A Cooperative Navigation Simulation Framework for Designing Robust Navigation Systems

Yetkin Ersoy[1], Gokhan Soysal[1], Murat Eren[2], Yakup Ozkazanc[3],
Asim Egemen Yilmaz[1] and Murat Efe[1]

[1]Center for Smart Systems and Technologies
Ankara University
TURKEY

[2]ASELSAN Inc.
TURKEY

[3]Dept. of Electrical and Electronics Eng.
Hacettepe University
TURKEY

yersoy@ankara.edu.tr
soysal@eng.ankara.edu.tr
meren@aselsan.com.tr
yakup@ee.hacettepe.edu.tr
aeyilmaz@eng.ankara.edu.tr
efe@eng.ankara.edu.tr

ABSTRACT

In this study a flexible and scalable simulation framework is presented for cooperative navigation in GNSS degraded environments. The framework is being developed as a part of project that aims to provide assured navigation output in two different integration architectures. It is capable of creating realistic scenarios and use realistic sensor/system models from sensor/system developers as a black box and analyze the contribution of the individual sensor/system under the created scenario.

1.0 INTRODUCTION

Navigation is the ability to determine own position, orientation and velocity relative to a known reference point and one's desire to travel to point A from point B has been the driving force behind developing systems/tools to navigate accurately and safely. Deployment of Global Positioning System (GPS) satellites by the US in 1960s has provided (almost) ubiquitous opportunity to determine the position of a GPS receiver with a relatively good accuracy. This unique feature is the main reason why GPS (or as a more broad term GNSS – Global Navigation Satellite System) is integrated with the Inertial Navigation System (INS) to reduce errors. INS uses sensors and mathematical models in order to produce position information with respect to a starting point and is always at the heart of a navigation system. However, due to mathematical model mismatch and sensor limitations, INS errors build up in time requiring a positional fix is for better accuracy and the GPS is usually the most reliable source to provide that positional fix information.

While INS/GPS integration is mostly used to accommodate military needs in terms of navigation, with the current technology and ever so cheap GPS receivers, GPS is increasingly being employed in many non-military equipment, such as mobile phones, to help with the needs of the general public. Recently, the GPS technology has been used in a large number of civilian applications as well ranging from intelligent transportation systems to precision farming. The functionality provided by the GPS/GNSS has gone beyond positioning and now has a more general term, namely, Position, Navigation and Timing (PNT)1. While GNSS is, without a doubt, at the heart of almost all military and safety and mission-critical PNT applications, it has vulnerabilities that can impair its performance significantly. The vulnerabilities stem

---

[1] Positioning is the ability to accurately and precisely determine one's location and orientation anywhere in 2D or 3D. Navigation is the ability to determine current and desired position and apply corrections to course, orientation, and speed to attain the desired position anywhere. Timing is the ability to acquire accurate and precise time from a standard anywhere. Timing includes time transfer [8].

from the efforts to make it widely available, namely using satellite constellations, and can be listed as follows;

- GNSS signals have a very well-known frequency/modulation and structure and travel a long way before they reach the receiver rendering the signal become rather weak. This makes GNSS vulnerable to jamming and spoofing. In fact while spoofing a GPS signal requires a bit more elaborate effort, a GPS jammer can easily be purchased cheaply online[2].

- Since GNSS works on the time of arrival of signals from each satellite in the constellation, line of sight (LOS) becomes an issue. High mountains and especially high rise buildings in urban areas prevent LOS and reduce GNSS availability especially when satellites are at low elevation angles.

Such vulnerabilities cause a phenomenon called “GNSS degradation” which is not to be taken lightly as GNSS degradation seriously hinders PNT capabilities. This is probably why “PNT in GPS-denied environments” was identified as one of the top 12 (in terms of priority) research areas that should be emphasized in the near future [1]. In fact, the research on improving PNT capabilities goes a long way before [1] for indoor applications where GNSS signals are naturally not available [2], [3]. The idea is very simple, identify other ways of determining position and use them either individually or collectively to elevate PNT capability. This is called “Collaborative PNT” or as the current technology dictates “All Source PNT” [4], [5] where the latter provides a comprehensive list of what it calls “Complementary PNT Technologies”. The list given in [5] also contains information on the expected performance of each complementary technology along with their potential advantages/disadvantages as well as a brief discussion of each technology. The list of complementary PNT technologies as given in [5] is repeated here without discussion.

- Multi-constellation (or multi-frequency) GNSS
- Signals of Opportunity
- Celestial Navigation
- Vision-based Navigation
- Terrain Referenced Navigation (TRN)
- Magnetic Anomaly Referenced Navigation (MARN)
- Gravity Gradient Anomaly Referenced Navigation (GGARN)
- Gravity Anomaly Referenced Navigation (GARN)
- Network Collaborative PNT

Utilizing whatever source of PNT available in order to yield a robust/assured/resilient PNT [6], [7] capability in a GNSS degraded environment requires a System of Systems Approach (SoSA) [8]. An assured PNT system must provide quantifiable accuracy, integrity as well as continuity and availability. Accuracy is defined by the error which is calculated as the difference between the estimated and actual positions. Integrity, on the other hand, is a measure of trust that can be placed in the correctness of the information supplied by the overall PNT and must always be monitored.

