canopy might be less desirable. In this case highest clamping can be used which will position the targets on top of the tree cover.

The main disadvantage of this temporary solution is that target tracking scenarios produce unrealistic behaviour when a target is moving on terrain and crosses roads, which in the current database do not have tree cover, thereby causing the target to move down and up as they are crossing the road.

The full solution would involve separating the tree cover in the terrain database from the actual terrain, and adding the ability for the system to attach geometry (such as the canopy) to the terrain. This would allow the experiment to select a bare terrain (no tree canopy) for target tracking experiments, and terrain+canopy for search experiments, both using lowest-ground clamping.

The client decided that the temporary solution would be adequate until the next phase of work, but that the full solution could be implemented if there was time remaining at the end of this project. There was no extra time available to implement the full solution.

5. **Scripting in display:** This consisted in adding the ability for the system developers (currently CAE PS) to make certain parts of *simDisplay* configurable via a configuration file, and for the experiment developers to make the Operator View configurable (the Operator View represents the Operator Console of the real system, i.e. shows sensor displays, a moving map display, and thumbnail selection bar).

CAE PS added a Lua interpreter to *simDisplay*, as was added to *simControl* many phases ago. However, contrary to *simControl*, this was added via a code generation tool, called *tolua++*, providing a much more object-oriented interface into the application's properties and behaviour. This required more time than planned as *tolua++* had never been used before in the system. The *simDisplay* will look for the configuration script, called *displayConfig.lua*, in the experiment's folder, but uses a default setup identical to the view used before this feature was added if the configuration file is not found.

Exporting new functions in a type-safe, robust and object-oriented manner is very straightforward using this tool and will support future extensions to *simDisplay* configurability. It is so powerful in fact that CAE PS recommends replacing the raw Lua API calls made in *simControl* by *tolua++* generated code, which will make the access to system constants, variables and functions easier, less verbose, and supportive of larger experiments.

Configuration of the Operator View was added such that the view can be divided into a hierarchy of subdisplays using an intuitive and simple ratio system, providing a large degree of flexibility in the view configuration. It allows the experimenter to decide not only the placement and size of the MMD, main and auxiliary sensor display ports, but also which sensors will be available to the

operator. The Display Selector Bar (see next item) can therefore appear almost anywhere on the display.

6. **Display thumbnails:** This consisted in adding a set of configurable, mouse-selectable sensor display thumbnails in the Operator View. The thumbnails are placed in a “bar” called the Display Selector Bar, which orients itself vertically or horizontally based on the aspect ratio given to the bar in the configuration file: if the bar is wider than it is high, a horizontal bar is created, and vice versa for a vertical bar. The five sensor displays (see next item) are visible in the bar and can be selected with the mouse. EO-type displays are shown in the *main* display port, whereas IR-type displays are shown in the *auxiliary* display port.

The notion of *main* and *auxiliary* display ports is new to the system and was necessary to allow for a convenient way of deciding where each sensor display should be viewed when selected. It was arbitrarily decided that EO-type sensors appear in the main port, whereas IR-type sensors go to the auxiliary port.

In the next phase, CAE PS recommends that the notion of primary versus secondary sensors be revisited, as it only works for a system having only two sensors. An N-sensor system, $N > 2$ as is now simulated in AIMSSim, could support the notion of one master versus $N-1$ slave sensors. It is not yet clear what the differences would be between master and slave sensors.

7. **Add display types:** This consisted in adding three new display types, each specified very broadly by the client in terms of EO/IR capabilities, zoom type and fields of view. The sensor modelling in AIMSSim was generalized to support future changes or additions of types of sensors.

Currently two types of sensors are supported by the system, namely “EO-type” or “IR-type”. The former maintains the de facto “EO” misnomer to refer to sensors that sense radiation in the visible part of the spectrum, whereas the latter are defined as sensors that sense radiation in the (non-visible) infra-red-and-beyond. If sensors are ever created for “above visible” light, such as UV and beyond, a third category could easily be created for “UV-type” sensors.

All sensors share a common set of capabilities. IR-type sensors have in addition the ability to change polarity, whereas EO-type sensors have in addition the ability to change to greyscale. In this new scheme the AGTV sensor from the previous version of AIMSSim is an EO-type sensor with active gating capability, i.e. a laser illuminator, while the FLIR is an IR-type sensor.

As per the client specification received during the project, two new EO-type sensors were added, and one IR-type sensor, a Thermal Imager (THIR). All types of sensors share the ability to have different types of zoom and fields of view, specifiable in the experiment scripts. It is therefore the experimenter’s responsibility to assign zoom types and field of views and constrain them as (and if) necessary during the experiment. All sensor types have a shader file that defines how their image is rendered.

8. **Dialog customization:** This consisted in adding the ability for the experimenter to define new dialog screens wherein they could post a question on the screen to the

operator, and capture a mouse click on buttons displayed in the dialog. The system supports the addition of any number of dialog screens containing text and buttons, with experimenter-specified event names to be used in their Finite-State machine for changing the course of the experiment or saving responses/data.

