FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Smart 2PL Inventory Transfers Management

A Distributed Simulation with Agent Process Models and Microservices Data Mining

José Pedro Teixeira Monteiro

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Dr. Henrique Lopes Cardoso

Co-supervisor: Thiago RPM Rúbio

July 25, 2019

Smart 2PL Inventory Transfers Management

José Pedro Teixeira Monteiro

Mestrado Integrado em Engenharia Informática e Computação

July 25, 2019

Resumo

Com a criação e desenvolvimento de novas tecnologias, novas soluções estão a ser estudadas para solucionar problemas em ambientes logísticos. A gestão de estoque 2PL é um caso específico de logística onde a transferência de mercadorias é feita pelas transportadoras e não pela empresa que as produz. Neste contexto, a previsão de possíveis erros está a tornar-se muito significativa, para que as empresas possam cumprir as promessas feitas aos clientes (entregar produtos no prazo) e também melhorar os prazos de entrega estimados. Acreditamos que isso pode ser alcançado com a recolha de dados do ambiente. Essa recolha pode ser feita com dispositivos reais ou virtuais, cuja integração pode ser chamada de Internet das Coisas (IoT), um campo de pesquisa crescente. Diferentes abordagens para explorar os dados recolhidos podem ser tomadas, com o objetivo de adquirir conhecimento que possa ser aplicado no ambiente.

Devido à descentralização na gestão de transferências e à possibilidade de representar autonomamente os processos manuais atuais que estão presentes na gestão de stock, a revisão de literatura realizada, aponta o uso de sistemas baseados em agentes como uma boa abordagem para representar entidades, orquestrando as interações entre eles. Isso significa que um agente - uma entidade de software autônoma que representa um ator no sistema e age de acordo com suas percepções e objetivos - pode se comunicar com entidades que estão presentes no ambiente e agir com os recursos que possui. Desafios importantes nesta área compreendem o estabelecimento de formatos e semânticas de mensagens comuns, bem como a determinação de quando os agentes podem ser confiáveis, em particular em situações em que a tomada de decisões pode ter um grande impacto no desempenho. Posto isto, o uso de Data Mining é introduzido para funcionar como uma solução, treinando diferentes modelos com dados recolhidos do ambiente para obter uma previsão válida para decisões futuras.

O objetivo principal desta dissertação é estudar as transferências de stock entre armazéns. Isso compreende um cenário distribuído, no qual nos concentramos no caso particular do 2PL, onde temos armazéns de propriedade de uma empresa que solicita transportadoras para fazer o transporte entre eles. Uma abordagem natural é modelar as entidades relacionadas e seus relacionamentos com um modelo de sistema multiagente que representa um cenário de aplicação real, onde um armazém é responsável por gerir as entregas que foram previamente agendadas, levando em consideração os desempenhos anteriores dos parceiros (camiões), a fim para evitar atrasos, melhorar o tempo de entrega, etc. Os dispositivos de IoT podem ser usados para coletar dados sobre os veículos (localização, tempo de viagem, etc.) para avaliar seu desempenho. No nosso caso, construímos uma plataforma de simulação que simula o movimento dos cameões. A solução é avaliada em termos de atraso (diferença entre tempos de chegada estimados e o que efetivamente foi alcançado) e as experiências realizadas mostram que nossa abordagem pode ser vista como uma ferramenta adequada para estudar novas políticas de decisão e planeamento para reduzir atrasos, número de falhas e prazos de entrega.

Abstract

With the creation and development of new technologies, new solutions are being studied to solve problems around logistics environments. 2PL inventory management is a specific case of logistics where the transfer of goods is made by carriers and not by the company that produces them. In this context, the prediction of possible disruptions is becoming very significant, so companies can comply with promises that are made to clients (deliver products on time) and also to improve estimated delivery times. We believe that this can be achieved with the collection of data from the environment. This collection can be made with real or virtual devices, whose integration can be called Internet of Things (IoT), an increasing research field. Different approaches to exploit the collected data can be taken, with the aim of acquiring knowledge that can be applied in the environment.

Due to the decentralisation in the management of transfers and the possibility of autonomously representing what are currently manual processes that are present in a inventory management, our literature review points towards using agent-based systems as a good approach to represent entities, orchestrating the interactions between them. This means that an agent – an autonomous software entity that represents an actor in the system and acts accordingly to its perceptions and goals – can communicate with entities that are present in the environment and act with the capabilities that it has. Important challenges in this area comprise establishing common message formats and semantics, as well as determining when agents can be reliable, in particular in situations where decision-making can have a big impact in performance. This is where Data Mining enters to work as a solution, by training different models with data collected data from the environment, to get a valid prediction for future decisions.

The main goal of this dissertation is to study inventory transfers between warehouses. This comprises a distributed scenario, in which we focus in the particular case of 2PL, where we have warehouses owned by a company that requests carriers to do the transports between them. A natural approach is to model the related entities and their relationships with a multi-agent system model that represents a real application scenario, where a warehouse needs to manage the deliveries that were previously scheduled, taking into consideration partners (trucks) previous performances in order to prevent delays, improve delivery time, etc. IoT devices can be used to collect data about the vehicles (location, time spent, etc) in order to evaluate their performance. In our case, we built a simulation platform that simulates the trucks movement. The solution is evaluated in terms of the delay (difference between estimated and actual arrival times) and the performed experiments show that our approach can be seen as a suitable tool for studying new decision and planning policies for reducing delays, number of failures and delivery times.

Keywords: Multi-agent Systems, 2PL inventory management, Supply Chain, Internet of Things, Logistics, Microservices

Acknowledgements

I would like to thank my supervisor, Dr. Henrique Lopes Cardoso, and my co-supervisor, Thiago Reis Pedroso Munhoz Rúbio, for all the help, feedback and support provided during the research and development for this thesis. Their knowledge and experience was a big help to achieve the results we got.

A big thanks to my friends, for all the support and help. We pushed each other, making us better professionals and persons.

Now, from the bottom of my heart, I have to say thank you to my family, my mother, my father, my sister who is and will be forever like a mother to me, my brother in law, my nephews and my girlfriend, Raquel, who has been my support in everything I do in life. For all the good and bad moments spent, there is nothing like family, to be in our side, to help us conquer the goals in life and go through the problems. For everything they abdicated in life for me, now is time for me to retribute. Everything that I concurred so far and that I will concur in the future, it is everything dedicated to them.

José Pedro Teixeira Monteiro

"Be grateful with everything you have and you will be successful in everything you do."

Conor McGregor

Contents

1 Introduction **1**
- 1.1 Overview . . . 1
- 1.2 Problem contextualisation . . . 4
  - 1.2.1 Application domain . . . 4
- 1.3 Motivation and Goals . . . 5
- 1.4 Research methodology and questions . . . 7
  - 1.4.1 Hypothesis and Research Questions . . . 7
- 1.5 Document Organisation . . . 8

2 Inventory Transfers Management **11**
- 2.1 Warehouse Transfers and Agents . . . 12
- 2.2 Business Process Modelling . . . 13
- 2.3 Warehouse Transfers Research and Evaluation . . . 16
- 2.4 Summary . . . 16

3 State-of-the-Art Review **19**
- 3.1 Literature Review Methodology . . . 19
- 3.2 New challenges in Supply Chain . . . 20
- 3.3 Multi-Agent Systems . . . 21
  - 3.3.1 Interaction in Agents . . . 22
  - 3.3.2 Agents Frameworks and Platforms . . . 23
  - 3.3.3 Agents and Data Mining . . . 24
- 3.4 Multi-agent system applied in logistics . . . 25
- 3.5 Microservices . . . 27
- 3.6 Summary . . . 27

4 An Intelligent 2PL Inventory Management Approach **29**
- 4.1 Introduction . . . 29
- 4.2 Overall architecture . . . 30
  - 4.2.1 Canvas service . . . 31
  - 4.2.2 Manager service . . . 33
  - 4.2.3 Data service . . . 34
- 4.3 Agent Process Modelling platform . . . 35
- 4.4 Simulation agents . . . 38
  - 4.4.1 Warehouse agent . . . 39
  - 4.4.2 Truck agent . . . 42
  - 4.4.3 Ticker agent . . . 44
  - 4.4.4 Informer agent . . . 44

4.5 Data Mining as a Service (DMaaS) 45
4.6 Simulation Workflow 47
4.6.1 Create simulation 48
4.6.2 Run simulation 50
4.7 Summary 53

5 Experimental Evaluation **55**
5.1 Introduction 55
5.2 Problem modelling 55
5.2.1 Evaluation Metrics 56
5.3 Simulation without Data Mining 57
5.4 Simulation with Data Mining 60
5.5 Challenges and limitations 63
5.6 Summary 63

6 Conclusions and Future Work **65**
6.1 Overview 65
6.2 Future work 66

References **69**

A Agents Processes **73**

List of Figures

1.1 IoT device classes bridging the gap in cyber-physical systems. 3

2.1 Warehouse transfers representation with info points defined. 12
2.2 Elements and respective groups present in business process modelling notation. . 14
2.3 BPMN elements used with the respective names and groups. 15

3.1 Agent representation (Adapted from [Russell et al., 1995]) 22
3.2 FIPA-ACL message structure. 23
3.3 Agent mining interactions. 25
3.4 Clusters of devices seen as an entity. 26