Having to employ numerous heterogeneous systems in a SoS structure and constantly monitor system’s accuracy and integrity is not trivial. In fact it involves a certain level of more difficulty than associated with developing an individual system. With so many components and their interdependencies, the Open Architecture (OA) approach [9] could provide flexibility and relative ease in developing an assured/resilient PNT system as it comes with predetermined standards and modularity. The OA also allows for multiple developers of sub-systems to work on an assured PNT system without having to know the specifics of other sub-systems but contributing to the performance of the overall system. In [10] the benefits of the OA approach was explained with specific reference to a GPS receiver but it could easily be generalized and applied to any system.

---

[2] Buying and using GPS jammers are illegal in most countries, therefore no reference to online purchasing is provided here.

While all source PNT system ensures the continuity and availability of the PNT approach, it requires detailed planning and critical design decisions. This could prove difficult even when the OA approach is followed and a tool to provide insight into the added contribution of each added sub-system to the overall system would be beneficial. In this study we discuss such a simulation tool/framework that would allow a system designer to construct an integrated navigation system, i.e., an assured PNT system, and test its performance in challenging environments. The simulation tool aims to provide a means for effortlessly define, include or create navigation signals/systems to integrate and test in simulated but realistic environment created with a choice of parameters through the tool's user interface. With its versatile design, the simulation tool supports both signal level integration and system level integration. The signal level integration, also called INS centric integration, allows to observe whether the performance of an INS can be improved with inputs from other systems whereas the system level integration aims to use data fusion techniques to provide robust and more accurate navigation output. The tool is designed in a modular fashion where each signal or system component can be included as a plug-in. This architecture enables the user to utilize either a modeled signal/system or a pre-recorded output from a real-life system. Such and architecture also opens the door for hardware-in-the-loop (HWIL) operation where almost real-life performance of the integrated system can be observed. Since the plug-ins work based on input-output relationships any novel system can be tested as a black box without any knowledge of the system specifics. Last but not least, the simulation architecture is in total synch with the OA approach and can easily support testing of such efforts as well.

2.0 ASSURED PNT SIMULATION FRAMEWORK – ASSURED-SIM

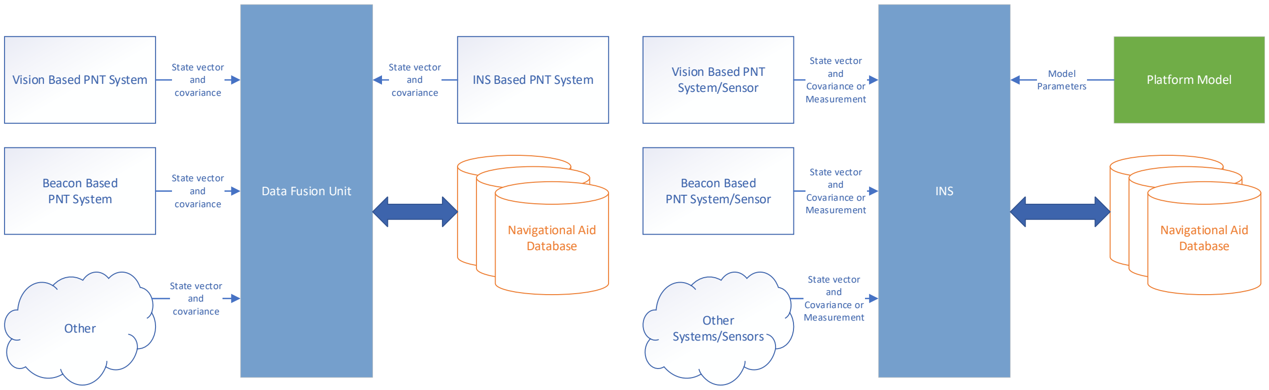
In this section the simulation framework that has been developed to test, verify and perform performance analysis of Assured-PNT Systems in GNSS denied environments will be detailed. The proposed framework has been designed to support both signal and system level architectures with its modular architecture. Although the aim is to design system of Navigation Systems, the framework also supports the system design with GNSS in order to allow user to observe the performance increase should the GNSS be available. On the other hand, the simulation framework will enable user to employ whatever complementary PNT technologies are available through easy to use graphical user interface (GUI) in order to observe whether the available sub-systems would provide reliable PNT output through the selected integration architecture when GNSS is not available or trusted.

Since each individual complementary PNT system and navigation sensor has their own intricate structure and unique features, the proposed framework models those systems/sensors based on their input-output relationships and available error models. This approach keeps the framework simple and let the user focus on the integration performance rather than design each PNT system through the framework. In addition, this approach, allows the developers of individual systems to withhold the details of their PNT systems and only share the unrevealing input-output relationships and error models that allows the assured PNT system designer to mix and match sub systems/sensors from different developers/nations without any design details being let on. Moreover, such a flexible and modular approach supports the OA and ensures scalability.