9. **SGE cleanup:** This was originally included in the SOW to allow for better coherence between the scenario generation environment (SGE) and the simulation application (AIMSsim). This weakened coherence was a result of a major rework done to the system prior to CAE PS's involvement in its porting and subsequent development. This made it difficult to ensure that values saved in files generated by the SGE matched the values expected by AIMSsim. As part of the combination of items 10 and 11 below into a "next generation" SGE (see below), this work item was abandoned and its hours informally transferred to other tasks on an as-needed basis.
10. **SGE usability:** This originally consisted of improving the current SGE to support the new capabilities of AIMSsim, added over the past two years. However, discussions with the client after the project started showed that the SGE, a legacy application almost 10 years of age, had become impractical to update and that it would be preferable to spend the effort specifying and designing a "next generation SGE".

This new GUI would support the creation and generation of scenarios as well as the management and monitoring of experiments (described in work item 11, below), making use of the current *simControl*, *simInputs* and *simDisplay* components. In this report the proposed application is referred to as the *AIMS Simulation Experimentation Station (ASES)* as it will support all aspects of the simulation-based experimentation of AIMS in one integrated application.

This work item was therefore modified to include the specification, design, and a preliminary implementation of the ASES application, limited by the constraint that the total project money could not be changed. The preliminary implementation is therefore at a rather early stage but demonstrates the integration of Python, OpenGL and wxAUI as effective implementation components for the system. The specification and design are described in the appendix.

11. **Manage simulation:** This originally consisted of creating a GUI to simplify the experimenter's task of starting and stopping an experiment and monitoring the log files for warnings or errors (especially during start-up and experiment transitions). As described in item 10 above, this work item was merged into the modified item 10, i.e. the specification and design of the ASES application, described in the appendix.
12. **Releases and Quality Assurance (QA):** This consists in the delivery over FTP of "releases" of the software as often as possible so the client can test and provide feedback. Several releases were made available to the client. Some unplanned effort was spent on the transport of the release over FTP since the FTP server used in the previous phase was no longer available and the new one could not be used, presumably due to firewall restrictions on both the CAE PS and client sides.

DRDC IT explicitly changed their firewall to allow CAE PS to upload files to their FTP site. The client was able to test each release and provide feedback or requirements for fixes.

13. **Final deliverables:** This consists of all the closure-related activities of the project, such as producing the CD with the deliverables and shipping, writing the final report, creating closure documents such as feedback forms, etc.
14. **Script support:** This was an unspecified task in the event that the client may need support in writing scripts for their new experiments or to test new features of the system.
15. **Update manuals:** This consisted in updating the System and User manuals to cover the new features and capabilities of the system. A significant amount of documentation was added, in some cases covering features that had been added in the previous phases but had been only partially documented.
16. **Additional effort as required:** This was an unspecified task which was not required.
17. **Setup and development environment:** This consisted of installing upgraded versions of several third-party libraries used in the building of AIMSSim. FLTK, boost, and several others were upgraded.

The upgrade to OpenSceneGraph (OSG) 1.2 was started but interrupted due to a critical bug in that version of the library that prevented the use of shaders (used for all sensor displays to model grey scaling, noise, fog, laser illumination, etc) in the AIMSSim sensor displays.

An upgrade to OSG 2.0 was successful just before the end of the project: no changes to the code were necessary to build and run AIMSSim using the OSG 2.0 distribution. However, there wasn't sufficient time to fully test the system to minimize the chance of broken features, hence the AIMSSim delivered to the client is still using OSG 1.0.

Over the past 8 months the OSG library has progressed to version 2.2, so another attempt to upgrade should be repeated in the next phase. The upgrade itself should be painless, however some time should be set aside for proper testing, and for replacing some of the code (if any) that uses deprecated OSG classes.

4. Future Work Recommendations

For the next phase, CAE PS recommends the following work items. These recommendations should be taken into consideration in light of any other client requirements.

Operator-Machine Interface (OMI):

1. Complete a prototype for ASES. This could involve item 6 below, as well as allowing the *simControl* and *simDisplay* to be “reset” without having to restart or reload the terrain (the most time-consuming aspect of start-up and resets).
2. Test with more USB joysticks: test and fix as required to support a specified set of USB joysticks.
3. Add sensor history to MMD: add the history of sensor coverage to MMD. This requires defining how sensor coverage is measured:
 - a. Which of the five sensors should be used, as they each have independent fields of view (possibilities are numerous, e.g. average of all FOVs, or a \$300 head-tracking device by NaturalPoint);
 - b. Is shadowing taken into account: uneven terrain limits the actual terrain seen in sensor displays, such that straightforward sensor FOV projection grossly overestimates how much ground was covered;
 - c. Is horizontal distance from aircraft taken into account: visibility distance, and pixel coverage ratio can severely limit how much is visible in the display and grossly overestimate how much ground was covered;
 - d. how often it should be recorded (every second, 30 times per second etc);
 - e. how long is the history kept for, e.g. 10 frames, 2 minutes, the whole experiment duration; impacts the memory, data storage and rendering performance characteristics.
4. Upgrade *simDisplay* to most recent stable version of OpenSceneGraph and replace old code that depends on Producer with new osgViewer-based code.
5. Add the ability to configure the sensor overlay text from display configuration script.
6. Add the ability to turn on some features which are currently available only to developers in debug builds (such as designation boxes, target placement beams, non-root window).