4.1 Overall architecture of simulation platform and interactions between services. . . 32
4.2 Canvas service framework and main libraries used. 33
4.3 Manager service framework and main libraries used. 34
4.4 Manager service and APM platform interactions. 35
4.5 Data service framework and main libraries used. 36
4.6 APM agent is autonomous process that orchestrates microservices as capabilities. 36
4.7 Agent process interaction with external service capabilities 38
4.8 Agent communication scheme using message events 39
4.9 Agent Process Modelling relationships 40
4.10 Warehouse agent process model for simulation 1. 40
4.11 Warehouse agent process model for simulation 2. 41
4.12 Truck agent process model for simulation 1. 42
4.13 Truck agent process model for simulation 2. 43
4.14 Ticker agent process model. 44
4.15 Informer agent process model. 45
4.16 Agent with IoT results in a agnostic cyber-physical system. 46
4.17 Simulation options 47
4.18 State Diagram representation of respective flow to create a simulation. 48
4.19 Create simulation menu with the different states. 1 - Place the warehouse. 2 - Select the warehouses and then define the info points between them. 49
4.20 Diagram with the sequence of messages exchanged between the services, APM and agents to start a simulation. 50
4.21 Diagram with the sequence of messages exchanged between the services and truck agents during a simulation of a transport. 51
4.22 Simulation running with trucks transporting units between warehouses. 52

5.1 Representation of the entities in the scenario. 56
5.2 Simulation scenario used in the experiences. 57

5.3 Number of transfers made by company, in first experiment. 58
5.4 Median percentage of delay, per trucks for each info point, in first experiment. . . 59
5.5 Average percentage of delay, per company for each info point, in first experiment. 59
5.6 Auto ML results for each model. 60
5.7 Auto ML Root Mean Squared Error results for each model. 60
5.8 Decision tree with the delays that are predicted. 60
5.9 Example of a decision tree to classify a delay. 61
5.10 Number of transfers made, by company, in second experiment. 62
5.11 Median percentage of delay, per trucks for each info point, in second experiment. 62
5.12 Average percentage of delay, per company for each info point, in second experiment. 63

A.1 Informer agent process model. 73
A.2 Ticker agent process model. 74
A.3 Warehouse agent process model for simulation 1. 75
A.4 Warehouse agent process model for simulation 2. 76
A.5 Truck agent process model for simulation 1. 77
A.6 Truck agent process model for simulation 2. 78

List of Tables

3.1 Literature keywords 20

4.1 Schedule file representation of its structure. 37
4.2 Start variable of warehouse agent. 41
4.3 Start variable of truck agent. 43
4.4 Start variable of informer agent. 44
4.5 Start variable of informer agent. 45
4.6 Example of data structure uploaded to RapidMiner as a CSV file. 46
4.7 Storage on Redis with the configuration of the simulation in the left and with the mapping of the agents to the objects in the right. 52

5.1 Trucks configurations for the experiments. 57

Abbreviations

BPM	Business Process Modelling: A structured representation that describes and defines the flow of activities in a particular business organisation
BPMN	Business Process Modelling Notation: A standardised modelling language for business process used to specify the work flow, roles and condition to activities in a set of graphic notation to create visual models of the process
CPS	Cyber-Physical Systems: An environment that consists in a computer-based controlled part that monitors and interacts with the real-world, physical system
ETA	Estimated time arrival: The expected arrival time that is related to a transfer
ETT	Estimate time travel: The expected travel time of a transfer. The departure time plus the estimated time travel gives the estimated time arrival
DMaaS	Data Mining as a Service: Software and computing infrastructure that allows interactive mining of scientific data in the cloud
IoT	Internet of Things: System of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction
MAS	Multi-agent system: A software system composed of multiple interacting (intelligent) software agents

Chapter 1

Introduction

The fast advances in e-commerce are supported by the growth in the global trade, but making the products go from one place to another requires big efforts in logistics: planning, scheduling and delivery are the most impacted areas. Although much attention has been put in delivery, companies have also been struggling with the problem of inventory management. It means transferring big lots of items from one warehouse to another. They must get there on time, but the information on the way (during the transport) is not always complete. Sometimes there is no information about the items until they arrive at the destination. Knowledge about delays and the capability to take corrective actions while a transfer is still ongoing could help warehouse managers to prevent delivery delays, saving money and time.

In this document, we discuss the state of the art in inventory transfers with carriers companies and how simulation can be useful for improving decisions. We intend to evaluate how agent-based simulation and data mining may help to reduce the delays in inventory transfers, focusing on the introduction of intelligence by gathering data from the environment and studying different configurations. We propose, develop and demonstrate the use of a flexible tool based on microservices to deal with challenges such as interoperability, scalability and reliability.

In this chapter, we introduce the methodological approach adopted, a brief contextualisation of the problem, the motivation and main goals. Finally, we describe the structure of this thesis.

1.1 Overview

Markets are continuously changing, and companies must always provide a fast and efficient response to disruption scenarios. In the context of logistics, the study of goods trade management until their arrival to the final client, schedule disruptions, as well as delays and misinformation are critical issues. If an item or a lot of items cannot be in place as scheduled, the whole logistic chain can break: the client does not receive the item in time, remediation actions may increase the costs and reduce profits, or even worse, the entire deal can be over. When talking about inventory management, each transfer may include hundreds or thousands of items, which represent a lot of money. Software plays a significant role in terms of developing new solutions to optimise the existing processes. The overall goals of the warehouse managers are: 1) reduce estimated time of arrival (ETA), and 2) minimise the number of times that an ETA is not fulfilled.

Amazon is an example of a company that takes these goals seriously. At Amazon, the goal is to deliver products to the clients as fast as possible, without promising an ETA that will not be accomplished and after that, reduce the costs that involve all of this. Forecasts based on historical data are made in order to predict what products they must buy and where products should be placed in their own warehouses as a way to reduce time response to new purchases. On the other hand, disruption management is still a challenge: Amazon and other e-commerce systems cannot guarantee the items are always there in time. In such cases, remediation is still a manual job, and despite all the data that can be collected, little exploitation of disruptions is currently being made during the transfers.

E-commerce companies usually rely on a hybrid transfer approach, using their own fleet of trucks or contracting third-party companies responsible for the transportation (which is a more common model). The study of this topic is called second-party logistics or *2PL*. It is described as a provider, an asset-based carrier that owns the means of transportation. The lack of information shared by the entities responsible for transportation is a drawback that opens a world of possibilities for improvement. Sometimes, transfers managers do not share all the information about an in-course transfer, either because they do not have the technology to collect and share such information, or by hiding as a business strategy. If a truck is stuck in a traffic jam, for example, the company may choose to not inform the seller that they will be delayed so they are not penalised. Other times, partial delays are compensated with faster sections. Damaged vehicles may need to be replaced and the items get a new carrier and this list of incidents goes on and on and on. It is clear that partners may not report possible bad cases, either to not impact future decisions by the client company or due to the massive volume of data and current ways of dealing with it, which makes it hard to track in a efficient way what is happening in the tranports. Also, there are problems more related to the operations side: delivery partners ca not assure their agreements (arrive on time, take the quantities necessary, etc.) or previous bad plans made that could lead to break constraints in warehouses, such as, dock constraints, critical departure times not covered and many others.

With the variety of cases and decisions that can be taken for each one of them, simulation is essential for two reasons: in order to validate possible solutions, before applying them in the real world, or generate data sets that are related to scenarios similar to the real world, so that we can use them in case no historical data was provided so far. Considering transfers between two warehouses, the more information we have about the whole path followed by the transporting unities (let's call them just trucks, to simplify, as this is the most used transportation unit worldwide), faster can be our remediation in case something goes wrong in a transfer. Thus, the best scenario occurs when we consider an internet of things approach [Trappey et al., 2017]. Internet of Things (IoT) devices can be separated in two different classes: **actuators** and **sensors**. Sensors listen and collect information from the environment and actuators are responsible for acting in the environment, as shown in Figure 1.1. Sensors are used to get a live track of the vehicles and the collected data during the process gives information about the sensible locations, where the delay occurs most, etc. On the other hand, we would end up having just the departure time and the arrival time, which cannot explain why and how the delays occurred. Even if there is no live track, companies can implement an info-point system that reports the truck location in some points of the route.

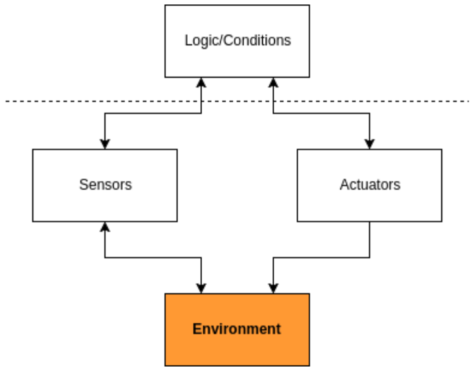

Figure 1.1: IoT device classes bridging the gap in cyber-physical systems.

Running simulations over this context and replicating the information sharing with more or less precision can help to create an information grid capable of transforming information into services, leading to the possibility of new decision-making. Artificial Intelligence techniques that allow data extrapolation such as Data Mining can provide predictions in terms of delays, probable faults and also improve trucks/company selection in terms of reliability. When it comes to cancelling or moving forward during transportation, autonomous decision-taking software, such as agent-based software could lead to better and faster decisions, reducing costs. Moreover, we move towards process automation, removing manual work and making it possible to scale to other dimensions without the need to increase the number of people responsible for managing these operations.

Multi-agent Systems are usually designed for that specific purpose: to model or simulate the behaviour of the entities the agents represent, the actors of the system. Each application domain has its own restrictions and requirements in terms of agent's capabilities and complexity. Hence, we introduce agents as an approach to distribute intelligence and be responsible for managing inventory transfers, solving run-time disruptions. Given that multi-agent systems and agent-based systems represent distributed entities that can take autonomous decisions and act to achieve their own goals, it appears to be a natural approach to introduce it in logistics scenarios with different application domains and not just for our specific case.

