

# IMPROVED VALIDATION & TESTING OF REMOTE ROBOTICS SYSTEMS BY INTEGRATING SIMULATION & GRAPHICAL USER INTERFACE CUSTOMISATION

Ian Matthews

National Instruments France, 2 Rue Hennape, 92735 Nanterre, France, Email: ian.matthews@ni.com

## 1. ABSTRACT

The paper looks at methodologies for validating remote robotics system design through the use of graphical system design tools. By developing integration between the design, prototyping and validation phases of the system design, robotics systems can make use of simulation models for both the initial development and integrate them into field-testing preparation. By integrating the simulation models at all stages, systems can be moved efficiently from simulated environments to physical testing. Technologies allow the same tool chain can be used to across the design and deployment of the robotics system without having to changed tools with development phases. The graphical system design methodology means that linking a variety of simulation systems with the robotics system can be implemented quickly and common graphical human interfaces can be linked into both simulation and physical systems. With standardized interfaces between stages, human interactions can be trialled without changing between simulated and physical stages. These methodologies have been successfully deployed by the Marine Robotics Research Centre at the University of Limerick [1] to enable rapid prototyping of control algorithms and then test them at the validation stages before field-testing with prototype hardware.

## 2. Introduction

The traditional design approach of working from a design through simulation to prototype and finally to deployment of a system is common across many domains including robotics. Often the tools used in design and simulations are separated from prototyping hardware and information is transferred in the form of specifications and requirements between phases. This leads to the scenario shown in Fig.1 where the developer works with simulated environments before prototype systems are tested in trial environments leading to final developed systems being used by operators in the final deployment. This often means that interfaces and tools developed to help with prototyping cannot take advantage of early stage simulation tools and that the final operator often will not have been able to work effectively with the simulation and prototype phases since the interfaces and tools will have been optimised for development of first the simulation then the prototype.

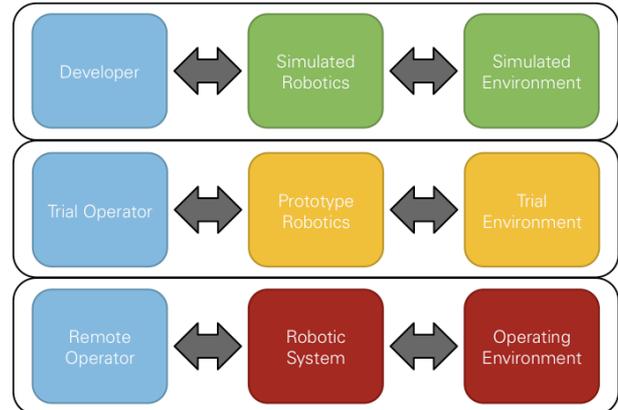


Figure 1. Traditional stages in robotic system design

## 3. Graphical System Design Approach

The graphical system design approach used by a number of developers of robotic systems can be very effective in allowing domain experts to focus on their system design and deployment and not on the individual tools and process used for development and deployment of systems. As shown in Fig.2 the concept is to allow systems to move across the design lifecycle reusing and sharing where possible tools and systems leading to a more complete development process with improved quality and faster product deployment.

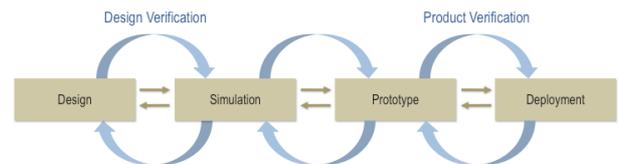


Figure 2. Linking design to deployment

Graphical system design is an approach to designing an entire system, using intuitive graphical software and off-the-shelf simulation and hardware to rapidly refine the design, create initial prototypes and more towards a final deployable system. By using a common set of tools across the design and development phases engineers can share information and processes between these phases and then link these to simulation models or hardware control as needed.

LabVIEW is recognised as the main development platform for the Graphical System Design approach and will be referenced in this article to show how the links

between simulated systems and real world data can be quickly formed and control code that has been developed in a simulated environment can be deployed to hardware quickly and easily.

With ever-changing system requirements and standards, developers are adopting a “software-first” approach for managing product lifecycles. System designer teams who use software-based tool chains that give them the ability to easily upgrade systems and solutions over time have a strategic advantage. This software-first approach can be a huge asset during the design phase, giving designers the ability to more easily take advantage of modern 32-bit processors, rapidly optimize their designs for conflicting requirements, and reduce development time. A software-first philosophy can also allow dramatic changes between initial design and prototyping phases as hardware platforms evolve. [3].