Scripting:

7. Add world hierarchy: split the terrain database and add capabilities to attach geometry to terrain so that canopy would be in a separate file that would be included by experimenter only in relevant experiments.
8. Revise notion of Primary versus Secondary displays.
9. Upgrade Lua scripting in *simControl*: upgrade Lua scripting engine in *simControl* to use tolua++ and provide a more powerful extension mechanism and experiment definition framework.
10. Facilitate copying of experiments: copying an experiment should not require the search of scripts for path names, which makes re-use of experiments very prone to errors.
11. Event-triggered Lua code: allow Lua code to be called in response to an event; currently a script must be run, which can cause a multitude of small scripts to be created and makes the experiment creation more difficult (may become a non-issue with ASES).

Documentation:

12. Rewrite documentation for online access.
13. Rewrite documentation to change presentation of system via ASES rather than via backend components (*simControl* etc).

References

1. Baker, K., & Youngston, G. (2006). *Advanced Integrated Multi-sensor Surveillance (AIMS) Operator-machine interface definition study*. CAE Professional Services. Defence Research and Development Canada - Toronto Contract Report: DRDC Toronto CR 2006-242.
2. McFadden, S., Crebolder, J., & Larochelle, V. (2006). *Development of an operator-machine interface for ELVISS Final Report*. Defence Research and Development Canada - Toronto Technical Report: DRDC Toronto TR 2006-055.
3. Schoenborn, O. (2007a). *AIMSsim User Manual*. CAE Professional Services, Ottawa, Ontario. Defence Research and Development Canada - Atlantic: DRDC Atlantic CR 2007-300.
4. Schoenborn, O. (2007b). *AIMSsim System Manual*. CAE Professional Services, Ottawa, Ontario. Defence Research and Development Canada - Atlantic: DRDC Atlantic CR 2007-301.

Annex A: Specification and Design of ASES

This appendix gives a preliminary specification and design for the AIMS Simulation-based Experimentation Station (ASES) application.

The development of ASES is motivated by the fact that the current Scenario Generation Environment (SGE) is based on old technology and has fallen far behind in its capabilities relative to AIMSSim, and updating it would be more costly than starting from scratch. Moreover, the SGE is merely a generation environment, whereas the client has expressed a need for a testing and control environment.

The concept of operation formed through discussions and interactions with the client is the following:

ASES should make it easy for a novice user to create a flight plan, add some targets, assign path plans to them, and run the scenario, all from the GUI.

CAE PS recommends that ASES divide the above into three “task areas” available from one GUI:

1. Create a scenario: this task area is where the user is provided with tools to create paths and targets, define events and transitions between stages of the scenario, set attributes of the entities available, and facilitate the use of the exported AIMSSim functions available in the scenario scripts; ASES must do so in such a way that the user can decide at which stage of a scenario each of the above changes must take place.
2. Test the scenario, in preparation for experiment: this task area is where the user can run the experiment and get extra information that will help troubleshoot the scenario behaviour, e. g. the list of timed events queued, the list of periodic scripts currently active, etc, all items that are difficult (or not currently possible) to get from the scenario scripts but are crucial in helping determine why some data is not being saved or why the scenario is not transitioning to an expected stage. In addition, being able to see the log files and filter them would be useful. The AIMSSim would probably run as a non-root decorated window beside the ASES GUI.
3. Run the scenario for real experiment: this task area is where the user can run the experiment and see filtered views of the log files but not the other test tools. The AIMSSim would probably run on root window in front of the ASES GUI.

Eventhough tasks 2 and 3 share some graphical elements and capabilities, there are also significant differences that indicate they should be kept logically separate at the architectural level.

Scenario Creation

A scenario consists of a set of stages and transitions between some of them, and only one “INIT” stage and one “EXIT” stage (previously known as finite states). Scenario creation can involve any or all of the following activities:

Create/define:

1. Path plans: define sequences of waypoints, speeds, accelerations, radius of turn; define paths from joining smaller paths; no limit on number of plans allowed;
2. Targets: define placement (i.e. position and orientation) and geometry; number of targets allowed may be limited but high;
3. User event names
4. Scenario stage names
5. Tasks (previously known as periodic scripts)
6. Dialog screens
7. Operator View configuration
8. Global variables
9. Variables local to a scenario stage
10. Log files
11. Data files

Change attributes of:

1. Created targets, such as which sensor display a target is visible in, what is the target’s label, x, y, z position, etc
2. Sensor displays
3. Moving map display (MMD)
4. Other system attributes (participant number, etc.)