Two different architectures can be studied using the proposed simulation framework, namely,

- Distributed Integration and
- INS centric Integration

where, the distributed architecture will allow to test and analyze the performance when the navigation outputs of disparate PNT systems are fused using data fusion approaches. On the other hand with the INS centric architecture the aim is to see whether separate complementary PNT systems could be used to provide positional fix information to the INS available either individually or in a fused manner. Both architectures are illustrated in Fig. 2.1.



(a) Distributed Integration Architecture (b) INS Centric Integration Architecture

Figure 2-1: Integration Architectures.

2.1. Design Decisions

Important design decisions associated with Assured-Sim are as follows;

- Object oriented methodology is used in analysis, design and verification of the software,
- Layered software architecture is used to maximize portability,
- Assured-Sim is developed in a modular fashion in order to support open software architecture and maximize flexibility,
- General purpose C++ programing language and ISO/IEC 14882:2017 are selected for Assured-Sim,
- Currently Assured-Sim runs on Windows 10 operating system.

2.2 Assured-Sim Software Architecture

Assured-Sim open software architecture accommodates the needs for separate sub-system developers. The framework offers a systematic way to maintain files, create ground truth based on scenario files, create outputs of based on individual sensor/PNT system model using the ground truth, integrate the outputs of individual sensor/PNT system and record/analyze the results. High-level software architecture for Assured-Sim id given in Figure 2.2.

The proposed simulation framework provides a user-friendly interface to create repeatable scenarios, adjust parameters, record and re-play. Environment information associated to each complementary PNT technology is available to support each available PNT system. Many different formats are already supported and additional file formats can be added if needed.

Ground truth data is the most important data in order to analyze the outputs. Ground truth is generated based on the user entered fundamental scenario data and recorded to be used in performance analysis.

As the outputs of individual PNT system and/or sensor will be generated using the ground truth along with the input-output relationship of individual PNT system and/or sensor to be analyzed, these input-output relationships and the error models are needed. Generation of system/sensor outputs will be performed through individual plugin of the system of interest that will either be developed by the technology developer following the guidelines or by Assured-Sim developers based on the information provided. Such an approach will allow Assured-Sim to be employed to analyze field data as well. This process paves the way for performance analysis of individual complementary PNT technology on the created scenario. Assured-Sim produces navigation results for individual sub-system as well as for the integration architecture for comparison. Analysis of the comparative results will reveal how much would be gained by inclusion of an individual PNT technology based on the scenario. This information could be used to select which available sub-systems to employed to improve the navigation output. Following subsections aims to increase familiarity with Assured-Sim by explaining the modes of operation through use cases.

2.2.1 Scenario Management and Ground Truth Generation

Ground truth generation starts with the selection of scenario file using the GUI. Then the scenario file is read and the necessary information is passed on to the ground truth generator. Generated ground truth is then transferred to available sensors and complementary PNT systems in order to generate measurements or navigation outputs. Ground truth also gets transferred to the record manager. Figure 2.2 depicts the process.

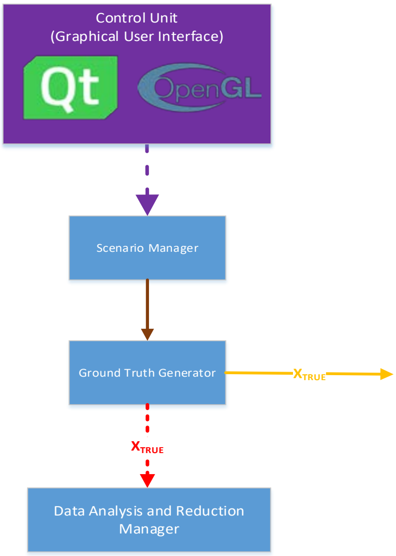


Figure 2-2: Scenario management and ground truth generation.

The module for scenario management and ground truth generation is responsible for the opening of saved scenarios as well as saving any changes made to them. When attempting to open a scenario file the “File Manager” lists all available files, then the file selected by the user is parsed by the “Scenario Parser”. A warning is generated in case of an error during parsing and necessary actions are expected to be taken by the user in accordance with the warning. Parsed scenario information is translated into the reference data by the “Ground Truth Generator” and the ground truth data is saved by the “Record Manager”. Current scenario can be saved through the GUI. Necessary information for a scenario to be completely represented is provided by the “Scenario Manager” and this information is saved into the file by the “File Manager” but the ground truth data will not be saved onto the scenario file. Ground truth data could be saved by the user along with the simulation environment parameters onto a separate file. The use case diagram for scenario management and ground truth generation is shown in Figure 2.4.