#### 4. Applying Graphical System Design to Robotics

To take advantage of the graphical system design approach system developers have to change their way of thinking about the challenges of developing a robotic system in phases and focus on the operational configuration of the final system. This is used in a variety of industries such as automotive and aerospace where system components in the development process can be virtual/simulated or physical as the car or plane moves through phases of development.

As shown in Fig.3 for a simple robotic system we can break this down into Human Interfaces (either operations or development), Robotics System (simulated, prototype or final) and the Operating Environment (simulated, trial or final).

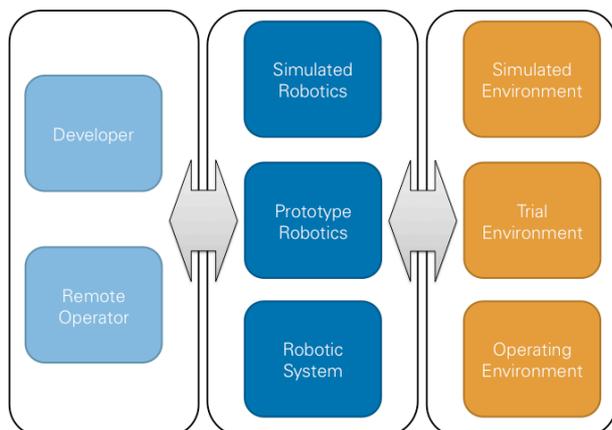


Figure 3. Shared capabilities for robotic system design

Using this approach you can break down the development and deployment of tools into these areas and with the correct development strategy each part of the system can talk to the others regardless of whether the system is simulated or physical [2].

#### 5. Simulation, Interfaces & Visualisation

Simulation is very helpful when an engineer needs to control and/or analyse a system that either doesn't exist yet or is expensive or difficult to test. Good examples are in complex equipment such as automotive and aerospace development and also remote and robotic systems. However simulation can also be used to validate prototype designs or integrated into final deployments to give powerful visualisations or operational checks. Challenges with effectively using simulation are often related to linking these simulations to live data and to deploying models into off the shelf hardware for prototyping and deployment.

Initial user interfaces could be developed specifically to interface with simulation models alongside 3D visualisation tools to display simulated environments.

An example of this is shown in Fig 4. Using the Simulation Interface toolkit to allow LabVIEW to connect display objects to I/O points within models.

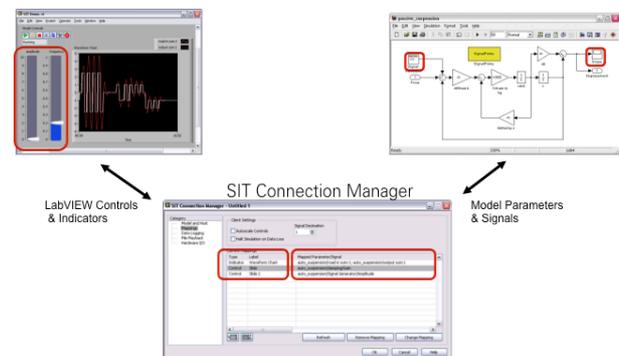


Figure 4. Customising connections between user interfaces and models using simulation interface tools.

However using the System Design approach it becomes more important to develop a software methodology so that interfaces are developed that can be reused across the development process. This means that development of an abstraction layer becomes key to making developing independent user interfaces.

By breaking out the key types of connections between models, interfaces and hardware a hardware & simulation abstraction layer can be developed as referenced in Fig 5.

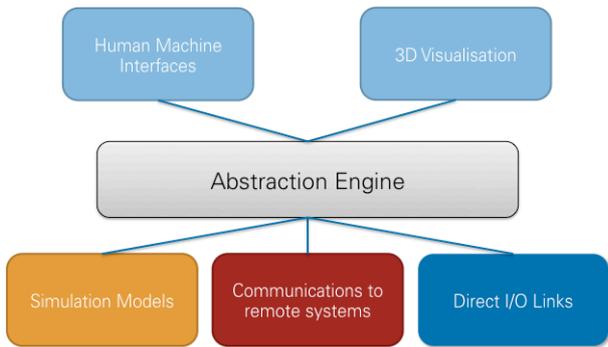


Figure 5. Abstraction between physical and simulated systems and the user interfaces.

This approach is also common in lead user research where robotics are constantly evolving in the prototyping stage so simulations are often key to avoid unnecessary hardware deployment costs with changing sensors and actuators on-board prototypes. Fig. 6 shows this the simple architecture of this case.

