A GUI BASED SYSTEM FOR AUTOMATIC CONSTRUCTION OF ENGINEERING MODEL SOFTWARE FOR COMMAND RESPONSE AND TELEMETRY GENERATION

Joe Parlanti and Ronnie Pinkerton
CAE-Link Corporation
2224 Bay Area Boulevard
Houston, Texas 77058-2099

ABSTRACT

There exists today, numerous off-the-shelf hardware solutions for the generation of simulated telemetry data streams. The ability to rapidly develop engineering models to drive the data contents of the telemetry is restricted by the lack of contemporary CASE tools. This paper presents an object-oriented Graphical User Interface (GUI) approach to generation of mathematical models in order to reduce the time required for model generation to a fraction of today's development time, eliminate the need to write substantial amounts of software, and allow reuse of model objects in a manner consistent with the GUI cut, paste, and copy metaphors.

Key Words: Telemetry Generation, Engineering Models, Object Oriented.

INTRODUCTION

Traditional methods of building engineering software models in support of the simulation of telemetry data revolved around classical functional decomposition methodologies and simple text-based software development tools. Functionality of a defined process was examined and models were coded, compiled, linked, and tested, using very cumbersome tools. This process could take as long as 1 to 6 hours per iteration for very complex systems. One of the key disadvantages to this method is that for each system the functionality is very unique thus demanding unique models and code to produce the desired results. Recent technology innovations in software engineering have seen the emergence of the Object Oriented methodologies and the Graphical User Interface (GUI). This technology promises to provide the most significant advances in the current computer and hours per iteration for very complex systems. One of the key disadvantages to this method is that for each system the functionality is very

unique thus demanding unique models and code to produce the desired results. Recent technology innovations in software engineering have seen the emergence of the Object Oriented methodologies and the Graphical User Interface (GUI). This technology promises to provide the most significant advances in the current computer and software industry. It not only changes the thought processes involved with the problem solution it also redefines how the users relate to computers and the potential that can be realized by utilizing this technology.

OBJECT ORIENTED APPROACH

Todays methods of performing high fidelity telemetry simulation usually consist of large scale embedded systems comprising thousands of lines of third generation language code (such as FORTRAN) which is extremely expensive to produce and costly to maintain. The telemetry interface to this system is usually accomplished on the hardware level which performs well but is expensive in its self and limited in its flexibility. The emergence of object oriented techniques in the 80's brought about several key methodologies and concepts which apply to real world modeling which is the essence of telemetry system models. By definition an object, in the object oriented paradigm, models the real world. It is a self contained collection of functions and data bound together in a single unit, designed with a strict interface to the “outside world”. Object oriented design and programming, OOD and OOP, respectively, focus on organizing programs or systems as collections of objects. The main idea of the object oriented approach is to focus development efforts on the objects of a system as opposed to the traditional approach of being concerned primarily with functionality which is prevalent in many of todays existing systems. In an Object Oriented approach, data and functions are not separated but are closely coupled allowing the functions to work on the data and the data to determine which functions will execute. With this methodology, software objects can be designed in accordance with their real world counterparts and only a well defined interface to the external world will be visible thus creating a black box or encapsulated object.

ENGINEERING MODEL BUILDING

These black boxes can then be used as building blocks. These building blocks can then be constructed and integrated to form a functional model of the real world system. This will require a rule or knowledge based system capable of defining and maintaining the properties of each object and then implementing or creating an instance of the object as required.

The object decomposition of a simple vehicle is presented in Figure 1.0. This system or vehicle consists of a collection of real world objects or components, each with a very specific function and behavior. By defining these characteristics to the knowledge base and then applying them to corresponding software objects, they can be used and integrated as a functioning model. In order to fully utilize the capabilities of this approach, a repository of objects or black boxes would need to be available and maintained for most real world components. Added flexibility could be gained by providing varying levels of fidelity and generic behavior in the models as to maximize fidelity/performance ratios.

The output of this system would be fourth generation language software objects that could then be used as is or integrated into larger systems or a real-time simulation environments for training or verification testing purposes.

GRAPHICAL USER INTERFACE

The next fundamental component is relaying or interacting to the user the information needed to construct this type of object oriented knowledge based system. The concepts or the Graphical User Interface (GUI) are the best suited for this capability based on several studies. One such study was conducted in 1989 when Microsoft and Zenith Data Systems commissioned Temple, Barker & Sloane to study the performance attributes of GUI's versus character based user interfaces. The findings were as follows:

> Experienced GUI users finished 58% more correct work than users of character-based interfaces.
>
> Frustration and fatigue levels were lower among GUI users.
>
> GUI users were better at self-teaching and exploring, not to mention being better at learning more capabilities of machines and systems.

The application of GUI's to the development of telemetry simulation system models points toward the possibility of similar results. GUI's also provide a natural interface to object-oriented applications such as model building systems.