Incorporate commands in a scenario stage:

1. Action commands (send message to display, timed events, etc.)
2. Query commands (e.g. get sensor orientation)
3. Other commands (e.g. set sensor orientation)

It is not possible to know at what scenario stage the user will want a given “thing” (object, attribute, command, etc.) to be created or used. It would also be useful to allow the user to make use of system-defined names in their scenarios. The following strategy can be used to facilitate this:

1. An “Entities” panel shows all the above “things” that can be created or defined, plus all the system entities. The latter cannot be deleted or renamed.
2. This “Entities” panel also shows all the attributes for the given entities. Each entity has a “preview” panel that can be made visible. The attributes can be changed in the tree, and the corresponding “preview” panel shows how this would affect the entity.
3. The user can drag entities and entity attributes from the “Entities” panel onto a “Stage Definition” panel, thereby creating an entry in their scenario stage for setting that attribute. Further changing the attributes in the tree does not affect the scenario stage command just created.

The default “Entities” view would start as the following tree:

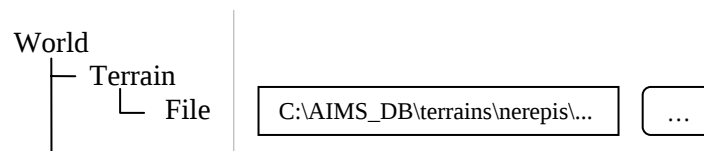
```
Global variables
OMI
  Dialog screens
  Operator View
  ROOT
  MMD
    Up orientation (North, Aircraft, Camera)
    ...
    AGTV sensor footprint visibility
  Sensor displays
    AGTV
      Color
      ...
      Laser illuminator on/off
    FLIR
      White is hot
      ...
      ...
World
  Terrain
  File
  Aircraft
    Position (x,y)
    Camera orientation (hpr)
    ...
    Bearing
  Targets
  Path plans
Events
  START_PRESSED
  OPERATIONAL_UPDATED
  ...
  BUTTON_TLL
  BUTTON_TCL
  ...
  BUTTON_BRR
Stages
  INIT (init)
  EXIT (exit)
Tasks
Log files
External scripts
Commands
  SendMessage
  ...
  SetLogLevel
```

A variety of actions would be possible on the above entities:

1. Left-click: select the entity; this would probably only make sense on entities that have attributes, such as sensor display entities (AGTV etc), MMD, targets, etc or on entities that can be dragged onto the editing areas (see below)

2. Right-click: show a pull-down context menu with available operations on the entity. E.g. right-clicking on 'Targets' node would show 'Create new target', amongst others. Right-clicking on a target would show 'Delete target'. Not all entities support operations. E.g. what can be done to a sensor display?
3. Left-click drag: dragging something from the tree onto another panel of the GUI, such as the "Stage Definition" panel; this would be the mechanism of choice to define scenario stages.

The entities window maintains a "state", i.e. shows the current value(s) of all attributes in editable text boxes in the window. This state can be reset without affecting the scenario stages, as it is merely for "previewing" to help the User decide what to put in the scenario stages. E.g. after selecting a terrain the Entities window might show