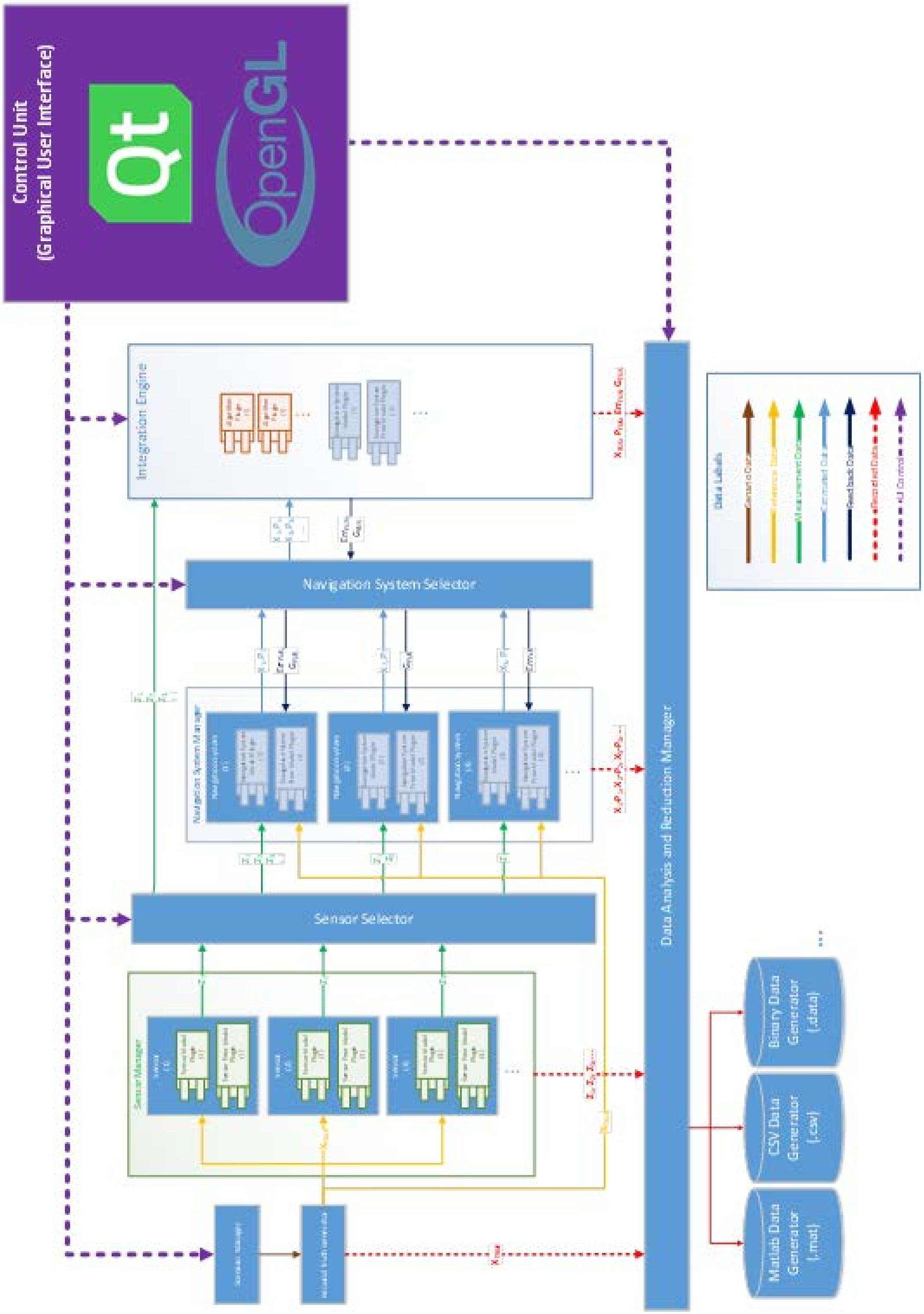


Figure 2-3: Assured-Sim High-Level Software Architecture

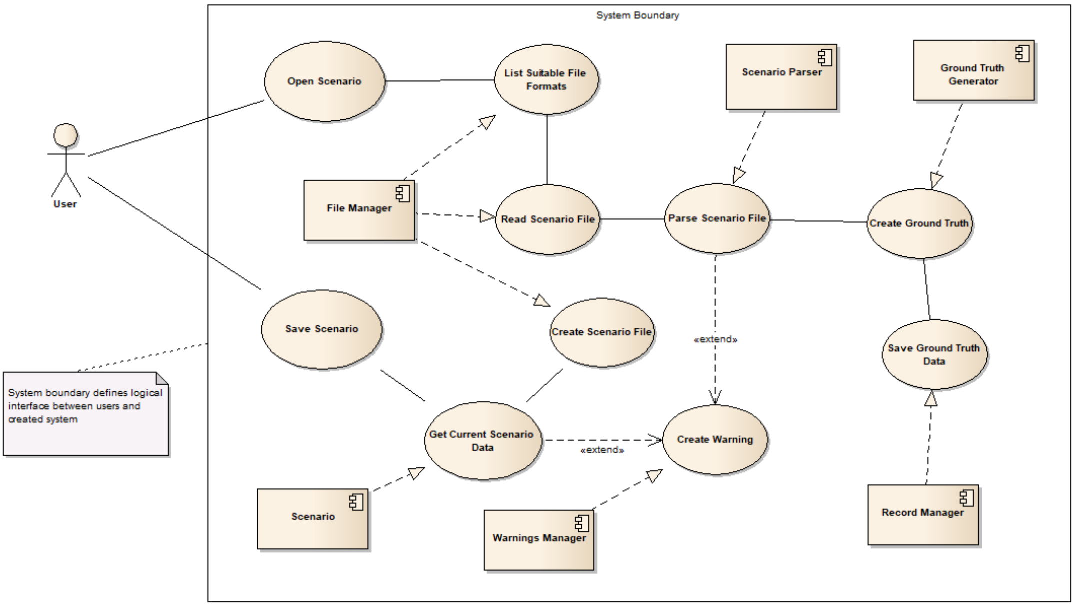


Figure 2-4: Use case diagram for scenario management and ground truth generation.

2.2.2. Generation of Sensor and Navigation System Outputs

Generation of Sensor and Navigation System Outputs has a similar flow. Each sensor and/or navigation system automatically translates the ground truth into sensor/system output using the system and error models associated with it. However, only the user selected sensor/system outputs are transferred to the following module, whereas generated but unselected outputs are saved by the "Record Manager". This way even unselected outputs could be used for comparison during performance analysis. Figure 2.5 illustrates the general flow for the generation of sensor/system outputs.

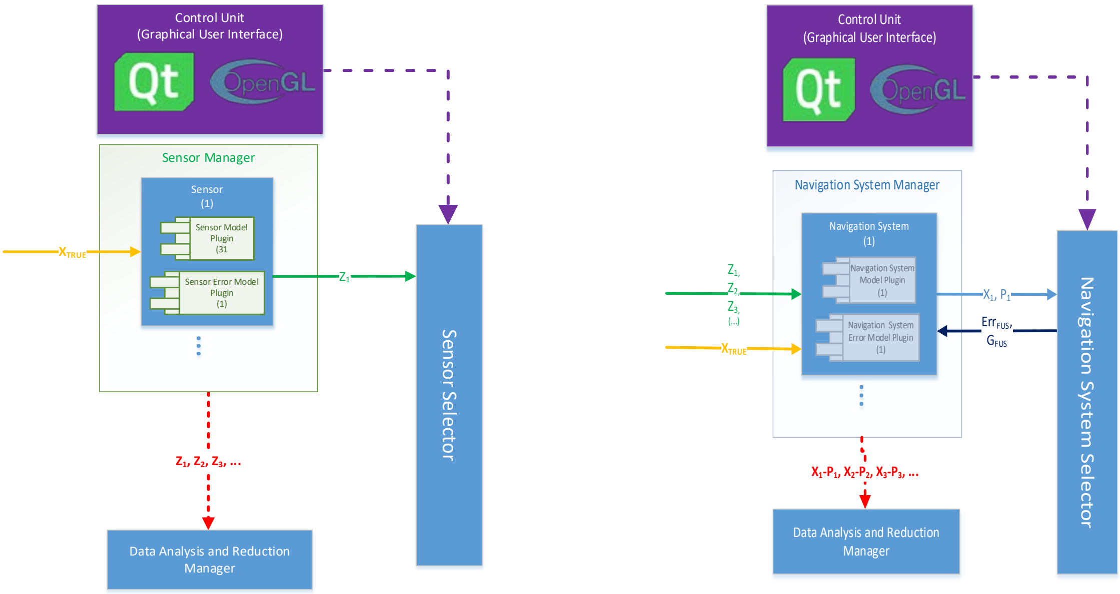


Figure 2-5: Generation of sensor/system outputs.