With the need to have a specific platform to simulate the different cases, we built a simulation platform specific for inventory transfers. With this, we separate the physical simulation of the transfers, such as the movement of the trucks, the constraints of the system and the devices that would collect the respective information, from the logic behind the interactions between entities and their actions by defining the physical part in the simulation platform and the logical role in the agent system. This separation allows creating different scenarios with different business logic, leading to the analysis of possible different cases.

The aim of this thesis is to introduce a novel approach for the development and study of solutions for current existent problems in 2PL inventory transfer, combining Multi-agent Systems and Data Mining, to orchestrate the environment the entities and interactions represented and learning from data collected during transfers, which in the real world comes from devices that form an IoT system. Following the recent trends in search for decentralised intelligence and distributed computation, which defines AI being a collaborative solution by a distributed group of intelligent agents, having has advantages minimal latency, efficient training, since we train for a specific group, and less power consumption once there are no dependency on network connection.

1.2 Problem contextualisation

Existing strategies to solve existent problems following distributed approaches are not suited to handle the increasing requirements regarding cost efficiency, flexibility, adaptability, stability and sustainability [Hofmann and Rüsch, 2017]. A lot of solutions studied, have software as the core and focus in the action of real-time interaction with the environment, in order to have control in what is happening in the processes. Taking decisions sooner then previously and also making available, tracking previous actions taken and correspondent reasons, which also opens a new door to use data to learn about the system and help in decision-making. In this sense, we focus on studying how the process of inventory management regarding 2PL logistics can be optimised using recent concepts in software, to bring solutions for use cases related to manual process reduction, processes automation and new ideas and ways of thinking to achieve solutions for similar problems. All of this, following areas such as microservices, to decouple actions and balance workload. IoT, to collect data from the environment, and AI to process the data collected and achieve knowledge from, in order to improve performance by covering topics, such as scalability and reliability. Singh mentioned that distributed computing is being adopted and a focus of research for multiple cases, however for settings where distributed computing is used, like edges devices and user control and privacy, did not have that much changes in their research to match these modern settings [Byers and Wetterwald, 2015]. In the following section we describe the problem for the different application domains we intend to explore and use as a environment.

1.2.1 Application domain

It is important to delineate the aspects on application scope to consider the proposed study as a feasible possibility. In this sense, we highlight our preliminary thoughts regarding the needed characteristics and possible application scenarios for experiments with our approach.

Focusing on 2PL inventory management problems, we intend to study the impact of our approach to critical factors for the system, which we point it out as the scaling part, coordination and flexibility. We have worked in a simulation of warehouse transfers environment, represented by a main company which requests third-parties to perform the transfers, in order to see the improvement it can take by exploring the integration of agents and devices in the processes, having the ability to control and track the information of each entity to cooperate in order to achieve the best decisions for the company. With some experience in this area, some problems were detected when in terms of managing the transfers that were previously placed, either because of the error associated to the prediction or real-time issues that need to be fixed.

The warehouse transfers process has the aim of distributing the products between warehouses to comply with the delivery in the estimated date given to the clients when they order a product. It consists in doing a forecast for a period of time with the number of trucks needed to transport this products between warehouses. After doing this there is a team responsible for maintaining the scheduling made and fixing problems that can be related to multiple factors, either the forecast was not accurate leading to schedule more or fewer trucks that were needed, or the carriers responsible for the transport can't do it at the time that was defined, etc. Being this maintained manually, it gets difficult to track the trucks and their status. We want to apply the MAS model in a simulated scenario that represents warehouse transfers, to study how can be improved with the usage of agents to perform current manual actions and also control the devices, that can be the trucks and warehouses. The schedule of vehicles from one to another has a lot of constraints that can break, having the need to fix them taking decisions about what actions should be applied.

1.3 Motivation and Goals

From the rapid and massive growth of the scale in logistics environments, manual work is something that cannot be though in a long term perspective or will have the need of having large teams for managing repetitive tasks. Also, with new times, new demands and the requirements from customers are even more strict, either in terms of pricing or delivery time.

In our approach for 2PL inventory management, we focus on the development of a new simulation tool, so we can provide the ability to scale to levels that are similar to the ones that exist in reality and focusing in having a general representation of how this process works but with variable configurations of what can change from company to company, in order to enable the usage for different cases inside of this environment. This platform tends to represent the need to observe and actuate in the real world, not just on the sites but also during the transfers that occur, which simulates the connected devices as a way of communication. The main objective of this work can be divided into two sub-goals. From the application side we want to study how usage of devices to access information during transfers can provide reliable solutions for scheduling disruptions in logistic problems and from the theoretical perspective, we want to analyse different agent configurations in terms of the shared capabilities between devices and agents in order to evaluate which capabilities should be in the agents and which should be in the devices.

The relevance of this kind of cyber-physical system is also evidenced by the dynamic changes in both business processes and personal tasks under the concepts of Industry 4.0 [Jazdi, 2014]. They suggest that the introduction of sensing devices can provide a more accurate, big set of data that can be used to improve current processes, train machine learning models to decision-support or even provide near real-time decisions. One can use this increasing volume of data to leverage autonomous processes and make it more dynamic, more efficient, providing problem-solving software that can outperform current systems where IoT is not present.

Once we have our simulation tool, we need now something that will be responsible for representing entities and model behaviours of what happens during a simulated environment. Since we want to achieve autonomy, agent-based software appears to comply with the job, being able to act as humans, being responsible for decisions along with the simulation. A problem that we face when using agents is to grant that we minimise the errors made by them, especially with devices that have high responsibilities. Also, while a Multi-agent System can provide the coordination mechanism for each device or group of devices, scalability issues arise in the current agent framework, and our work towards more dynamic and scalable agents take us towards a microservices approach. From a technological perspective, microservices have been gaining attention in the last few years because of their availability and their scalability capabilities regarding the decentralised computation, which could leverage the overall Multi-agent System possibilities by introducing agents capabilities as services. As such, agents with the same reasoning process could have different capabilities just by changing the services they call as data-mining, monitoring, external computation services, human interface services, etc. We intend to explore this configuration to get a loosely coupled organisation of collaborating services. This would be helpful to detach the agent workload into smaller software and to help to create a more dynamic population of agents with different capabilities. Thus, our insights about the possibility of modelling agents processes with a flexible modelling paradigm, such as Agent Process Modelling highlight that new studies combining are much needed and could be very useful in industry every-day problems.

Although our study is based on simulation, it is related to real cases, and everything is taken in consideration so it will be possible to apply and be a applicable in real cases. Even though we don't involve physical devices, nothing that is simulated would be possible if in a real case IoT is not introduced. It is what will allow collecting data that has the purposes to be used for AI mechanisms, such as, Data Mining, in order to improve our agents decisions and achieve a reliable method of automating the inventory management.

The specific goals of this work are:

- **G1:** Study the inventory transfers problem from a distributed perspective
- **G2:** Develop a distributed simulation tool based on microservices
- **G3:** Analyse and model behaviours of entities with agent process models
- **G4:** Analyse how the performance of a third-party can be used to evaluate its quality when assigning loads
- **G5:** Analyse how to use information collected during transfers to act proactively seeking to reduce problems (delays)

With this, we intend to contribute for the introduction of new ways to model agents behaviour represented as processes, a distributed simulation system for the specific case of inventory management and respective transfers and finally the usage of AI with Data Mining for our analysis of the data collected to impact actions to be taken during processes.

1.4 Research methodology and questions

This thesis follows a *problem-oriented* approach to solve the identified problems. Although an application domain is chosen to validate our solutions, the core concepts and models are expected to be sufficiently generic to be used in the hole class of related problems that could benefit from our contribution.

We intend to develop and apply new models for real-world scenarios that use IoT. To achieve our solutions, we have adopted the following research steps:

1. **Problem identification:** we have identified problems related to logistics, giving the approach to be taken and making the bridge to specific problems in the scenario of 2PL inventory management, as described in Section 1.2.

2. **Deep study of the problem (gap analysis):** our systematic literature review should provide solid information about the problems to characterise the key points where our new models will contribute.

3. **Hypothesis formulation and goals:** within the deep study of the problems and possible solutions we could assert some hypotheses that will allow proposing our new models to solve the identified problems.

4. **Modelling, simulation and experiments:** we build a simulation tool focused on inventory transfers to validate and verify the usage of agents and Data Mining in different application scenarios.

5. **Analysis and evaluation:** After validating the newly proposed mechanisms solve the problems we will perform a critical analysis to conclude if our hypothesis were confirmed or wrong.

1.4.1 Hypothesis and Research Questions

In our study, we can separate the simulations to be made in two types: one we will let entities have the decisions be taken based on constraints given in the beginning and then we will use the data collected on the first simulation, to learn and improve decisions based on Data Mining. We will follow a step by step development by experimenting with a more simple case like the first type and then with a more similar to the second type, but the aim is to finish with a Multi-agent System that will be responsible for all processes, from choosing the partner to realise a transfer, to take the best action in situations during a transportation and be responsible for the all process of management between the company that owns the inventory and the third-parties. Therefore, the hypotheses formulated are:

Hypothesis 1

Simulation can help to evaluate and select appropriate policies to choose delivery partners

Hypothesis 2

Data Mining can be used to point out disruptions during a transfer

The following research questions are considered:

- **RQ1:** How much advantage can we get from tracking transfers during the process?
- **RQ2:** What is the impact of intelligent agents in the automation of warehouse transfers?