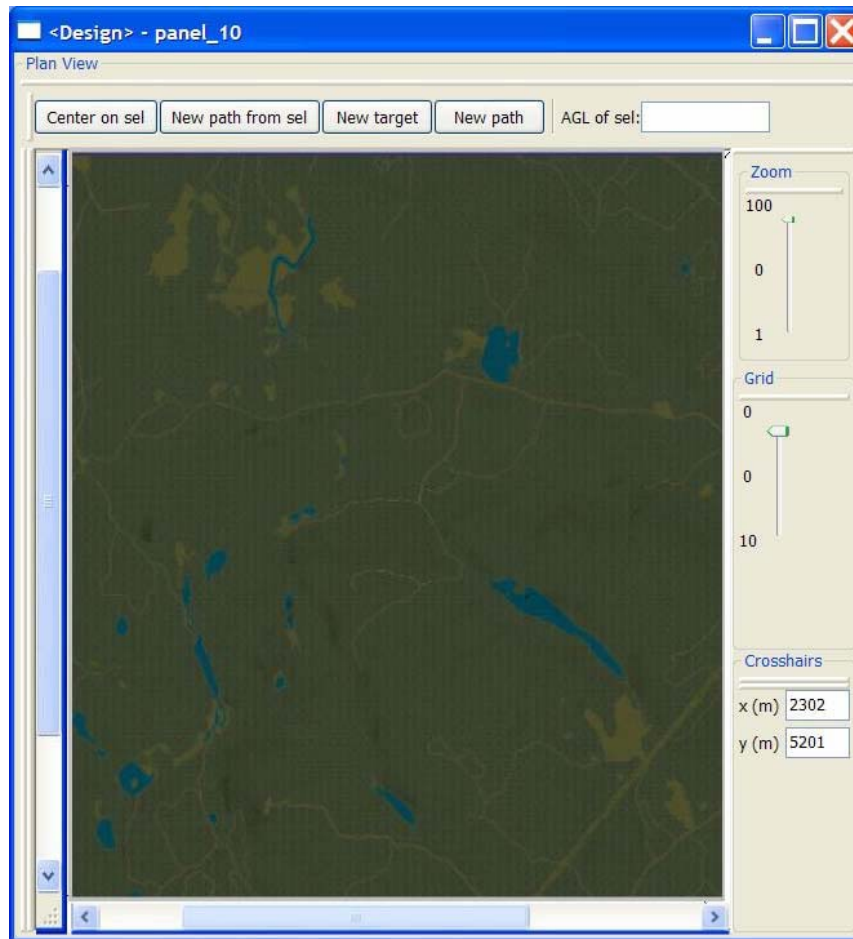


And the World preview pane would show the terrain.

A full specification of ASES would take all attributes and commands detailed in the User manual and define where in the Entities window they would appear.

The following explains some use cases:

1. Define terrain
 - a. Select "World->Terrain->File" node's browse button; ASES shows file browser;
 - b. Select terrain from file browser;
 - c. OK accepts the selection and loads the terrain; cancel abandons the selection of a terrain, leaving the previous selection, if any, loaded;
 - d. Once you see the terrain loaded and you are satisfied that this should be used in scenario, drag the node onto the scenario stage in which it should be defined, e.g. the INIT stage's node; alternately, you can right-click the File node and select "send to stage" and "INIT";
 - e. ASES shows the INIT node's EZedit panel with the new command for terrain.

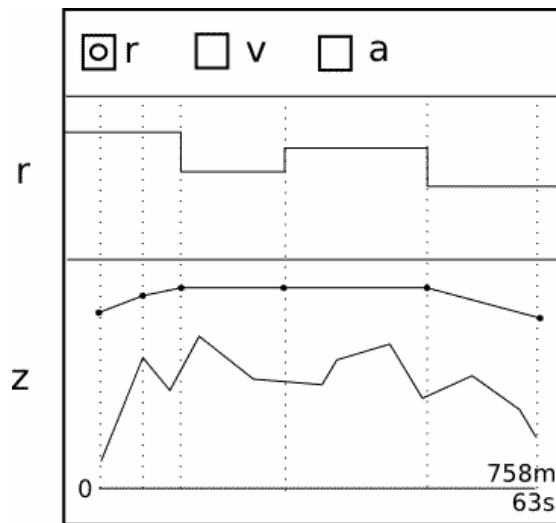


2. Create target:

- a. Right-click on Targets in Entities panel and select “Create new target”; this is only available if a terrain has been defined;
- b. ASES shows the World preview panel and changes cursor shape when mouse over World, indicating a target is about to be “placed”;
- c. Scroll World view until satisfied; click on map where target should be positioned;
- d. ASES adds a node as child of Targets with new Target entity and makes visible its attributes, such as name, label, position, etc;
- e. You can make changes to any of the attributes and the changes get immediately reflected in World view;
- f. Once satisfied with the settings, you drag the Target node to the stage where it should be created (or, right-click, Send to stage, select the stage);

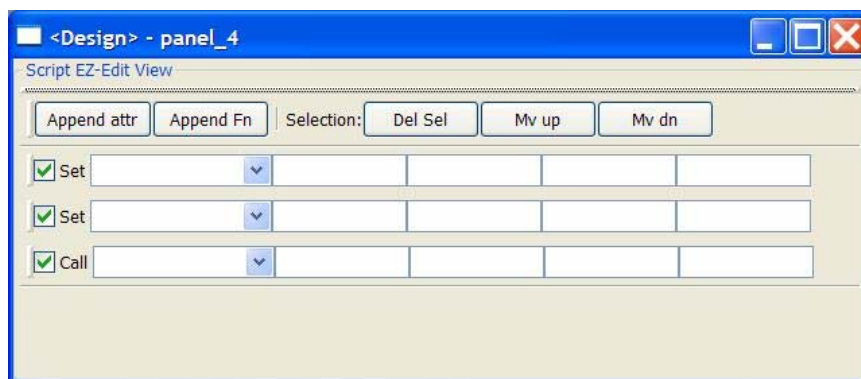
- g. Drag the new target's attribute nodes to the appropriate scenario stage(s); multiple selections are allowed at once;
 - h. You can edit the Target's attribute nodes in the Entities panel, this updates the World view but not the entries added to the scenario stages; or you can move the target in the World view, this updates the target's position attribute in the Entities window, but doesn't affect the scenario stage data; drag the new values to the desired scenario stage to make use of new values in scenario; ASES should mark edited attributes that have not been "sent to" a scenario stage to help you remember what you have edited but not yet "committed" to the scenario;
 - i. You can edit the Target's attributes in the EZedit window, this does not affect the values in the Entities window; for that, right click on the EZedit entry and select "send to preview";
3. Change target visibility in the AGTV sensor display
- a. Bring up the preview for the AGTV sensor display; this will only be possible if a terrain has been loaded;
 - b. Verify on the World view that the aircraft is at a convenient location: drag the aircraft to a good position, not too far from the target;
 - c. Use the FlyPanel controls to move the camera to bring the target into view;
 - d. In the target's attributes area of the Entities window, change its visibility attribute (e.g., color, etc); observe the effect in the AGTV preview;
 - e. When satisfied, drag the attribute to the desired scenario stage where the attribute should be set;
 - f. If you want the aircraft to go back to where it was, find the stage in which the aircraft position is set, and do "send to preview";
 - g. Hide any windows you don't need.
4. Find in what stages an attribute is changed
- a. Right-click on the attribute in the Entities window and select "find where changed";
 - b. ASES brings up the EZedit window for the first stage that it finds the attribute gets changed;
 - c. Repeat to find the next occurrence.
5. Create a path plan