In order to ensure design flexibility individual sensor and sensor error models as well as individual complementary PNT system and system error models are included as plugins. Following the design decision to run Assured-Sim on Windows operating system plugins will be developed as dynamic link library (.dll). Guidelines and interface information are provided for the technology developers for problem-free integration of plugins.

Although C++ is the selected programing language for Assured-Sim development, C programing is used for plugin interface in order to increase portability. The designed interface is optimal for providing the necessary functionality for feeding data in and retrieving data from the plugins. In anticipation that each individual sensor/system need and the data they provide would be different only state vectors and associated covariance will be transferred through the interface. Any unforeseen input/output data needs will be accommodated through specifically developed data transfer infrastructure when needed. A generic plugin is shown in Figure 2-6.

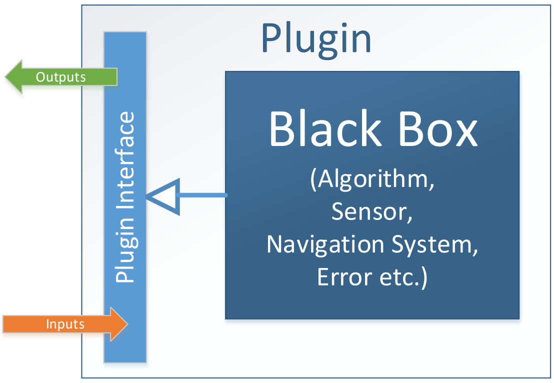


Figure 2-6: Generic plugin structure.