Finally, one intrinsic research goal/question in this work regards the study of the suitability of the Agent Process Model approach as a flexible way to define and coordinate agent behaviours, synchronising agent capabilities with microservices. Throughout this thesis, we will seek to answer these questions and others related with the development of the models and services. From a practical perspective, the results of this work include a simulation tool, where entities of simulation are represented by agents, having the goals of solving 2PL inventory management problems with an agent-based approach.

1.5 Document Organisation

The present document is organised in four chapters. In the first chapter, consisting in this first chapter (Chapter 1), we introduce and discuss the problems and motivations that will be addressed as well as our research goals. In the second chapter, we describe our problem and explain concepts used in our research (Chapter 2). The third chapter contains the State of The Art, presented to better explain the contextualisation of the investigation, the limitations found in related works and open fields that could guide our work (Chapter 3). A fourth chapter (Chapter 4) describe the preliminary advances towards Intelligence on 2PL inventory management, details about the environment to be used and some technical characteristics about the software development that was made. The fifth chapter contains the details of experiments made and respective results (Chapter 5). Finally, a chapter with conclusions (Chapter 6) highlight the importance of this research and guide the readers for future research steps. In the following list is a detailed structure analysis:

- *Chapter 1 - Introduction*, presents the contextualisation, motivation, hypothesis and research questions of this thesis, as well the main goals and contributions;
- *Chapter 2 - Inventory Transfers Management Problems*, describes the different scopes of inventory transfers, existent problems and the description of the concepts used along with the research;

- *Chapter 3 - State of the Art Review*, gives an overview about the methodological approach for reviewing the literature, describing new trends in logistics inventory transfers environments, distributed intelligence and its relation to agents, Data Mining cases applied in logistics scenarios and the usage of business process modelling to management in logistics;
- *Chapter 4 - An Intelligence of Things Approach*, describes the simulation tool as well the specific environment created, the preliminary architectural aspects of having in consideration for the development to be made, the models that are related to each agent represented and finally the Data Mining approach to achieve better results;
- *Chapter 5 - Experimental Evaluation*, describes the simulations and their configurations, explaining all the steps made as well metrics used for evaluation;
- *Chapter 6 - Conclusions, and Future work*. Finally, in this chapter, we provide a discussion about the literature review and the research questions and hypothesis proposed, explaining the contributions that are expected and future work.

Chapter 2

Inventory Transfers Management

The research in inventory management can have different targets since it is a broad context that involves many steps and phases. It can be related to a warehouse management with the focus of improving operations inside of a warehouse, such as, the definition of patterns for a warehouse structure, the management of units and their arrangement in the warehouse and more other cases, or it can be about the management of transfers between warehouses, either to improve predictions of transports that are going to be needed or improve the actions taken during the transports management. There are different scopes of dealing with all the operations that inventory Supply Chain involves, companies can be responsible for managing all the process, allow providers to handle some of the procedures or hand all the management. Accordingly to the participation level in the whole logistics process, different names can be defined for such logistics management:

- 1PL: Represented as a first-party logistics provider is a firm or an individual that needs to have cargo, freight, goods, produce or merchandise transported from a point A to a point B. The term first-party logistics provider stands both for the cargo sender and for the cargo receiver;
- 2PL: A second-party logistics provider is an asset-based carrier, which owns the means of transportation. Typical 2PLs would be shipping lines which own, lease or charter their ships;
- 3PL: third-party logistics provider provides outsourced or 'third party' logistics services to companies for part or sometimes all of their supply chain management functions;
- 4PL: A fourth-party logistics provider is an independent, singularly accountable, non-asset based integrator who will assemble the resources, capabilities and technology of its organisation and other organisations, including 3PLs, to design, build and run comprehensive supply chain solutions for clients;
- 5PL: A fifth party logistics provider will aggregate the demands of the 3PL and others into bulk volume for negotiating more favourable rates with airlines and shipping companies.

This thesis study the specific case of 2PL, where company A requests company B to realise transfers with its goods and between its sites, and company B is responsible for the trucks and routes took during the trip. In this specific scenario, company A manages the warehouses and plans the transfers that are needed, according to the demand from clients and distribution of loads from suppliers, and requests third-parties (partners) to then realise them. Our research is towards finding a way to automate the process of management in transfer and base the automation in the improvement of two cases: **improve decision-making when selecting partners** and **predict the delay of transports**.

2.1 Warehouse Transfers and Agents

Transfers between warehouses are represented as the transports that occur, where a truck leaves from the origin and arrives at the destination. For our study, they have the same meaning, but with an additional feature, when a truck leaves the origin, go through intermediate points (which we call, info points) that are defined along the route, and then it arrives at the destination. The info points are established, in order to track the trucks driving, since they have to report information every time they arrive to a info point. Also, they are identified as a flexible and simple way to define decision points during the course. Trucks are autonomous in this points, having the ability to report or not, the information about its status in the info point. Figure 2.1 shows a representation of the related entities the rectangles W1 and W2 represent the warehouses, whereas the small squares stand for the intermediary points where the trucks must pass to inform their location and time. Each warehouse knows the transfers that are going to occur from it, since they are previously planned, and for each shipment, the warehouse is responsible for interacting with carriers (partners) to realise the transports. In our research, we consider the interaction at truck level, where warehouses communicate directly with the trucks (which are identified with the company they belong).

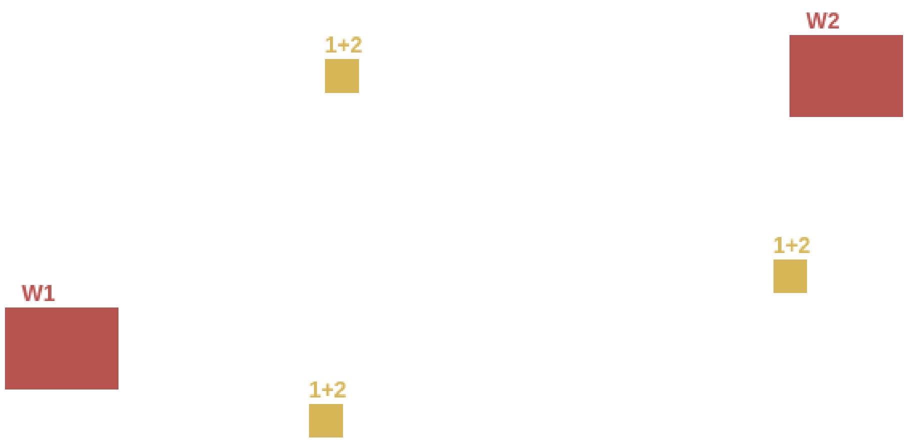

Figure 2.1: Warehouse transfers representation with info points defined.

In a real scenario, the whole process of picking partners to carry specific shipments and tracking ongoing transfers is usually made by manual operations, which can be a burdensome work when the network and the number of shipments increase considerably. It might not be possible to scale in an efficient pace. Some strategies for automation in this scenario have already being studied [Xiong et al., 2015, Jayaram, 2017] and our proposal is to use an agent-based model, since the distributed and complex nature of the interactions make agent based systems a suitable framework. More specifically, our idea is to use agents are modelled as processes, using Business Process Modelling (BPM). Given that the current methods are manual and based in human decisions, agent-based decisions could help us leveraging such decisions by automating the process, replacing some actions that do not require manual help and mapping agents to represent the respective entities needed, in our case, the **warehouses** and the **trucks**. Agent Process Modelling corresponds to the methodology in which agent behaviours are modelled through BPM and capabilities are distributed along with other services. In this context, agents run processes that orchestrate their behaviours, distributing the workload, improving the scalability and reliability of the autonomous entities.

2.2 Business Process Modelling

Inventory management can be described as a process that requires a lot of validation and a strict road map with certain norms to be followed according to the events that are triggered. Our literature review highlighted some relations between logistics and Business Process Modelling, since a lot of companies in this area are used to have BPM as a way to define internal processes [Grzybowska and Kovács, 2017]. Business Processes [Jeston, 2014] can be defined as a logical sequence where input is computed to produce some good or service, or even to deliver a decision action [Chiarini, 2012]. This [Scholz-Reiter and Stickel, 2012] allows to centre the focus on the functional aspects of the system, while agents must deal with how to accomplish such activities. BPM can be used to model agent behaviours flow, representing their states and the interactions between the multiple agents in their environment. It allows to abstract capabilities having them defined and distributed along with other services which make the model mainly responsible for the coordination of each agent and interactions, removing computation from the agent itself.

Among the business process modelling methodologies, we highlight the Business Process Modelling Notation (BPMN), a standard notation that makes it easy for users, managers and designers to understand the process by simple symbols [Chinosi and Trombetta, 2012] and unify process modelling knowledge and representation and implementation. BPMN links the process design with the semantics of the process, allowing the integration with software tools that interpret and create process execution instances. BPMN diagrams do not comprise all the information needed to run a business process, but instead, attributes and rules support the additional information required to describe roles, names and conditions. The four basic categories of graphical BPMN elements are shown in Figure 2.2 and comprise: flow elements as events, activities and gateways; connectors as message and sequence flows; swimlanes and artefacts. As we can see, BPMN is a very robust notation and can describe the interaction flow between process actors [White, 2004].

