

**A SOFTWARE FRAMEWORK TO SUPPORT DISTRIBUTED COMMAND
AND CONTROL APPLICATIONS**

by

Arno Duvenhage

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering (Software Engineering)

in the

Faculty of Engineering, Built Environment and Information Technology

Department Electrical, Electronic and Computer Engineering

UNIVERSITY OF PRETORIA

July 25, 2011

SUMMARY

A SOFTWARE FRAMEWORK TO SUPPORT DISTRIBUTED COMMAND AND CONTROL APPLICATIONS

by

Arno Duvenhage

Promoters: Prof G.P. Hancke and Prof D.G. Kourie
Department: Electrical, Electronic and Computer Engineering
University: University of Pretoria
Degree: Masters (Software Engineering)
Keywords: Distributed simulation, interoperability, command and control,
legacy systems, net-centric systems, IPC, middleware,
software framework, software quality, software architecture

This dissertation discusses a software application development framework. The framework supports developing software applications within the context of Joint Command and Control, which includes interoperability with network-centric systems as well as interoperability with existing legacy systems.

The next generation of Command and Control systems are expected to be built on common architectures or enterprise middleware. Enterprise middleware does however not directly address integration with legacy Command and Control systems nor does it address integration with existing and future tactical systems like fighter aircraft. The software framework discussed in this dissertation enables existing legacy systems and tactical systems to interoperate with each other; it enables interoperability with the Command and Control enterprise; and it also enables simulated systems to be deployed within a real environment.

The framework does all of this through a unique distributed architecture. The architecture supports both system interoperability and the simulation of systems and equipment within the context of Command and Control.

This *hybrid* approach is the key to the success of the framework. There is a strong focus on the quality of the framework and the current implementation has already been successfully applied within the Command and Control environment. The current framework implementation is also supplied on a DVD with this dissertation.

OPSOMMING

'N SAGTEWARERAAMWERK WAT BEVEL- EN BEHEERTOEPASSINGS ONDERSTEUN

deur

Arno Duvenhage

- Studieleiers: Prof G.P. Hancke en Prof D.G. Kourie
- Departement: Elektriese, Elektroniese en Rekenaaringenieurswese
- Universiteit: Universiteit van Pretoria
- Graad: Magister (Sagteware Ingenieurswese)
- Sleutelwoorde: Verspreide simulاسie, interoperabiliteit, bevel en beheer,
ouderwetse stelsels, netwerkgesentreerde stelsels, IPC, middleware,
sagtewareraamwerk, sagtewarekwaliteit, sagtewareargitektuur

Hierdie verhandeling bespreek 'n sagtewareraamwerk wat gebruik kan word om toepassings in die bevel- en beheeromgewing te ontwikkel. Hierdie tipe toepassings sluit die interoperabiliteit met netwerkgesentreerde stelsels sowel as interoperabiliteit met ouderwetse militêre stelsels in.

Die volgende generاسie van bevel- en beheerstelsels gaan heelwaarskynlik geskoei wees op die tipe middleware wat algemeen in die besigheidswêreld voorkom. Hierdie tipe middleware spreek ongelukkig net nie die integrاسie van bevel- en beheerstelsels, operاسionele stelsels of taktiese stelsels aan nie. Die sagtewareraamwerk wat in hierdie verhandeling bespreek word, help met die integrاسie van ouderwetse stelsels, taktiese stelsels en bevel- en beheer besigheidsagteware. Dit vergemaklik ook die ontplooiing van gesimuleerde stelsels in die regte wêreld.

Die raamwerk doen al hierdie dinge deur 'n unieke verspreide argitektuur. Die argitektuur ondersteun interoperabiliteit en die simulاسie van bevel- en beheerstelsels en -toerusting. Hierdie tweevoudige argitektuur is die geheim vir die sukses van die raamwerk. Daar is 'n baie sterk fokus op die kwaliteit van die raamwerk en die raamwerk word tans gebruik om suksesvolle bevel- en beheertoepassings te ontwikkel. Die raamwerk is op 'n DVD saam met hierdie verhandeling ingesluit.

LIST OF ABBREVIATIONS

List of abbreviations where the notation is different than the norm.