In order to ensure confidentiality, the functionality provided by any plugin will be treated as a black-box. In other words, Assured-Sim will only use the outputs generated by the plugins and will be oblivious to how that output is generated. As mentioned before the technology developers will be provided with necessary guidelines to develop a dynamic library e.g., .dll, for their proprietary sensor/system so that input/output communication could be established with it through an interface.

During operation all user selections will be saved into the scenario file. Management of selected sensor and/or system will be handled by “Scenario Manager”. When the user plays the scenario, sensors and systems defined in the scenario as well as sensor/system models and their error models will be loaded by “Sensor Manager” and “Navigation System Manager”. When the scenario run completes all plugins will be removed from the simulation. Loading and removing of plugins are handled by “Plugin Manager”. The ground truth is fed into successfully loaded sensor and navigation systems to generate sensor/system outputs which are then transferred into the integration engine in the following stage. Time synchronization throughout the scenario running process is provided by “Scenario Manager”. Scenario manager is also responsible for tracking any errors during run time and ensuring that simulation is completed. The use case diagram for sensor/system output generation is given in Figure 2-7.

2.2.3. Sensor/System Integration

Generation of sensor/system outputs triggers sensor/system integration engine. Integration engine acts on the sensor/system outputs, selected algorithm information and system and error models. Record manager saves all results produced by the integration engine. The process is shown in Figure 2-8. Integration engine supports both integration architectures. Integration for distributed architecture uses algorithm plugins whereas INS centric architecture uses the INS and INS error models plugins.

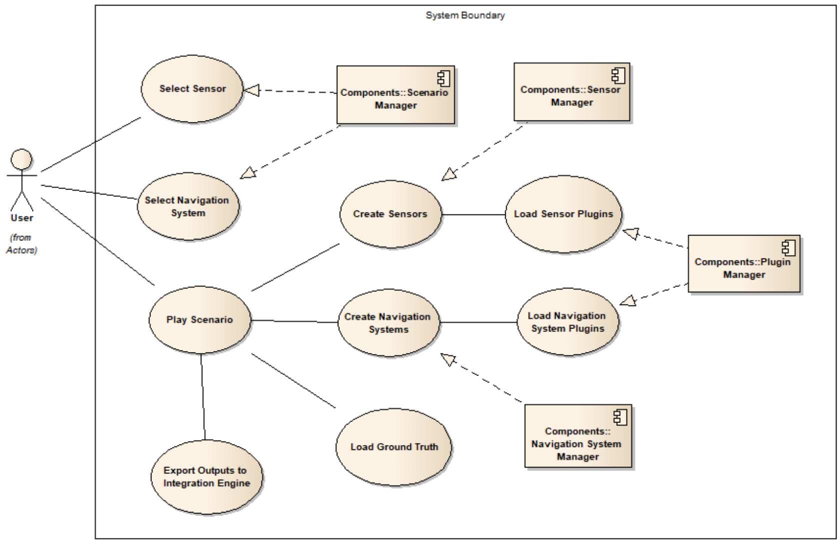


Figure 2-7: Use case diagram for the generation of sensor/system outputs.

When a scenario runs in Assured-Sim, the required algorithm and/or model plugins are loaded into the simulation environment by “Plugin Manager” and configuration parameters are adjusted. Then the generated sensor and/or system outputs are fed into the configured integration engine. The integration output is saved by the record manager for analysis. Time synchronization of integration engine with other scenario components is handled by the scenario manager. If defined in the scenario file, Assured-Sim is also devised to feed the integrated output (or any desired portion of it) back to any selected individual complementary PNT system (in case they have mechanism to correct their output). Figure 2-9 depicts the use case diagram for the integration process.

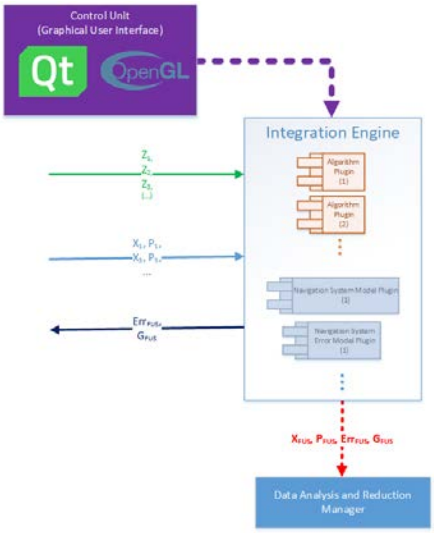


Figure 2-8: Integration process.