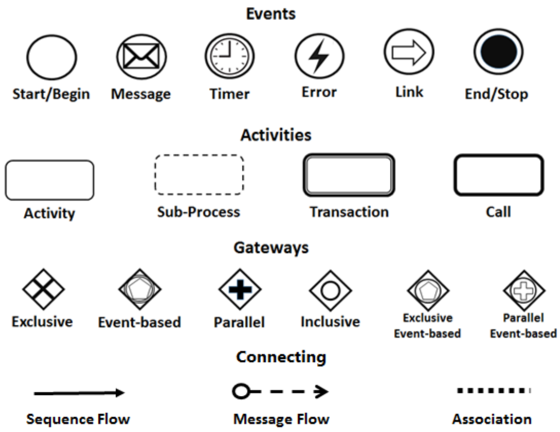

Figure 2.2: Elements and respective groups present in business process modelling notation.

Although the whole set of bpmn elements is big and complete, in our processes we used some of them more frequently. Given the four basic categories of graphical BPMN elements, we identified the more relevant ones for our processes, shown in Figure 2.3. Their definitions are as follow:

- **Events**: an Event is represented by a circle and is something that "happens" during the execution of a business process. These Events affect the flow of the process and usually have a cause (trigger) or an impact (result);
  - **Start Event**: event that marks the start of the process;
  - **End Event**: event that marks the end of the process;
  - **Catch Message Event**: event used to wait for a message;
  - **Throw Message Event**: event used to send messages.
- **Activities**: an Activity is represented by a rounded-corner rectangle (see the figure to the right) and is a generic term for work that the company performs. An Activity can be atomic or nonatomic (compound). The types of Activities are Task and Sub-Process;
  - **Service Task**: task used in a process to call services. It has *url*, *method*, *payload* (optional) and *headers* (optional), as input. Receives and stores the result in a *response* output variable;
  - **Script Task**: task used to run code inside of the process in order to perform some computation. Allows to write scripts in groovy.
- **Flows**: an flow is used to connect the elements in order to define and represent the skeletal structure of a process;

  - **Sequence Flow**: represented by a solid line with a solid arrowhead (see the figure to the right) and is used to show the order (the sequence) that activities will be performed in a Process.

- **Gateways**: a Gateway is represented by the familiar diamond shape (see the figure to the right) and is used to control the divergence and convergence of Sequence Flow. Thus, it will determine traditional decisions, as well as the forking, merging, and joining of paths. Internal Markers will indicate the type of behaviour control.

  - **Exclusive Gateway**: allows defining the course of the process by applying a Boolean analysis. If the flow with the analysis is true that the course of that is followed, otherwise it follows the default flow;

  - **Event based Gateway**: used to listen to more than one event at the same time. Follows the course of the event that is triggered first.

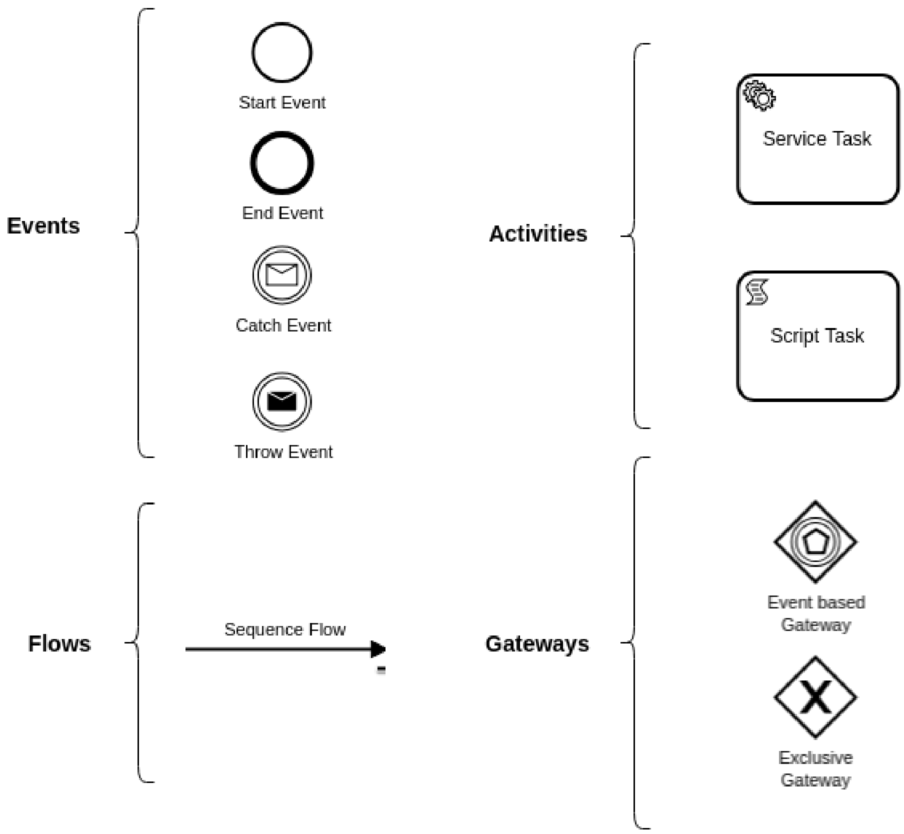

Figure 2.3: BPMN elements used with the respective names and groups.

There is another essential element due to its impact on the transition from manual to automation: the **User Task**. Although we do not use it in our processes in this thesis, it is important to refer that represents an activity made by human interaction to do something so an agent modelled by a process will continue its flow. During the transition, so don't take risks of the automation makes wrong decisions that might have a negative impact for the company, we can still have human interaction in these critical points, for instance, to validate or not the decision that was taken by the agent.

2.3 Warehouse Transfers Research and Evaluation

Since warehouse transfers represent physical actions that have a high price in terms of resources used, we focus our research in the simulation of these in a simulation tool, built by us, focusing on simulating transfers between warehouses and existence of the two entities we referred, warehouses and trucks. With the simulation, we can run multiple times the same scenarios, generating the amount needed of data to achieve some knowledge about the environment since the idea is to learn from that data (through Data Mining) and add a new capability to the warehouses, which is to decide better the trucks that perform the transfers and also to know beforehand that a carrier will arrive late. Since our focus will not be what is the best Data Mining algorithm to obtain the best results in terms of decisions, we intend to use an AutoML platform to simplify our studies and let us focus more on the automation side.

It is also good to mention that the application of our studies is only valid in the real scenarios when tracking devices are introduced to the system. What we simulate as the capabilities of tracking the movement and information of the trucks related to the transport, is only possible when physical devices are present and collect data of the environment (time, distance, location, etc.). There are multiple devices available to realise some of these processes, such as, RFID which uses electromagnetic fields to identify and track tags attached to objects automatically. The tags contain electronically stored information, sensors that collect data from the system (temperature, movement, etc.) and GPS a global navigation satellite system that provides geolocation and time information to a GPS receiver. With the improvement of these devices and also new inventions, the integration and possible application in complex cases become more accessible and more affordable.

To evaluate the results, we focus on two metrics that have a direct relation with inventory transfers management, estimated time arrival (ETA) and delay. When a warehouse has a list of schedules (transfer to realise), each one has a departure time and an estimated time arrival, which is calculated previously when the transport was planned. We evaluate then the delay according to the estimation that was made and the time that was made to realise the transfer. In this way, we analyse and score the truck's performance according to the times they took to do all the route (from origin to destination) and also to do the transport between the info points. We don't focus, for now, on the trucks price, giving priority to the delivery performance only.

2.4 Summary

Warehouse transfers is a process inside of inventory management, that is responsible for the transports of units between warehouses and its respective management, which involves the transfers planning that is needed and the tracking during them. We described what transfers between warehouses is considered to our approach and how we intend to automate and improve the processes involved. Following our strategy, we explained essential parts for the understatement of the following chapters, by giving details about technical themes involved but not studied in our research. The next chapter reviews the current state of the art of the technologies and scopes that are related to our study and context.

Chapter 3

State-of-the-Art Review

This chapter presents a curated analysis of the related work, reviewing the State of the Art associated with the main research areas of this project. More than a conceptual overview, the literature review aims to identify the gaps and possible research lines regarding agent-based simulations and distributed intelligence regarding the warehouse transfers domains.

3.1 Literature Review Methodology

Reviewing literature consists in finding the most relevant, state of the art scientific works in a study field. In this thesis we adopted the way of collecting keywords that are related to the theme, apply multiple searches and collect the information that made more sense to focus as a relation to our problem. The reviewing process only considered publications in English that are present in the following search engines: Scopus, Springer Link, IEEE Xplore and Google Scholar. In the table 3.1 we can find the keywords that were used in our queries, separated by their main relation.

Field	Keywords
Multi-Agent System	agents multi-agent agent-based framework
Internet of Things	internet of things IoT simulation
Logistics	logistics inventory management warehouse transfers 2PL
AI and Logistics	data mining in logistics data mining as a service

Table 3.1: Literature keywords

3.2 New challenges in Supply Chain

The need of a more efficient and accurate delivery of inventory demand is growing. Where previously, wait a couple of days for a product bought was considered amazing, nowadays companies have the need to reduce this waiting times and at the same time improve their costs and success rate of delivery. To achieve these goals, companies have to face multiple challenges that requires changes to existing operating practices [McFarlane et al., 2016]. Some of these challenges are:

1. Business efficiency: Target the focus on "leaner" industrial supply chains to reduce costs and waste and improve delivery performance. Main effects are to reduce inventory being held and to shift towards Just In Time (JIT) delivery models [Naim and Gosling, 2011].

2. Uncertain operating environment: Congested transportation routes and complex access leading to uncertainties in collection and delivery times [Sanchez-Rodrigues et al., 2010].

3. Needs of individual customer: Customers are also demanding the option to change orders after submission, both in terms of order composition and delivery options [Sik Jeong and Hong, 2007].