ADC	Air Defence Control
JC2	Joint Command and Control
GBADS	Ground Based Air Defence System
MSDS	Modelling and Simulation based Decision Support
OIL	Operator In the Loop
OT&E	Operational Testing and Evaluation
VGD	Virtual GBADS Demonstrator

Contents

INTRODUCTION	1
1 Background	3
1.1 The Software Application Framework	3
1.2 Modelling and Simulation	4
1.3 C2/M&S Interoperability	4
1.4 Supporting the C2 Enterprise	6
2 Research Overview	9
2.1 Research Characterisation	9
2.2 Research Plan and Dissertation Outline	9
LITERATURE REVIEW	13
3 Software Architecture	15
3.1 Moving from Stovepipe to Network-Centric Architectures	15
3.2 Key Architectural Styles	16
3.3 Design Patterns	17
3.3.1 Creational Patterns	18
3.3.2 Structural Patterns	18
3.3.3 Behavioural Patterns	19
3.3.4 Service Access and Configuration Patterns	20
3.3.5 Event Handling Patterns	20

3.3.6	Synchronisation Patterns	20
3.3.7	Concurrency Patterns	21
3.4	Publish/Subscribe Networking	21
3.5	Service Oriented Architecture	22
3.6	Software Frameworks	23
3.7	The OSI Reference Model	23
4	Formal Analysis of Software Behaviour	25
4.1	UML Use Case Diagrams	25
4.2	UML Finite State Machines	25
4.3	Communicating Sequential Processes	26
4.3.1	Language Constructs	26
4.3.2	Describing Distributed Simulators	27
5	Distributed Simulation	29
5.1	SIMNET	29
5.2	NPSNET	31
5.3	DIS	32
5.4	HLA	32
5.4.1	An Overview of the HLA	33
5.4.2	An Overview of the HLA Evolved	35
5.5	VGD	36
5.5.1	VGD 2	36
5.5.2	VGD 3.0	37
5.5.3	VGD 3.1	39
5.5.4	Migrating to a Quantised Discrete Event Architecture	40
6	System Interoperability	43
6.1	System Interoperability and Joint Command and Control	43
6.1.1	Tactical Networks	44

6.1.2	C2/M&S Protocol Gateways	45
6.2	The Command and Control Enterprise	46
6.2.1	Enterprise Services	46
6.2.2	The Enterprise Service Bus	47
FRAMEWORK IMPLEMENTATION		49
7	Framework Requirements	51
7.1	Framework Use Case	51
7.2	Framework Requirements	52
7.2.1	Interoperability with C2 Systems	53
7.2.2	Virtualisation of C2 equipment using M&S	53
7.2.3	Application Development	53
7.2.4	Good Code Quality	54
7.2.5	Performance and Portability	54
8	Framework Design and Implementation	55
8.1	Design Overview	55
8.2	The Backbone Layer	57
8.2.1	Inter Object Communication	57
8.2.2	Inherent Object Construction	59
8.2.3	Distributed Object Execution	59
8.2.4	Subscriptions and Publications	60
8.2.5	Core Backbone Components	61
8.3	The Infrastructure Layer	62
8.3.1	Spatial Reference and Environment Models	62
8.3.2	The Bootloader	63
8.3.3	The Node Hub	63
8.3.4	Information Representation and Translation	64
8.4	The Interoperability Layer	65

8.5	The Simulation and Application Layers	66
8.5.1	Application Integration	66
9	Framework Evaluation	69
9.1	Performance and Scalability Testing	69
9.1.1	Expected Behaviour	70
9.1.2	Parallel Performance	71
9.1.3	Distributed Performance	74
9.2	Application Examples	77
9.2.1	A Simulation of Flocking Behaviour	79
9.2.2	Conway's Game of Life	80
9.2.3	An Tactics Evaluation Tool for Fighter Aircraft	82
9.2.4	A Command and Control Protocol Gateway	83
9.2.5	A Radar Emulator	84
9.2.6	A Joint Operations Operator Console	85
9.3	Formal Evaluation	87
9.3.1	Distributed Execution	87
9.3.2	The Frame Execution and Multi-threading	90
9.4	General Discussion	93
	CONCLUSION	95
10	Conclusion	97
10.1	The Framework Implementation	97
10.2	Future Work	98
10.3	An Open Unified Architecture for System Development	98
10.4	Final Thoughts	98