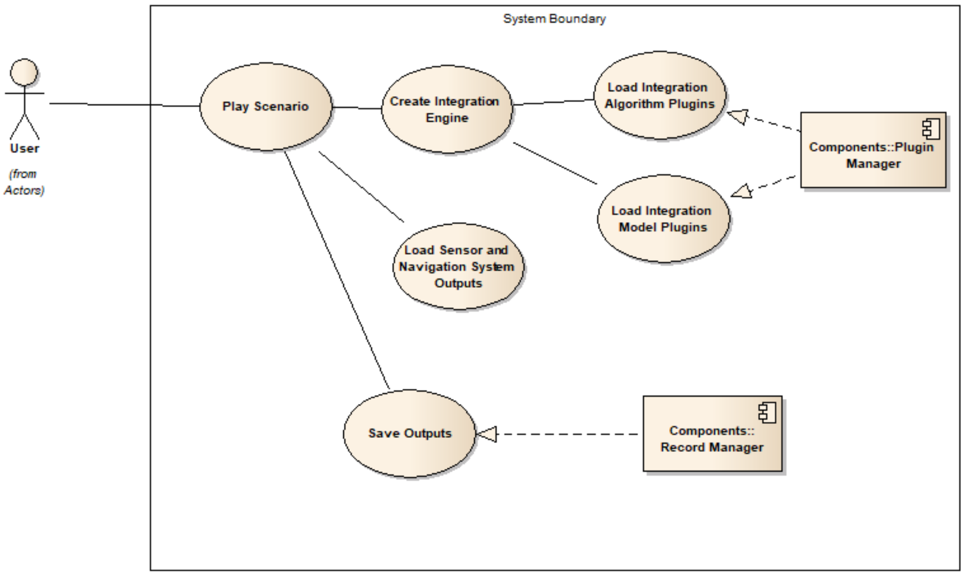


Figure 2-9: Use case diagram for integration process.

2.2.4. Data Recording and Analysis

Data recording and analysis is responsible for recording of all data and exporting it on demand. Data is exported in selected format. The process is handled by "Data Analysis and Reduction Manager" and is shown in Figure 2-10. Assured-Sim allows users to analyze data and/or results and visualize the analyzed data through GUI. Analyzed and/or reduced data is used only for visualization, if needed, the data must be exported for later use. The use case diagram for data analysis and recording is given in Figure 2-11.

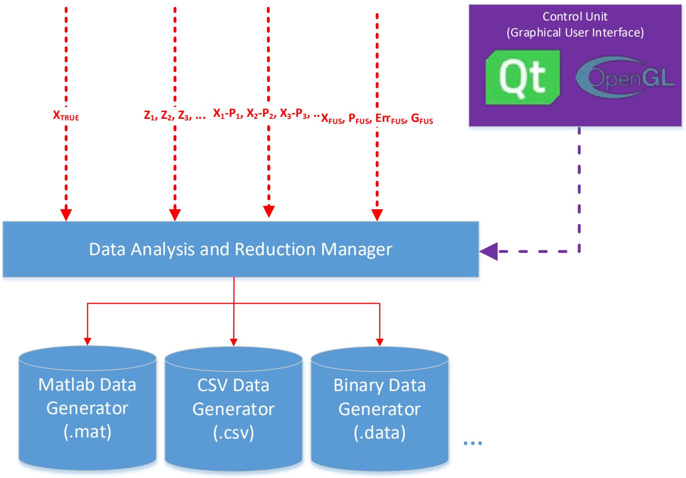


Figure 2-10: Data analysis and recording.

3. PLANS FOR THE FUTURE

In this section possible future extension to the Assured-Sim is discussed. All source PNT aims to offer PNT solution with accuracy and integrity, requiring a number of disparate complementary technology working together to contribute to a common purpose. This structure coincides with the distributed decision making (DDM) process.