4. Market variability and changes: Rise of internet shopping and with it direct purchase from supplier warehouses, increased ranges on offer, vastly increased numbers of small orders, demanding delivery times with guarantees [Davarzani and Norrman, 2015].

Recent advances in computing and communication have been studied as a solution for several of these problems that companies are facing. The need to predict future transfers to be made and how products should be distributed along multiple sites is crucial, in order to improve metrics, putting the companies in front of the rest of the competition. For this, companies must be [McFarlane et al., 2016]:

- Cost-effective: able to provide services with a competitive price level.

- Robust/resilient: able to absorb disruptions such as delivery delays, employee absences, incorrect order preparation and others.
- Customer-oriented: anticipate the changing customer base and the needs of the customer in order to respond effectively.
- Adaptable: able to change logistics priorities rapidly without due cost increases.

To evaluate results, there are specific metrics that are used in inventory management, such as, order accuracy that corresponds to the amount of orders that are shipped, processed and delivered without any incidents on its way. Delivery time that measures the time that units take from the moment they are shipped and the moment it is delivered to the customer. Transportation costs that is the value that costed to process the order from the begin to the end. Inventory accuracy which is related to avoid problems about bad inventory management, by having units in the warehouse closer to the clients, without running out of stock or exceed the need of it, etc.

Since the transport process in Supply Chain has a lot of constraints and requires a big investment in resources, most e-commerce companies, like Amazon, Ebay, Aliexpress, etc., request specialised delivery companies to be responsible for this process since they have the resources to do the transports needed across different locations. Having partners to realise this process, the companies as clients have the responsibility to plan the transports following their needs and predictions, being then agnostic to how the transfer will be made which leads to a lack of information from when the product leaves the warehouse until it arrives to the destination. The collection of data of what is happening in run-time in the warehouses is an approach taken in order to improve the management side and also to reduce manual effort needed by processes automation [Harukawa et al., 2001]. Although, the management side of what happens during transfers has a gap of missing information that is shared between the partner and the company, which makes it difficult to automate the side of handling problems during the process and also base the election of certain companies/trucks according to their success rate of comply with the promises that were made before starting the transport. Until now we couldn't find any approaches that attack this disruptions considering the data that might be possible to collect during transfers. There are methods studied where the foundation is only the information that is kept from the initial and final point, such as, the departure and arrival time, the respective ETA that was estimated, the company name and the truck type. This allows to possibly improve the decisions of picking a certain carrier or truck, but doesn't close the gap of predict delay of transfers during the process.

3.3 Multi-Agent Systems

The growth in computational power lead to decentralised architectures with the development of distributed systems. In this context, Multi-Agent Systems have been proposed as a distributed artificial intelligence approach in order to solve complex problems where the entities are independent and autonomous, designated as agents. An agent-based system comprises the perspective of each agent by defining his actions, interactions and behaviours in order to achieve its goals.

Some characteristics of the Multi-Agent System (MAS) [Pavón et al., 2006, Bordini et al., 2009] such as coordination, autonomy and decentralisation allows to model agent according to in order to represent entities, roles and decisions. MAS has also been gaining crescent attention in the last years since it represent a natural evolution of the object-oriented programming [Bresciani et al., 2004], offering a new way to develop distributed systems. MAS technology can be considered as an adequate way the address problems that are better modelled using a distributed approach.

The basic entity for the MAS paradigm is the agent. The agent does not have a clear consensus of what it represent, it can be defined as intelligent agent or autonomous agent. Given this, we consider the definition that is widely adopted in literature, in our case is the one from Jennings and Wooldridge [Wooldridge, 2009]:

> "An agent is an encapsulated computer system that is situated in some environment that is capable to act flexibly and autonomously in this environment in order to achieve their design goals."

Given this, we can look to an agent as software in a computing device that has the capabilities of sensing and actuation, allowing to perceive the environment and act on it according to its goals. It can have either reactive or proactive processes, in the sense that he acts according to changes in the environment but also act to achieve its goals. Thus, we can say that the agent is autonomous and goal-driven, since he has the autonomy to make decisions and control its behaviour independently to other systems intervention.

In Figure 3.1, we can see a representation of the sensors and actuators that a agent has as well the way they interact with the environment. The perceptions are the information that is received by the agent which acts according to [Wooldridge, 2009].

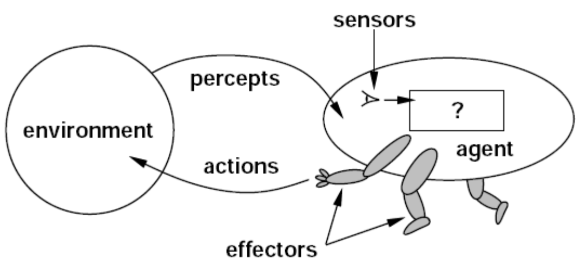

Figure 3.1: Agent representation (Adapted from [Russell et al., 1995])

Given the autonomy of the agents, once they are in charge of decisions it is not possible to control their inner motivations. The motivations are related to their goals. This goals in a MAS can be described as system goals that are related to collective goals or agent goals that are individual goals.

3.3.1 Interaction in Agents

A Multi-agent system can be characterised in three main topics: agent domain, environment domain and interactions. The agents that are in a system do not have a complete knowledge of the environment, however they observe with their sensors, plan to achieve goals and act. They can interact with others through communication mechanisms to request or coordiation the execution of some tasks. There are two considerable agent communication languages (ACL) that are often used: KQML (Knowledge Query and Manipulation Language) and FIPA-ACL (FIPA - Agent Communication Language). The latter was proposed by the Foundation for Intelligent Physical Agents (FIPA) as a standard language of agent protocols for negotiation, cooperation and competition. In Figure 3.2 that is the structure of an ACL message with the performative, content and parameters.

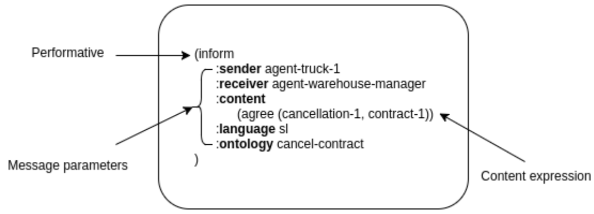

Figure 3.2: FIPA-ACL message structure.

3.3.2 Agents Frameworks and Platforms

Given the growth of agent-based models and multi-agent systems paradigm in the last decades, a lot software agent platforms have been proposed to address simulation and deployment. Surveys about Agent-Based Modelling (ABM) toolkits describe the most used platforms and their limitations [Railsback et al., 2006]. The purpose of ABM frameworks is to enable developers to work deeper on the logical aspects and modelling properties of their studied cases rather then to spend efforts on the basic internal infrastructure of the multi-agent system. In this sense, the frameworks present common features for communication, execution and addition of new agents.

Different frameworks are built for different purposes, some are simulated oriented, others are more generic or specific for deployment. To compare the toolkits we can take in consideration features such as scalability, learnability, simplicity and programming language. In the following list we can see the frameworks that were found in the literature:

- **AnyLogic** [Borshchev and Filippov, 2004]: simulation platform with a graphical interface for a discrete event modelling

- **Cougaar** [Helsinger et al., 2004]: based in a Cognitive Agent Architecture for logistics problems

- **JADE**: follows FIPA specifications supporting distributed environment using Java. Currently is the most popular framework agent platform.

- **Jadex BDI Agent System** [Pokahr et al., 2013]: follows the Belief Desire Intention paradigm, supports simulation, scheduling and mobile computing using active components.

- **Jason** [Bordini et al., 2006]: BDI agent-oriented logic programming language that implements the semantics and support the development of multi-agent system.
- **NetLogo** [Tisue and Wilensky, 2004]: uses the Logo language to enable exploration of emergent phenomena by providing MAS simulation that enables participatory and social simulation, making it very popular in the academic world.
- **MASON** [Luke et al., 2005]: is a simulation agent platform to serve as the basis for a wide range of multi-agent simulation tasks allowing data visualisation and processing.
- **RepastSuite** [Bellifemine et al., 2007]: open source agent-based modelling and simulation toolkit with multiple implementations and adaptive features widely used.

Besides these more generic agent based frameworks, there are also some that were built specifically for logistics domains. In the following list we can see other frameworks with a logistic domain that were found in the literature:

- **MISIA** [García et al., 2011]: allows simulation, visualisation and analysis of agents behaviours, which fits well for logistic domains. It makes use of existent technologies, JADE and Repast, for the development of multiagent systems known and widely used, and combines them so that it is possible to use their capabilities to build highly complex and dynamic systems.
- **PlaSMA** [Gehrke et al., 2007]: is a event-driven simulation system which has been designated to solve and evaluate scenarios of the logistics domain. Allows to simulate autonomous logistic processes where autonomous agents perform planning and decisions processes.

Literature analysis indicates multiple gaps when considering scalable agent systems, flexibility regarding agent modelling and possibility to integrate agents with services. Our preliminary analysis points towards using Agent Process Modelling, a novel approach that uses Business Process Models to describe agents' behaviours as a good way to model and orchestrate our agents and services. We should also need to validate the impact of this approach in the multi-agent system we intend to reproduce. One of the key aspects of involving such a big confluence of technologies should be the possibilities of integrating simulated IoT devices and real ones in order to evaluate the performance of the system under different configurations, such as improve agents knowledge agents to take decisions under a set of simulated conditions and then verity the solution in a real scenario, coordinate mixed groups of devices (real and simulated) in a seamless way, etc.