- a. Right-click the Path Plans node in Entities window and select “Create new simple”;
- b. ASES shows the World view and the cursor changes shape to indicate that the next click on the World will position the path’s first waypoint;
- c. Click on the view to position first waypoint; ASES shows a waypoint and a line that joins the new waypoint to the mouse cursor, indicating that the next click on map will position next waypoint and line is the path segment between the two waypoints;
- d. As the path is created, ASES updates the “Path profile” preview, which gives a visual representation of the attributes of the sequence of waypoints;



- e. Drag the node for the path plan from Entities window and drop it onto the stage where the path plan should get created.
6. Create a composite path plan
 - a. Right-click the Path Plans node in Entities window and select “Create new composite”;
 - b. ASES adds a node in the Entity window;
 - c. Drag other plans onto the new node to create the sequence of plans to follow.
 7. Assign a path plan to a target and/or aircraft
 - a. Create a path plan;

- b. Drag the path plan's node (in Entities window) onto the target(s) and/or aircraft nodes (also in Entities window) to make those vehicles follow the plan; ASES shows the total plan assigned so far in the World view;
 - c. Verify on the world view that this really is what you want, make any adjustments to the plan object (e.g. by moving waypoints on the World view) ;
 - d. Drag the "Path plan to follow" attribute of the target into the scenario stage where the target should be given the plan;
 - e. Determine at what stage the target should actually follow the path plan;
 - f. Drag the "Resume motion" command from target node on the stage when target should start moving along path.
8. Create a new scenario stage
- a. Right-click on Stages in Entities panel and select "Create new stage";
 - b. ASES creates a new entry in the tree and allows you to name it; the name must be different from an existing one;
 - c. If desired, select new stage in tree to bring up its EZedit window, and edit using the window's tools; alternately, drag nodes from the Entities window onto the EZedit window of the stage to copy them over into the scenario.



9. Define a new transition between two stages
- a. Right-click the node of the "from" stage in Entities panel and select "Add transition"; this adds an entry with a "to" box, and an "event" box;
 - b. Drag one of the other stages onto the "to" box to fill it with that stage's name;

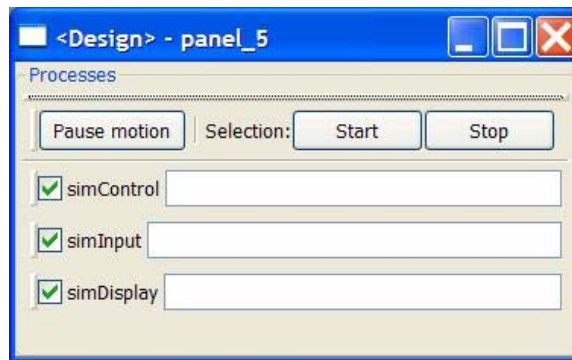
- c. Drag one of the Event nodes onto the “event” box to fill it with that stage’s name;
- d. If available, the Stages preview panel should show the updated stage-transition diagram.

Scenario Testing

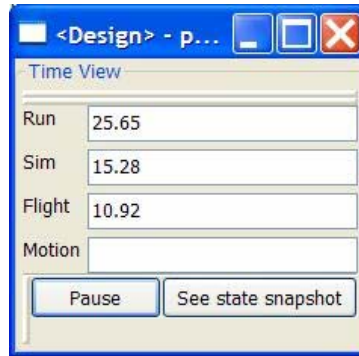
Scenario testing can start if the scenario scripts have been generated since the last scenario project save. The GUI should hide all the scenario creation panels and show the scenario testing panels. Two monitors will be recommended: one for ASES, one for AIMSsim.