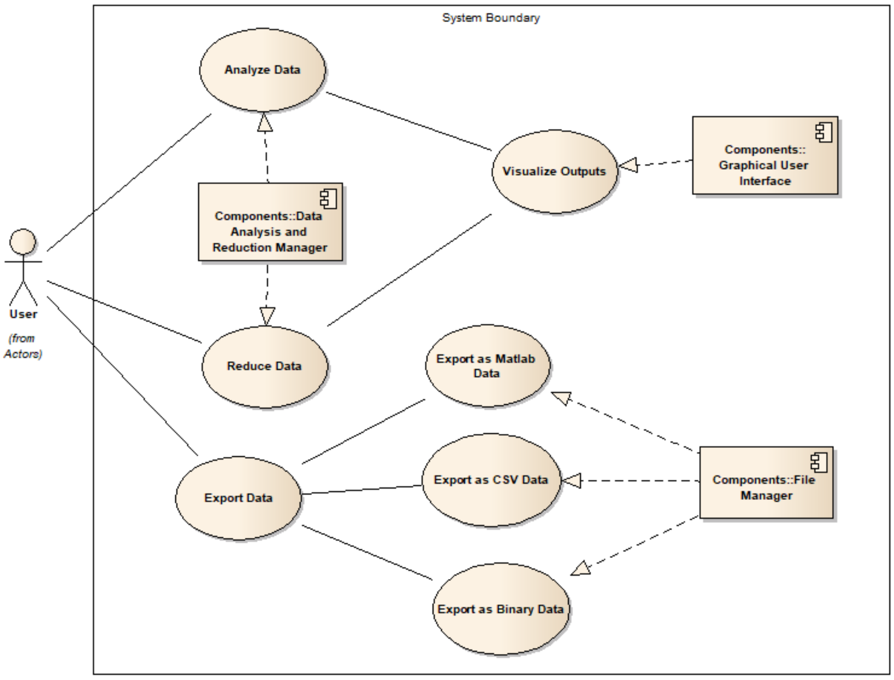


Figure 2-11: Use case diagram for data analysis and recording.

DDM is nothing but a decision-making process where several agents (platforms) are involved to reach a single decision (PNT solution); for example, a problem solving activity among a few agents when the problem is too complex for any agent to solve it alone; or any single agent has only limited amount of information, which is insufficient to provide a solution of the whole problem [10]. More challenging issues arise when the communication/coordination among the problem-solving agents gets impaired due to various technical reasons. DDM process requires a “language” among the agents and a “protocol” (sequential and conditional set of rules) in order to achieve a consensus and a controlled guess about the quality of the obtained solution. Since 1990s, various studies have been conducted to define such languages (which are usually referred to as “Agent Communication(s) Languages - ACLs”) [11] and relevant protocols. As an ongoing work, the current Assured-Sim framework is being extended to support a custom ACL, namely, “DiDeML - Distributed Decision Making Language”; and an additional DDS-based service to achieve the inter-agent communications. When the implementation is complete, it will be possible to solve a navigation/localization by means of the DiDeML enhanced Assured-Sim. With the addition of DiDEML, Assured-Sin will be able to

- model the imperfections (such as communication delays, packet losses, etc.) of the Information Sharing Network seen in the relevant figure;
- find out suboptimal solutions via imperfect Information Sharing Networks; and
- analyze the resilience of the proposed protocols in challenging PNT scenarios.

4 CONCLUSIONS

A simulation framework called Assured-Sim for all source PNT applications is presented. The framework is flexible, scalable and supports open architecture. It allows performance analysis of different complementary PNT technology along with different navigation sensors in two different integration architectures. Moreover, flexible and scalable structure of the framework design also allows other integration architectures to be constructed, studied and analyzed. The proposed framework provides means for designing resilient PNT systems in GNSS degraded environments and is expected to help system designer to decide which components to include in the overall system. With little effort and modification Assured-Sim could be used in Hardware-in-the-loop (HWIL) configuration.

REFERENCES

[1] United States Air Force Chief Scientist (AF/ST), Report on Technology Horizons: A Vision for Air Force Science and Technology during 2010–2030, vol. 1, AF/ST-TR-10-01-PR (Washington, DC: Headquarters US Air Force, Office of the USAF Chief Scientist, 15 May 2010).

[2] M. Achtelik, A. Bachrach, R. He, S. Prentice and N. Roy, “Stereo Vision and Laser Odometry for Autonomous Helicopters in GPS-Denied Indoor Environments”, Unmanned Systems Technology XI, Proc. of SPIE Vol. 7332, 733219, 2009

[3] R. He, S. Prentice and N. Roy, “Planning in Information Space for a Quadrotor Helicopter in a GPS-denied Environment”, IEEE International Conference on Robotics and Automation Pasadena, CA, USA, May 19-23, 2008.

[4] K. A. Fisher and J. F. Raquet, “Precision Position, Navigation, and Timing without the Global Positioning System”, Air & Space Power Journal, Vol. 25, No. 2, p. 24, 2011.

[5] R. Li, “All Source Positioning, Navigation, and Timing”, Artech House, 2020.

[6] N. Ward and M. Bransby, "Requirements for Resilient PNT," Proceedings of the 2015 International Technical Meeting of The Institute of Navigation, Dana Point, California, pp. 234-238, 2015.

[7] E. Engler, M. Hoppe, R. Ziebold, Z. Dai, and T. Noack, “Resilient PNT: Vision and Mission”, Proceedings of. E-Navigation Underway Conference, 2012.

[8] US Army Positioning, Navigation, and Timing (PNT) Reference Architecture, v1.0, 1 July 2018.

[9] Y. Koren, S.J. Hu, P. Gu and M. Shpitalni, “Open-architecture products”, CIRP Annals, Vol. 62, No. 2, pp. 719-729, 2013.

[10] C. Schneeweiss, “Distributed decision making—a unified approach”, European Journal of Operational Research, Vol. 150, No. 2, pp. 237-252, 2003.

[11] G. K. Soon, C. K. On, P. Anthony and A. R. Hamdan, “A review on agent communication language”, In Computational Science and Technology (pp. 481-491), Springer, Singapore, 2019.