3.3.3 Agents and Data Mining

Decision support systems have become an indispensable tool for managing complex supply chains. Our research point three fundamental dimensions of supply chain agility - responsiveness, flexibility and speed. With the need to achieve these, the integration of agents to automate the processes and also to be responsible for the handling of data that is collected from the environment is a feasible approach, since we can map agents to entities that are defined. Having the possibility to collect data, the incorporation with big data has become a research area in order to build new decision support systems (DSS) [Giannakis and Louis, 2016]. The integration between agents and data mining is referred as agent mining and is driven by challenges faced in both areas, and the need of developing more advanced intelligence, information processing and systems. With the ability of agents to automate processes and with the knowledge discovery in domain-oriented decision systems through data mining, the usage of sensing and acting in real environments and with the knowledge that can be achieved, this combination has been a case of study for different applications since the concepts are general and what might change are the processes and the data that can be collected [Cao, 2009]. Figure 3.3 shows a possible interaction between agents and data mining system.

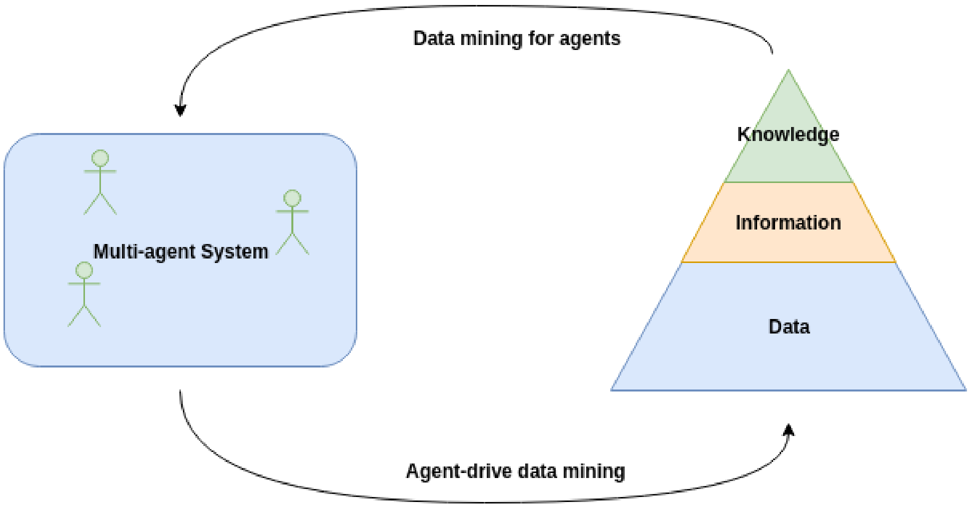

Figure 3.3: Agent mining interactions.

3.4 Multi-agent system applied in logistics

The usage of Multi-agent Systems in a real environment started being a solution to automate processes in industries that have logistics involved. As a way of allowing agents to interact with the environment, companies use devices with the capacity to perform human tasks improving productivity, security and speed. This also brings new possibilities to implement new processes that were not consider before [Broy and Schmidt, 2014].

Internet of Things is known as the capability of devices that interact with each other and transform information from the real world into services that are accessible from anywhere, in order to compute something and take corresponding measures. Being Cyber-Physical Systems linked to Internet of Things, the growth was taken also in this context and impact multiple domains, innovating processes in automation system at their networking and access to cyber world, being necessary to redefine industries, society, and everything that interacts with it [Jazdi, 2014]. The integration of

IoT and CPS into logistics promise to enable real-time tracking of flows [Hofmann and Rüsch, 2017]. Given this, researches wre made to find solutions by redefining how the interoperability is established among various IoT entities, proposing new architectures that are oriented to distributed computing. This pretends to solve cases of large scale applications, being a need that is increasingly since the scenarios of application are becoming bigger [Sarkar et al., 2014].

The distribution of processes with agents allows to scale enough for complex environments where the centralised approach is not efficient to handle the volume of information and decisions that need to be taken. If we gather data with the right distribution this will improve as well the performance since network resource preservation will be achieved by reducing flows between devices which can impact the network bandwidth. Having Distributed processing the goal is to only send the needed information to the cloud and also the data process is made in remote site being much closer to data sources reducing the delay between data gather and data process. Another fact is clustering that makes us able to define interactions between individual devices or clusters, where a group of devices (cluster) is seen as a single entity [Byers and Wetterwald, 2015]. Figure 3.4 represents a example of distributed intelligence with the properties described.

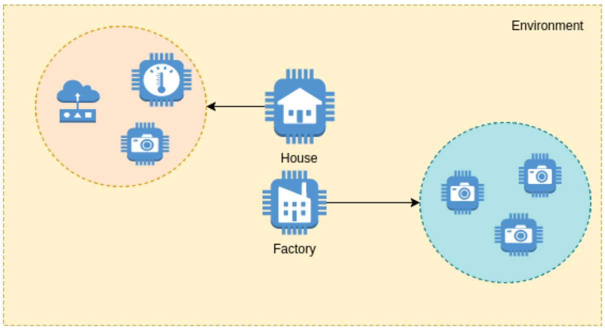

Figure 3.4: Clusters of devices seen as an entity.

Having software to track the processes and collect data in a logistics environment was already achieved, although taking decisions according to that data, removing the challenges referred previously, does not has a solid definition. The introduction of MAS has the impact of improve the way devices work and interact with the environment. There are a considerable amount of research about the usage of agents with devices, from applications already created to protocols of communication that can be implemented, but a problem with this ones founded is that they are based in a central controller, having a single knowledge base. This leads to the solutions be disabled to dynamic cooperation between devices, which only allows to be used in fixed scenarios. Thus, it arises the need of studying new ways of decouple this central controller, being agents an approach to merge in the protocols and architectures already used in order to decentralise the knowledge [Kato et al., 2015].

With research made, about the application of Multi-agent Systems in real environments of logistics, we could see that there is not much usage currently. Given the current frameworks and platforms that are available, these are good for a scientific study and usage, although, they are don't perform so well when applied in real cases. Most of the frameworks and platforms provided, tend to be difficult to use when technical requirements as scalability are a must in the application to be made. Also, simulation platforms provided are normally, specific for certain cases and don't provide the flexibility needed, in order to apply it in different scenarios.

3.5 Microservices

There are multiple methods related to the architecture on how we compute tasks, from the initial implementation where a computer could only do one task at a time, to multitask operations running them in the same computer. Due to new needs and market changes new ways of thinking in how to architecture systems started to being consider. Recently, Microservice Architectures started to be seen as a new way of building all types of services and applications [Thönes, 2015]. It breaks the typical monolithic approach to put all the functionalities into a single process, where in the need of scaling the solution was replicate the monolith to more servers, to decomposing in elements of functionality and separate them in different services. In this way, we can scale by distributing the services across servers as needed. The usage of this architecture applied with a Multi-Agent System can bring the scale to our implementation, by distributing each agent or a group of agents per service, where we can distribute the workload by having multiple services with the same agent defined and the service itself, takes care of the communication between other services. For communication between services, the normal usage with a Microservices Architecture is a publish and subscribe method which allows asynchronous messages, where the sender publishes messages to a topic and it is immediately received by the subscribers to the topic. Another possibility is to allow services to communicate throw API and each one works as a web service [Dragoni et al., 2017].

3.6 Summary

In this chapter we discussed the State-of-the-Art related to logistics, focusing in supply chain and 2PL inventory management. We studied BPM and its relation with companies that have logistics processes. After the research related to the domain, we focused in the usage of Multi-agent Systems nowadays and current frameworks available and the applications in real scenarios with a brief research of the interaction between agents and devices. We analysed the different approaches taken so far regarding how the distribution of intelligence has been made and how this intelligence can be achieved with Data Mining. Moreover, agent-based approaches applied in logistics environments were studied in order to understand capabilities that multi-agent systems can bring to this specific domain and the existent problems with the introduction. In the end we made some research about microservices and the pros and cons of following this architecture pattern. Our insights about the literature review points to new research opportunities to help systems become more autonomous and take more advantage of the information that is sensed from the environment.

Chapter 4

An Intelligent 2PL Inventory Management Approach

We have reviewed the related literature in logistics problems and the different approaches in this area, going into more details about inventory management and possible approaches to automate processes through AI. In this chapter, we introduce our solution methodological approach and development a more detailed description about the simulation platform as well as the specific scenario of the 2PL inventory management.

4.1 Introduction

Our efforts are towards finding a possible solution to automate 2PL inventory management having agents and data mining as resource to achieve this. Multi-agent system comes from the need to automate where we use them as entities responsible for specific actions in the environment and data mining with the goal of having a good decision-making platform. The knowledge is intended to be achieved from the information that is collected from the environment. Since our research is based in a problem that happens in real scenarios and requires the access to a lot of physical and expensive resources, we conducted simulation experiments in order to reproduce the problem scenario under a controlled set environment. With this approach we can study different cases to learn from the environment and also use knowledge achieved in the simulation and use it in real environments.

Having the problem definition and our approach defined, we broke our development in three main phases. The first one was to develop a simulation tool that would be responsible for the physical simulation of the transfers between warehouses, being responsible for the movement of the trucks according to some characteristics (speed and volatility), the second phase was to model our agents entities according to our case of 2PL inventory management and the third phase was to run experiments and analyse the results.

4.2 Overall architecture

In our work, we have two main applications in order to get a full simulation platform that would use agents to orchestrate entities and have a clear visualisation of what happens during simulations. A simulation tool, created by us, and Agent Process Modelling (APM), an application where agents are modelled through processes, following Business Process Modelling Notation. The tool created was designed with the goals of being scalable, resilient, specific for the type of scenarios we want to study and flexible to future enhancements and additions of new features. Given this, we followed Microservice Architecture, building the application as a collection of loosely coupled services where each one is responsible for a certain scope. With the evaluation of the requirements established we divided the platform in three services: **canvas service** (4.2.1), **manager service** (4.2.2) and **data service** (4.2.3). The communication between the services can be made through HTTP protocol, sockets and Kafka.