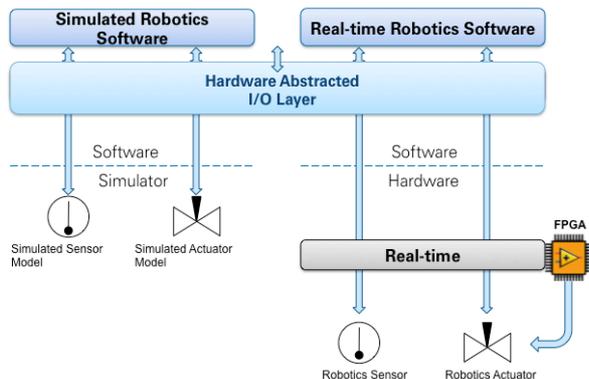


Figure 6. Hardware Abstraction Layer for I/O

This abstraction gives a common set of connection points for the interface development and also simulation, physical I/O and communication systems to remote deployments. As systems are developed further user interfaces can be adapted to take into account changes to the system without forcing a redesign of the software architecture. Likewise links can be changed between both simulation and physical systems as new technologies or hardware become available. On nearing deployment final operator interfaces can be tested on simulated systems before deployment via linking. Custom development of the abstraction layer can be a complex challenge in early stages of development but the benefits in system evolution and speed of future development generally outweigh these costs especially in complex system such as robotics. The benefits of working with common interfaces are also seen when final operators of equipment can help guide early stage development of interfaces while still in the simulation phase. Development of the abstraction layer historically is done using system design or programming tools such as LabVIEW and has enabled complex systems to be

developed in a variety of industries [2], [4].

Fig 6. Shows various options to integrate a wide range of software concepts and links to other tools and systems.

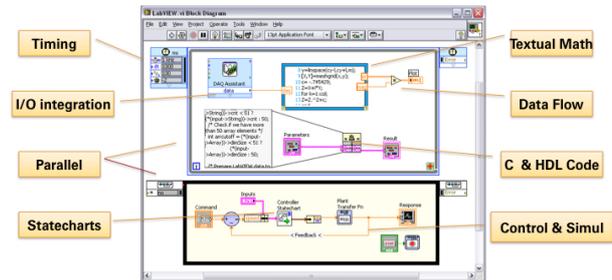


Figure 6. Using graphical system design to combine models of computation across a single system

Recent developments in model based tools also mean that real time testing tools such as National Instruments VeriStand offer a configuration based approach to system abstraction letting developers focus on the application functionality.

Regardless of the abstraction methodology the development of common user interfaces between simulation and deployment means that challenges that would normally be discovered in prototyping or deployment phases become visible at an early stage in the development process. This means that interfaces can be tested and developed in parallel to simulation and model development. Examples of linking simulated environments and visualisation can be seen in Fig.7 & Fig.8 and examples of the University of Limerick interfaces that are common between simulation and operation in Fig.9.

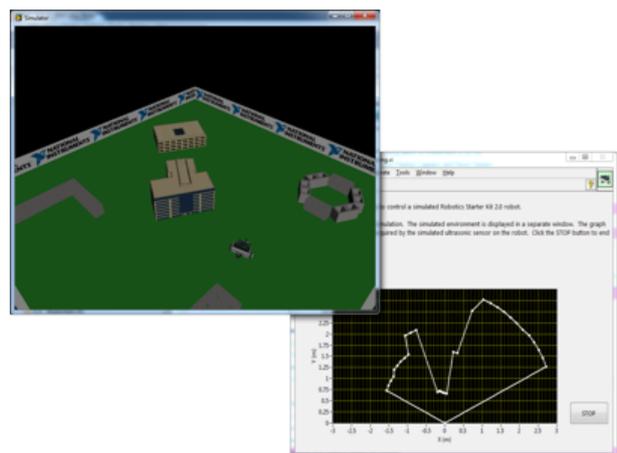


Figure 7. Linking simulated environment to real world data

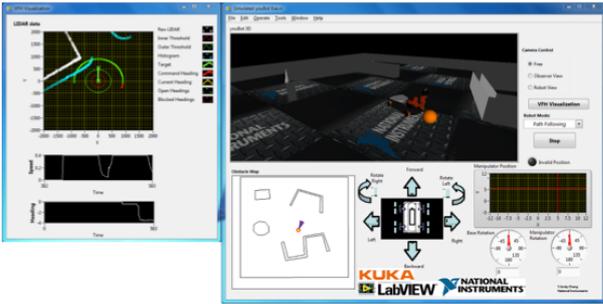


Figure 8. LabVIEW Robotics module visualisations



Figure 9. Shared user interfaces for UL Marine Robotics

## 6. Off the shelf hardware for prototyping

With an abstract layer in place to handle interfacing between interfaces and simulations developers can also take advantage of off the shelf deployment targets for prototyping control algorithms and sensors. By using the same tools that are used to develop the interfaces and communications between systems to develop measurement and control deployment platforms engineers can prototype quickly and rapidly evolve system architectures without moving to other deployment tool chains.