The following activities are typical of testing a scenario being created:

1. Start the simulation system for the currently loaded scenario; starting it causes simulation processes to be started in right order, from correct folder; user can select to not start some of the default processes that would otherwise get started; user can enter extra parameters
2. Stop the simulation system, causing graceful exit; user can choose which processes to stop if desired;



3. View logs created: one tab per log, showing one column for time, one for log id, one for message; and updated in real time; each log view allows a filter on that log; the filter can be copied to other log views; new log tabs are generated if scripts create new logs;
4. View the time information for system: time since start of system (run -- ie clock -- time), cumulative simulation time (which advances only if at least one vehicle is moving), flight time; view the state snapshot which is the subset of Entities tree window of Scenario Creation GUI, showing current values of attributes.



5. View timed events queue: this shows when (simulation time) the events will be triggered, and what script will be run;
6. View tasks (periodic scripts);
7. View stages: shows history of scenario stages executed so far, the transitions that caused them to execute; time of transitions;
8. View World (read-only mode); when user selects something in view, its current state, as per simulation, is shown in a separate panel. This is useful since it may happen that the MMD in the simulation has been turned off or may not contain all information necessary for testing.

Scenario Running

For running a scenario, most of the testing panels are not required. In addition, the simDisplay should occupy the whole root window atop the ASes window so that the operator cannot tamper with the settings. If this is not practical, the ASes GUI could be “locked” after a run is started, requiring a password to be unlocked. In this way the logs would still be visible. The first three items from the Scenario Testing activities should be available.

Main frame

The GUI main frame is the “whole application” GUI, i.e. the main window of the application with the menus, toolbar(s), status bar and central (“client”) area where the panels change based on the task at hand.

The main frame should support docking and undocking of any panel created such that the user can decide where they want to position panels. This configuration should be saveable so that next time the application is run the default views will be the same.

The menu bar should have at least the following items, before a scenario is loaded or a new one started:

File

- New, Open, Save, Save As...
- Generate scripts
- Print
- Exit
- View
 - Show panel...
- Task
 - Create
 - Test
 - Run
- Tools
 - Options
- Help
 - Manual
 - About

As soon as a scenario has been loaded a new one created, the above menu will contain extra items based on the chosen task (create, test, run).

Bibliography

1. Gamble, M. (2001). *ELVISS Software Prototype - System Manual*. The HFE Group. Defence Research and Development Canada - Toronto Contract Report: DRDC Toronto CR 2001-029.
2. Gamble, Murray G. (November 10, 2000). Proposal to Modify ELVISS Prototype Software. Volume One - Technical and Management Proposal. The HFE Group.
3. Neal, B. (April 28, 1999). *ELVISS Human Engineering Design Approach Document – Operator*. Canadian Marconi Company, Human Factors Engineering Aerospace Division. Defence Civil Institute of Environmental Medicine Contract Report Number: DCIEM-CR-1999-078CMC-1000-1182.
4. Schoenborn, O. (2006a). *AIMSsim version 2.2.1 User Manual*. CAE Professional Services, Ottawa, Ontario. Defence Research and Development Canada - Atlantic: DRDC Atlantic CR 2006-280.
5. Schoenborn, O. (2006b). *AIMSsim version 2.2.1 System Manual*. CAE Professional Services, Ottawa, Ontario. Defence Research and Development Canada - Atlantic: DRDC Atlantic CR 2006-270.
6. Schoenborn, O. (2006c). *Herc SAR Task 106: AIMS Feature Development Final Report*. CAE Professional Services, Ottawa, Ontario. Defence Research and Development Canada - Atlantic: DRDC Atlantic CR 2006-278.

List of abbreviations and acronyms

AGTV	Active-Gated TV
AIMS	Advanced Integrated Multi-Sensor system
API	Application Programming Interface
ASES	AIMS Simulation-based Experimentation Station (ASES)
CAE PS	CAE Professional Services
DLL	Dynamically Loaded Library
DND	Department of National Defence
EO	Electro-Optical
FLIR	Forward-Looking IR
FOV	Field-of-View
FTP	File Transfer Protocol
GUI	Graphical User Interface
HMI	Human-Machine Interaction
IR	Infra-Red
IT	Information Technologies
MMD	Moving Map Display
OMI	Operator-Machine Interface
POV	Point-of-View
PDF	Portable Document Format
QA	Quality Assurance
SAR	Search-and-Rescue
SGE	Scenario Generation Environment
SOW	Statement of Work
THIR	Thermal Imager
UV	Ultra-Violet

This page intentionally left blank

Distribution list

Document No.: DRDC Atlantic CR 2007-302

LIST PART 1: Internal Distribution by Centre:

- 5 DRDC Atlantic Library
- 2 DRDC Atlantic Scientific Authority
- 7

TOTAL LIST PART 1

LIST PART 2: External Distribution by DRDKIM

- 1 Jocelyn Keillor
DRDC Toronto
PO Box 2000
Toronto, Ontario M3M 3B9
Canada
- 1 DRDKIM
- 2

TOTAL LIST PART 2

9 TOTAL COPIES REQUIRED

This page intentionally left blank

UNCLASSIFIED

DOCUMENT CONTROL DATA (Security classification of the title, body of abstract and indexing annotation must be entered when the overall document is classified)		
1. ORIGINATOR (The name and address of the organization preparing the document, Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's document, or tasking agency, are entered in section 8.) Publishing: DRDC Atlantic Performing: CAE Professional Services, 1135 Innovation Dr. Suite 300, Ottawa, ON , K2K 3G7 Monitoring: Contracting:		2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.) UNCLASSIFIED
3. TITLE (The complete document title as indicated on the title page. Its classification is indicated by the appropriate abbreviation (S, C, R, or U) in parenthesis at the end of the title) Human Factors Research Task 2006-111: AIMSSim Feature Development II (U)		
4. AUTHORS (First name, middle initial and last name. If military, show rank, e.g. Maj. John E. Doe.) O. Schoenbom		
5. DATE OF PUBLICATION (Month and year of publication of document.) January 2008	6a NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.) 38	6b. NO. OF REFS (Total cited in document.) 4
7. DESCRIPTIVE NOTES (The category of the document, e.g. technical document, technical note or memorandum. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.) Contract Report Associated documents: DRDC Atlantic CR 2007-300 – User Manual, DRDC Atlantic CR 2007-301 – System Manual.		
8. SPONSORING ACTIVITY (The names of the department project office or laboratory sponsoring the research and development – include address.) Sponsoring: Tasking: DRDC Atlantic DRDC Valcartier		
9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant under which the document was written. Please specify whether project or grant.) 13dx	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)	
10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document) DRDC Atlantic CR 2007-302	10b. OTHER DOCUMENT NO(S). (Any other numbers under which may be assigned this document either by the originator or by the sponsor.)	
11. DOCUMENT AVAILABILITY (Any limitations on the dissemination of the document, other than those imposed by security classification.) Unlimited distribution		
12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11), However, when further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.)) Unlimited announcement		

UNCLASSIFIED

UNCLASSIFIED

DOCUMENT CONTROL DATA

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

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

(U) This document is the final report for Human Factors Research Task 2006–111 – AIMS Feature Development which consisted of software development for a research platform designed to simulate the operator–machine interface and controls for an airborne multi–sensor surveillance system. The research platform is used to evaluate design concepts and operator performance issues and is modelled on the Advanced Integrated Multi–sensor Surveillance (AIMS) system (formerly known as the Enhanced Low–Light Level Visible and Infrared Surveillance System – ELVISS) which is being developed at DRDC Valcartier. With up to five sensors, the AIMS will enhance the capability of search and rescue (SAR) by allowing crews to operate effectively at night and in degraded weather conditions. The report summarizes the work performed in the project and makes recommendations for the next phase. Core tasks included adding functionality to over–ride the default sensor controller on the control box and replace it with a universal serial bus (USB) controller, and adding the capability to rearrange the sensor windows and other features on the interface display and other areas on the interface. Both capabilities were added for the purpose of evaluating tools and interface design concepts.

(U) Le présent document est le rapport final sur la tâche de recherche sur les facteurs humains 2006–111 – Développement des caractéristiques du système perfectionné de surveillance multi–capteurs intégré (AIMS) – qui consistait à élaborer le logiciel pour une plate–forme de recherche conçue pour simuler l'interface opérateur–machine et les commandes d'un système de surveillance multi–capteurs aéroporté. La plate–forme de recherche sert à évaluer les principes de conception et le rendement de l'opérateur et est dérivée du système AIMS (anciennement appelé Système perfectionné de surveillance à intensification de lumière visible et à infrarouge ou ELVISS) qui est en cours de développement à RDDC Valcartier. Équipé de deux à cinq capteurs, le système AIMS améliorera la capacité de recherche et sauvetage (SAR) en permettant aux équipages de travailler avec efficacité la nuit et par intempérie. Le rapport résume le travail effectué dans le cadre du projet et fait des recommandations pour la prochaine phase. Les tâches de base comprenaient l'ajout de fonctions pour surpasser le contrôleur des capteurs par défaut sur le boîtier de commande et le remplacer par un contrôleur à bus série universel (USB) et l'ajout de la capacité à réaménager les fenêtres de capteur et d'autres caractéristiques sur l'affichage d'interface et d'autres zones de l'interface. Les deux capacités ont été ajoutées dans le but d'évaluer des outils et les principes de conception de l'interface.

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

(U) simulation, airborne, interface, search and rescue, surveillance

UNCLASSIFIED

This page intentionally left blank.

Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale



www.drdc-rddc.gc.ca