Figures 2.0 and 3.0 represent a model for a GUI to a conceptional engineering model building system. The key elements consist of the object oriented approach to the model building capabilities and the graphical representation of real world objects to the user.

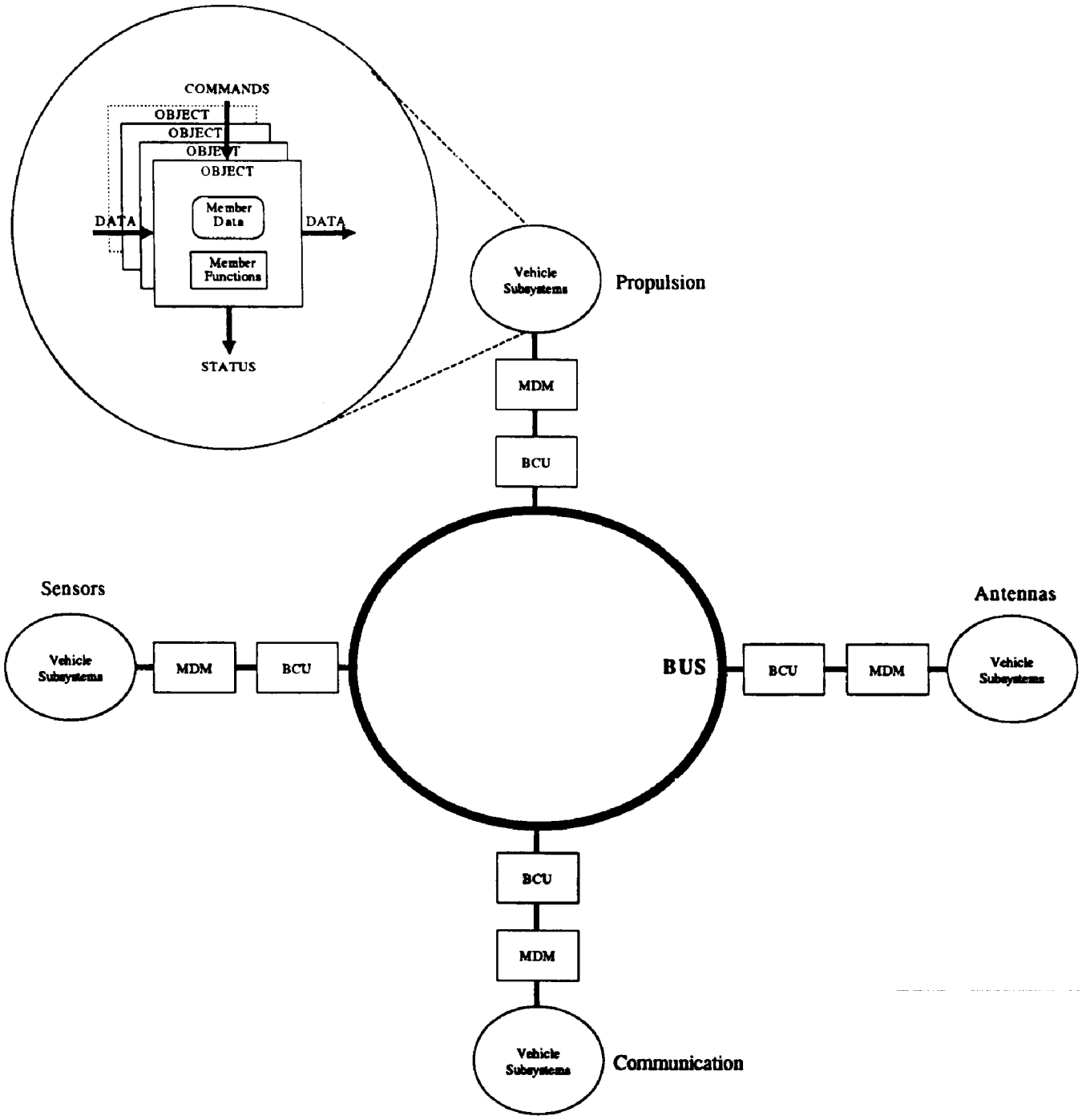

Figure 1.0 Simplified Vehicle Model

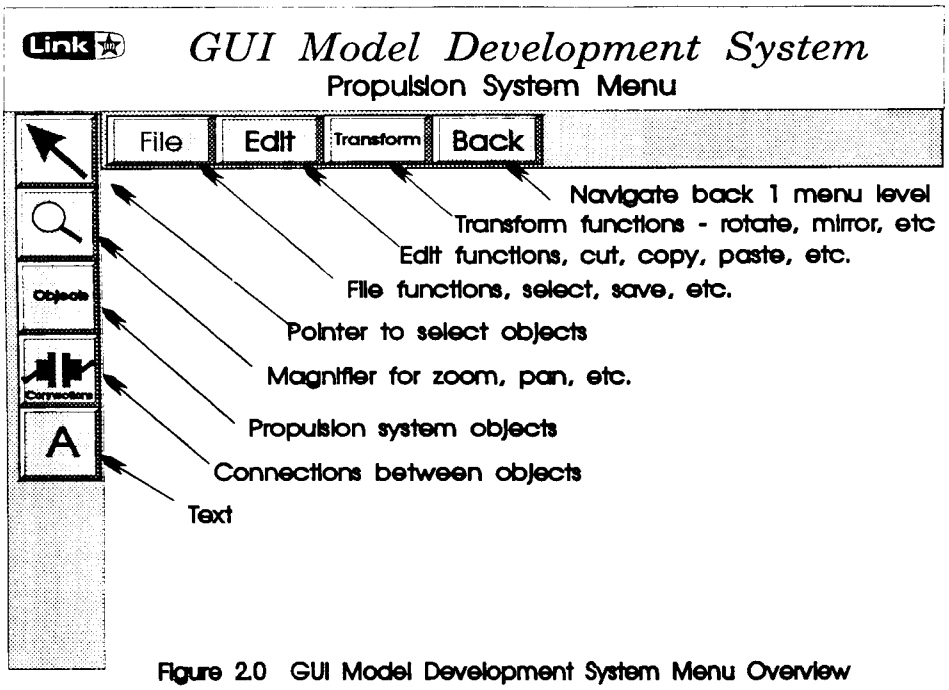

Figure 2.0 GUI Model Development System Menu Overview

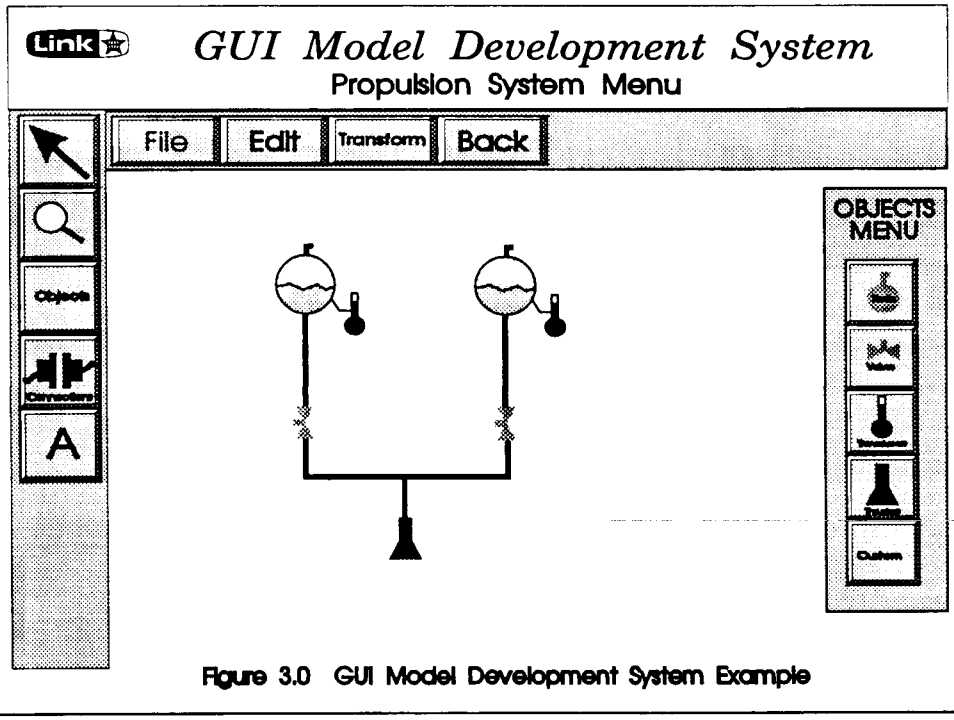

Figure 3.0 GUI Model Development System Example

CONCLUSION

By utilizing a GUI based modeling and telemetry generation system, rapid prototyping efforts would be much more feasible and provide faster more accurate data to aid in further design processes. The cost of such a system would be considerably less that contemporary systems and provide much more flexibility. With this model building capability, entire systems or vehicles could be modeled and then plugged into generic type environment simulation systems to provide an entirely integrated vehicle telemetry simulation suitable for engineering development as well as turn key training devices. This could also lead to a greater increase in efficiency in actual flight article systems by providing a what-if capability to system designers.

Also, this technology does not strictly apply to telemetry systems. There are others areas within aerospace which are studying the potential of this technology or are currently implementing similar systems. With the concepts of OOD, these existing systems could possibly be tailored or reused in this particular application.

ACKNOWLEDGEMENTS

Coad, Peter and Yourdon, Edwards, Object-Oriented Analysis, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1990.

Object Oriented Structured Design, Interactive Development Environments, Inc., San Francisco, California, 1990.

Martin Heller, Ph.D., Graphical User Interfaces, 1990.