The Graphical System Design approach means that LabVIEW abstracts concepts such as multithreading, real time operating systems and FPGA code development from domain experts so that there is not the need to bring in deployment experts while evaluating control algorithms and system design. The RIO architecture from National Instruments takes advantage of this to offer a variety of platforms that incorporate I/O, fast control via FPGA and real time processing via microprocessor. Fig. 10.

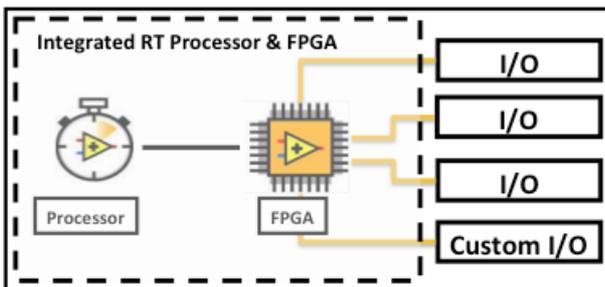


Figure 10. RIO architecture from National Instruments

In the case of both the University of Limerick [2] and MDA-US [4] the Compact RIO platform gave a small rugged form factor that could be used for both prototyping and in certain cases moving towards deployment.

Depending on development and prototyping environments a variety of platforms can be used in system development to take advantage of processing power and rugged form-factors. Fig. 11

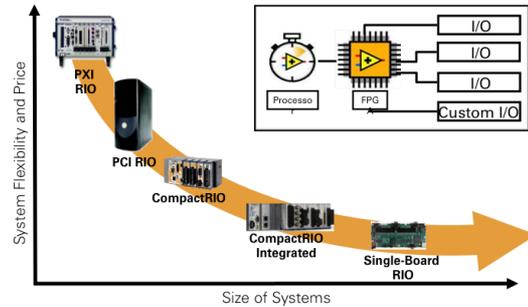


Figure 11. Range of deployment platforms using RIO architecture.

The Graphical System Design approach applied to the RIO platforms also means that simulations and models can be directly deployed to the processor component of the architecture giving access to deployed models in off the shelf hardware platforms. This means that control algorithms can continue to be developed and tested in prototyping stages without having to convert them to embedded code.

On moving to final deployment targets the abstraction layer developed enables remote targets to continue to interact via communications systems to both user interfaces and simulation environments without redevelopment of these aspects of the system.

## 7. Conclusions

Abstractions layers are critical in good system design to take advantage of simulation and physical systems. This can be to take advantage of common interfaces and simulation or to develop hardware independent software platforms.

This means that a software defined approach can rapidly move systems from design to operational deployment with custom hardware developments saved until the design is fully validated. Fig. 12



Figure 12. Moving from software to hardware using software validated design.

Engineers can develop robotic systems that integrate complex user interfaces and visualisation that use simulation models in parallel to deployed robotics to give more effective control and monitoring of activities. This configuration is shown in Fig 12 and is roughly based on the best practices learned from National Instruments working closely with Edin Omerdic's team at the University of Limerick on the underwater robotics platform OceanRINGS. [1]

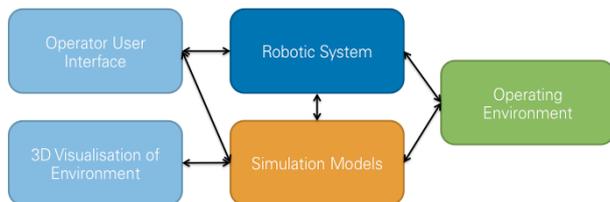


Figure 13. Deployed Robotics using simulation in parallel to aid visualisation and control.

Key benefits from the graphical system approach for robotics are

Signal-level compatibility between simulated and real-world environment improving accuracy of control algorithm develop.

Fast implementation of 3D real-time visualisation interfaces for the operating environment and robotic system data.

Open off the shelf deployment targets help with rapid prototyping and hardware-in-the-loop development of robotics systems

Software connectivity between both simulated and physical systems allows advanced user interfaces, improved reliability from shared testing in simulation and prototyping phases and reduced system development time.

## 8. REFERENCES

1. E. Omerdic: Keynote Talk “OceanRINGS: Smart Technologies for Subsea Operations” at National Instruments Offshore Measurement & Control Summit, 21 Mar 2012, Aberdeen, UK.
2. E. Omerdic, Researchers Use Graphical System Design for Development and Control of Unmanned Underwater Vehicles, Case study: [www.ni.com](http://www.ni.com)
3. Embedded Systems Outlook 2012, National Instruments [ni.com/eso](http://ni.com/eso)
4. S. Dougherty, Developing Interplanetary Robotics with NI Technology, Case study: [www.ni.com](http://www.ni.com)