

LB/DON/01/2019

CS 02/77

**PERCEPTION.JS – A FRAMEWORK FOR CONTEXT
ACQUISITION, PROCESSING AND PRESENTATION**

Supun Dissanayake

(148210U)

**LIBRARY
UNIVERSITY OF MORATUWA, SRI LANKA
MORATUWA**

M.Sc. in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri-Lanka



TH3708

May 2018

TH 3708

+

CD-ROM

004"18"

004(043)

TH 3708

**PERCEPTION.JS – A FRAMEWORK FOR CONTEXT
ACQUISITION, PROCESSING AND PRESENTATION**

Supun Dissanayake

(148210U)

**Thesis/Dissertation submitted in partial fulfillment of the
requirements for the degree M.Sc. in Computer Science Specializing
in Software Architecture**

Department of Computer Science and Engineering


**University of Moratuwa
Sri-Lanka**

May 2018

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: .....

Date: 05/06/2018

Name: Supun Dissanayake

The supervisor/s should certify the thesis/dissertation with the following declaration.

I certify that the declaration above by the candidate is true to the best of my knowledge and that this report is acceptable for evaluation for the CS6997 MSc Research Project.

Signature of the supervisor: ... *UOM Verified Signature*

Date: 2018/06/06

Name: Dr. Malaka Walpola

ABSTRACT

Context Awareness which is an area of Pervasive Computing that enables a person to accomplish his day-to-day tasks by seamlessly interacting with the “smart” devices that are embedded in the environment (smart space). In contrast to how a user interacts with a desktop computer or a mobile device using various input/output devices, Pervasive Computing paradigm acquires user’s context using sensors embedded in the surrounding environment, and identifies the actions the user would need to perform in a specific context. The Pervasive Computing application would then perform the required action on behalf of the user or it would give recommendations on the action the user would need to perform.

Because the number of smart devices are being produced and increased rapidly the demand for context awareness applications also increase, software developers can exploit the new computing paradigm to provide more innovative user-centered software solutions. However, the biggest obstacle for Context Awareness application development is its high complexity due to its broad technical areas (i.e. handling sensor imperfections, modelling smart environments, inferencing context, integrating with heterogenous systems or sensors, etc.) Hence software developers fail to provide quality context awareness applications that meet end-user requirements or fail to accurately identify context. Additionally, such software development increases project schedules, and could increase its bug rate.

This research project addressed the above problems by developing a software framework that enables the software developers to develop their applications using the fundamental features of Context Awareness such as Context Acquisition, Context Processing and Context Presentation. Apart from its functionality this research project focused on enhancing the quality of the framework by introducing quality attributes such as extensibility (which enables the developers to address the problem of heterogeneity), portability (which enables toe developers to use the framework in various devices and platforms), and usability (which enables the framework more usable by the developers).

From the technical perspective, the framework is based on the architecture of Sentient Object which this project aims to implement the architecture using a JavaScript technology stack. JavaScript enables to mitigate the problem of heterogeneity because the technology stack that will be used to develop the framework consist of Apache Cordova which enables to implement sensing mechanisms in a broad range of smart devices, and Node.js which enables to execute the context server in multiple platforms.

One of the most prominent aspect of the framework is that when the framework is embedded in a JavaScript application, the framework can transform the application into a Sentient Object. A Sentient Object can acquire contextual information using sensors, model and processes the context using an inferencing engine, and to respond to context changes using actuators. Scalability can also be achieved through Sentient Objects which can separate contextual information capture from context processing using a context server approach.

ACKNOWLEDGEMENTS

I would like to express profound gratitude to my advisor, Dr. Malaka Walpola, for his invaluable support by providing relevant knowledge, materials, advice, supervision and useful suggestions throughout this research work. His expertise and continuous guidance enabled me to complete my work successfully.

Further I would like to thank all my colleagues for their help on finding relevant research material, sharing knowledge and experience and for their encouragement. I am as ever, especially indebted to my parents for their love and support throughout my life. Finally, I wish to express my gratitude to all my colleagues at 99X Technology, for the support given me to manage my MSc research work.

Table of Contents

DECLARATION	i
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	vi
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
INTRODUCTION	1
1.1. Overview of Pervasive Computing	2
1.2. Overview of Context Awareness	3
1.3. Problem Background	4
1.4. Problem Statement	6
1.5. Research Objectives	6
1.6. Solution Outline	6
1.7. Outline of the Thesis	8
LITERATURE REVIEW	9
2.1. Pervasive Computing	10
2.2. Context Awareness	12
2.2.1. Introduction	12
2.2.2. Context acquisition	13
2.2.3. Context processing	18
2.2.4. Context presentation	25
2.3. Related Work	27
2.3.1. Context-aware frameworks	27
2.3.2. Context-aware systems	28
2.4. Software Reusability and Software Frameworks	30
2.4.1. Usability of Frameworks	31
2.4.2. Architectural Pattern for the Framework	35
2.5. JavaScript and its Related Technologies	38
2.5.1. JavaScript programming paradigm	38
2.5.2. JavaScript design patterns	38
2.5.3. JavaScript Development Stack	40
DESIGN AND IMPLEMENTATION	43
3.1. Introduction	44
3.2. Requirement Analysis	44
3.2.1. Requirements of the framework	44

