Master's Thesis

A mediator system for querying heterogeneous data in robotic applications

Rubanraj Ravichandran

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Erwin Prassler
Prof. Dr. Manfred Kaul
Nico Huebel
Sebastian Blumenthal

April 2019

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

____________________ Date

____________________ Rubanraj Ravichandran

Abstract

Your abstract

Acknowledgements

Thanks to ....

Contents

List of Figures

List of Tables

1

Introduction

Robots generate a large amount of data from different types of sensors attached to it and also from its hardware components. In our previous research work [5], we have conducted an extensive qualitative and quantitative analysis to find better databases and architectures that effectively store these data and consume it for further operations. Results from our previous work show that a single database is not suitable for every robotic scenario. For example, in terms of handling large BLOB data, MongoDB stored them faster but reading the data was slower compared to CouchDB [5]. Also, to complete a given task robot depends on multiple sources of information from internal sensors, as well as external sources for example world model, kinematic model, etc..

Adoption of multiple databases for robotic applications requires a unique way of mediation to view multiple databases as a single federated database. Mediator approach helps to integrate data from different sources and produce an only result back to robots. Mediator abstracts the information of how data is being stored in various data sources from a robot and allows robotic applications stream data to mediator independent of databases used in the back-end.

To Map the data generated by robots with multiple databases, the mediator system requires a proper data model predefined in the context of robotic applications. Modeling robot produced data helps to generalize the structure of data and defining relations between different entities (e.g., tasks, sensors, robots, location) in a robotic application scenario. If we have a well defined robotic data models, then the mediator will get the ability to mutate or query data from different data sources. Also, it is essential that any robotic use-cases should be able to extend these data models.

As mentioned in these papers [1, 4, 2, 3, 3, 6], mediators are being used to integrate data from different data sources, and few architectures support single data model (e.g., SQL), and others recommend for different data models (e.g., SQL, NoSQL, document store, etc..). Also, they differ from query languages, ease of implementation, components used in their architecture. This project mainly focuses on defining semantic based models for sensor data to make it more interoperable with other systems or even in multi-robot systems, and implementing a mediator system which acts as a middle-ware between robots and databases.

1.1 Motivation

Streamlining the data produced from different sensors in robotic applications is a tedious task, and there are no specific standards to organize the data in terms of making relations between the entities and also giving context to the data. It will be even more complicated when we have a multi-robot platform and sharing data between them, and backing up the data into a database for fault diagnosis.

Currently, in the ROPOD[1] project, there is a single black box component has been developed to simulate the robot test cases. During the simulation black box stores the data produced by the sensors as dumps into a single MongoDB instance locally.

The first problem here is since the sensor data stored as dumps which makes the consumer's[2] inability to make queries against the data.

And the second problem is missing contexts and the entity-relationship model. For example, if a consumer tries to query the data from dumps, it will be unsure that which sensor produced this data from which robot/black-box at which location and time, and who triggered this test case. What we mean "missing context" is if humans read the data they will understand what's the meaning of each parameter, but if a different robot/black-box tries to consume the data produced by other robots, then the context about the data should be shared somewhere globally.

[1] ROPOD is a EU funded project to develop "Ultra-flat, ultra-flexible, and cost-effective robotic pods for handling legacy in logistics"

[2] A consumer can be either humans or machines.

The final problem is, what if we have a situation where multi-robots tries to share data or human controller wants to do fault-diagnosis on data shared on multi-robots.

These significant issues inspired us to find a suitable Entity-Relationship data model and unique mediation system to query heterogeneous sensor data from multiple data sources regardless of the database type.

1.2 Structure

- Section x concisely describes the background knowledge of the topics which are relevant to this work.

2 Problem Statement

Our previous work results reveal not all databases reacts similarly for different heterogeneous data from robotic applications. Also, there are no concrete data models has been defined in the context of robotic applications. For example, the black box designed for ROPOD project uses MongoDB to store data from different sources such as Ethercat, Zyre, ZMQ, and ROS topic. The data is being transformed into a simple flatten JSON document to store the values. These documents are stored under a single collection which is created for each ROS topic or other sources. Each record holds only the information of data generated by the sensors or application itself, but these values are not useful without additional details for example, who created the data, if it is a robot then what type of robot-generated this data from which location? Then in what context other systems should interpret this data.

Listing 2.1: geometry_msgs/Pose ROS topic

```
double timestamp
double position/x
double position/y
double position/z
double orientation/x
double orientation/y
double orientation/z
double orientation/w
```

For example in the black box, geometry_msgs/Pose ROS topic will be flattened to a simple JSON document which has the data structure mentioned above.

In the above format, 'position/x' is a key and the value will be attached with it. Now only with position x,y,z and orientation x,y,z,w, another system which consumes this data would not be able to say who generated this data or at which location this data is being generated and if the other system is doing mathematical calculation, then this data is missing its own context such as unit, dimensions, etc.

Periodically, these massive amounts of data are dumped and backed up to a file system or cloud. After every test run in the black box, a report is generated using the FMEA tool which contains the information regarding the test and components involved in it. Also, these reports include the file location where the dump is stored. During fault diagnosis, these dumps will be restored manually to the database and fetch data using the querying tool provided by the black box itself.

This approach is not scalable and inefficient in terms of multi-robot systems since there will be individual database instances running in each robot. Moreover, this querying tool is incapable of making queries on multiple MongoDB instances at a time.

In terms of supporting various types of databases setup for robots, there is no systematic approach to store and retrieve data from external sources. Also, a well-defined data model hasn't defined yet that can map robot components (e.g., sensors) to a robot and even with the world model (e.g., locations). In this case, no mediator system has been developed before to connect between robots and different databases.

A

Design Details

Your first appendix

B

Parameters

Your second chapter appendix

References

[1] Rafi Ahmed, Philippe DeSmedt, Weimin Du, William Kent, Mohammad A. Ketabchi, Witold A Litwin, Abbas Rafii, and M-C Shan. The pegasus heterogeneous multidatabase system. *Computer*, 24(12):19–27, 1991.

[2] Yigal Arens, Chun-Nan Hsu, and Craig A Knoblock. Query processing in the sims information mediator.

[3] Sudarshan Chawathe, Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman, and Jennifer Widom. The tsimmis project: Integration of heterogenous information sources. 1994.

[4] Gustav Fahl, Tore Risch, and Martin Sköld. Amos-an architecture for active mediators. 1993.

[5] Rubanraj Ravichandran, Nico Huebel, Sebastian Blumenthal, and Erwin Prassler. A workbench for quantitative comparison of databases in multi-robot applications. 2018.

[6] Kurt Shoens, Allen Luniewski, Peter Schwarz, Jim Stamos, and Joachim Thomas. The rufus system: Information organization for semi-structured data.