3.2.2. Use case scenarios.....	47
3.3. Architectural Design.....	48
3.2.1. Architectural considerations	48
3.2.2. Implementation of the architectural baseline	52
3.4. Detailed Design and Implementation.....	53
3.4.1. Portability.....	53
3.4.2. Bootstrapping process	54
3.4.3. Extensibility	55
3.4.4. Context representation	58
3.4.5. Context processing.....	59
3.4.7. Context acquisition and presentation	65
3.5. Integrating with Existing Technologies	66
3.5.1. Integrating with technologies to acquire contextual information	66
3.5.2. Integrating with technologies to store contextual information	70
3.5.3. Integrating with technologies to process contextual information	71
3.6. How to Use the Framework	72
3.6.1. Getting Started	72
3.6.2. Using Framework Features	77
3.6.3 Best Practices	80
TESTING AND ANALYSIS	83
4.1. Introduction.....	84
4.2. Unit Testing	84
4.2. System Testing.....	93
4.2.1. Testing of functionality, and extensibility	93
4.2.2. Testing of context server approach, and portability.....	97
4.3. Usability Testing.....	100
4.3.1. Test goals	100
4.3.2. Test scenario	100
4.3.3. Test process.....	101
4.3.4. Test findings	106
4.3.5. Analysis of usability	108
CONCLUSION.....	111
5.1. Summary of the Developed Framework.....	112
5.2. Advantages of the Framework.....	112
5.3. Limitations.....	113
5.4. Future Enhancements.....	113
5.5. Conclusion	114
REFERENCES	116

LIST OF FIGURES

		Page
Figure 1.1	Definition of context	3
Figure 2.1	Ubiquitous computing framework	10
Figure 2.2	Layered conceptual framework for context-aware systems	14
Figure 2.3	Use of aggregation to obtain more accurate contextual information	16
Figure 2.4	Implementation effort to handle uncertainty in various methods	17
Figure 2.5	CONON ontology for context awareness applications	22
Figure 2.6	Domain-Specific ontologies and upper ontologies in CONON	23
Figure 2.7	Learning barriers for APIs and how they relate each other	34
Figure 2.8	Sense compute control architectural pattern	36
Figure 2.9	Sentient object model architecture	36
Figure 2.10	Non-blocking I/O feature of Node.js	41
Figure 2.11	PhoneGap architecture	42
Figure 2.12	Platform support for Apache Cordova	42
Figure 3.1	Use case diagram of Perception.js	47
Figure 3.2	Architecture of Perception.js	48
Figure 3.3	Internal representations of Perception.js	50
Figure 3.4	Class diagram of Perception.js	51
Figure 3.5	Integrating extensions	55
Figure 3.6	Default ontology in Perception.js	58
Figure 3.7	Internal representation of contexts using expression trees	60
Figure 3.8	How actuators are triggered	61
Figure 3.9	How expression trees are created for sub classes	62

Figure 3.10	Polling consumer integration pattern	67
Figure 3.11	How polling consumer integration pattern is used	67
Figure 3.12	Event driven consumer integration pattern	68
Figure 3.13	How event driven consumer integration pattern is used	68
Figure 3.14	How storage extensions can be used to integrate storage mechanisms	70
Figure 3.15	Integrating external context processing systems	71
Figure 4.1	Architecture of the first sample application	94
Figure 4.2	Ontology of the first sample application	94
Figure 4.3	Architecture of the second sample application	98
Figure 4.4	Architecture of the third sample application	102
Figure 4.5	Class diagram of the client application of the third sample application	103
Figure 4.6	Class diagram of the context server of the third sample	104
Figure 4.7	Work breakdown structure of the third sample application	105

LIST OF TABLES

		Page
Table 2.1	Appropriateness indications of context modelling approaches	21
Table 4.1	Unit testing results of the component: Helpers	84
Table 4.2	Unit testing results of the component: AsyncIterator	85
Table 4.3	Unit testing results of the component: EventManager	86
Table 4.4	Unit testing results of the component: ConfigurationManager	86
Table 4.5	Unit testing results of the component: RuleManager	87
Table 4.6	Unit testing results of the component: TransportManager	88
Table 4.7	Unit testing results of the component: OntologyManager	88
Table 4.8	Unit testing results of the component: OntologyClass	89
Table 4.9	Unit testing results of the component: OntologyProperty	90
Table 4.10	Unit testing results of the component: Scheduler	90
Table 4.11	Unit testing results of the component: ExpressionTree	91
Table 4.12	Unit testing results of the component: BinaryExpression	91
Table 4.13	Unit testing results of the component: UnaryExpression	92
Table 4.14	Test results of the first test scenario	96
Table 4.15	Test results of the second test scenario	99
Table 4.16	Developer feedback questionnaire	106
Table 4.17	Time taken complete each task	107

LIST OF ABBREVIATIONS

API	Application Programming Interface
CONON	Context Ontology
GPS	Global Positioning System
GPIO	General Purpose Input/Output
HCI	Human Computer Interaction
HMM	Hidden Markov Model
HTML	Hyper Text Markup Language
IO	Input Output
IoT	Internet of Things
JS	JavaScript
JSON	JavaScript Object Notation
MQTT	Message Queueing Telemetry Transport
NPM	Node Package Manager
NoSQL	Not Only Structured Query Language
OO	Object Oriented
OWL	Ontology Web Language
QOS	Quality of Service
REST	Representational State Transfer
RFID	Radio-Frequency Identification
SCC	Sense Compute Control
SDK	Software Development Kit
SOA	Service Oriented Architecture
TCP	Transmission Control Protocol
SDK	Software Development Kit
SOCAM	Service Oriented Context Awareness Middleware
SWRL	Symantec Web Rule Language
UML	Unified Modelling Language
URL	Uniform Resource Locator
W3C	Worldwide Web Consortium
WBS	Work Breakdown Structure
XML	Extensible Markup Language