As resources to our simulation platform we use Redis, DynamoDB and S3. We opted to go with these because of our previous experience on working with them and also because of their good relation performance/price compared to others similar. Redis is an open source, in-memory data structure store, used as a database, cache and message broker, and supports multiple data structures. We use it as a database to store data related to simulations that are running, in this way we have a safe and fast way to access to the data during simulations, granting a small delay when accessing to information needed, e.g., putting a new truck in simulation, inform that a truck arrived, etc. DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster database with built-in security, backup and restore, and in-memory caching for internet-scale applications. This resource is used for the storage of information which the access time won't impact simulations results, in this sense, it is used to store simulations created, canvas size, warehouses coordinates, trucks used and S3 file key of the schedules ((plans with the transfers that must happen between warehouses, by tick) that were generated for the simulation. Finally, S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. We use it to store the schedules and results of each simulation.

The description of each interaction with the services and resources. is explained in the following list, given the Figure 4.1. It describes (through the numbers), each interaction between the services and resources (databases and object storages), with the representation of the simulation platform architecture.

1. Represents the HTTP protocol established between Canvas service and Manager service. This communication is used for synchronous interaction between both where the Canvas requests to Manager and waits for a response. In this perspective, Manager acts as a RESTful service.

2. Asynchronous connection made between Canvas service and Manager service. This is made through sockets and is used only when we run a simulation since is the only case we need

to communicate from the Manager to the Canvas asynchronously.

3. Connection between the Manager service and Redis. Manager has read and write actions on Redis.

4. Connection between the Manager service and DynamoDB. Manager has read and write actions on DynamoDB.

5. Connection between the Manager service and S3 Bucket. Manager has read actions on S3 Bucket.

6. Connection between the Data service and S3 Bucket. Data has read and write actions on S3 Bucket.

7. Synchronous connection established between Manager service and Data service. This is used to delegate tasks from the Manager to the Data. In this perspective, Data acts as a RESTful service.

8. Represents the HTTP protocol established between Manager service and Agent Process Modelling (APM) platform. In this representation, both work as a RESTful service where Manager requests to APM and APM requests to Manager.

9. Synchronous connection between Manager service and Agent Process Modelling platform. This is established by Kafka.

Given the description of the overall architecture, with the communications established, the decoupling of the simulation platform into multiple services and the resources used, now we go through a more detailed description of each service, in the following sections, and respective scopes that each one represents and is responsible for.

4.2.1 Canvas service

Once we decoupled our platform into services, we had to build a service that would be responsible for all the frontend side, having not only the interfaces responsible for the user interaction and visualisation of the simulations but also the logic behind the transfers between warehouses, which is responsible for the movement of the trucks. Being a service dedicated to the frontend part, it is built in **NodeJS** and due to the need of visualise the simulations we used a library named **P5.js**, which allows to use the browser as a sketch where we can draw wherever we need in a efficient way. The service has only direct communication with the Manager service and with no other resource. This communication is made in two ways: asynchronous through sockets (the communication is bidirectional and is used only during a simulation to transmit information without the need of request for new updates) and synchronous through HTTP requests (used for all the actions that does not involve run a simulation). Figure 4.2 has the specific libraries used in the service that were described previously.

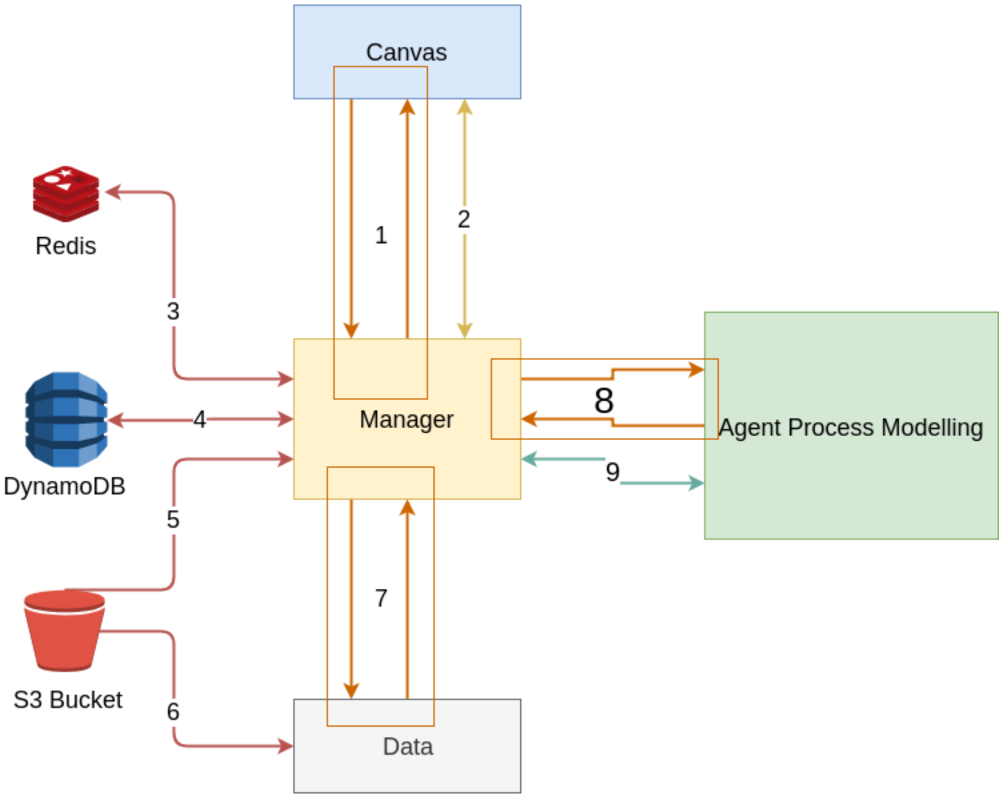

Figure 4.1: Overall architecture of simulation platform and interactions between services.

For the transfers logic between points, we defined a function that calculates the next position where the truck will move to. Each truck has as characteristics a **force** and **volatility**, impacting the time that will take to move from the origin to the destination. The formula to calculate the next position is the represented in Equation 4.1. The equation takes into consideration the current position of the truck and calculate the vector that point from its position to the destination, which is then divide by the force. Then, a random number between -volatility and volatility is generated. Finally, we subtract to the vector that points to the destination the number generated and the result is added to the current position. With this equation, the closer the truck is to the destination, the more impact the force and the volatility have on the movement. This was the method we calculated in order to have an variable transportation time that is impacted by truck configurations, which with a sample of multiple transports executed by different trucks we can use it to apply in a Data Mining model and improve the predictions of delay during transfers and picking the reliable trucks to do a transport.

$$\begin{cases} x = x + ((destinationX - x)/force - random(-volatility, volatility)) \\ y = y + ((destinationY - y)/force - random(-volatility, volatility)) \end{cases} \tag{4.1}$$

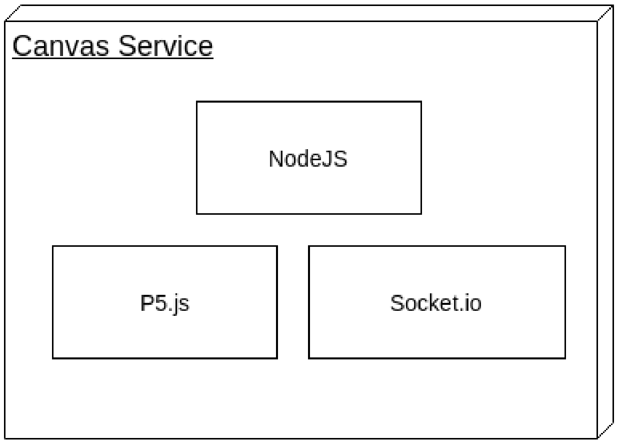

Figure 4.2: Canvas service framework and main libraries used.

4.2.2 Manager service

Given the definition of the service responsible for the interaction with users and also the logic of the transfers between warehouses, we had to build another service that would be responsible for the management of the resources used (redis, dynamodb and s3), providing CRUD (create, read, update and delete) operations to these, interact with the APM platform to manage the agents life and also to interact with the agents during simulations. Following the approach of the Canvas service, this one is built in Express, a minimal and flexible NodeJS web application framework that provides a robust set of features to create robust APIs. Manager service has the capacity to communicate through HTTP, sockets and Kafka. It is built as a RESTful service, behaving as an API for the Canvas service but also these two communicate through sockets during the state of running a simulation. In Figure 4.3 we identify the main libraries that the service uses. In the following list we describe the socket event topics that are defined and the description of what they trigger.

- **start_simulation**: Manager informs Canvas that all the agents were started and that it can start the simulation;
- **new_truck**: Manager informs Canvas that a transfer should be made, giving the transport information needed (truck details, agent that represents the truck and respective schedule);
- **tick_update**: Manager informs Canvas that the tick changed and sends new tick;
- **create_agents**: Canvas informs Manager to create the agents in the APM platform;
- **truck_point**: Canvas informs Manager that a truck arrived to an info point, sending the details about the transport at that time (truck and agent details, the info point number, the time the truck arrived to the info point and the ETA for the destination;

- **truck_end**: Canvas informs Manager that a truck arrived to its destination, sending the details about the transport made (truck and agent details, respective schedule made initially and the arrival time;

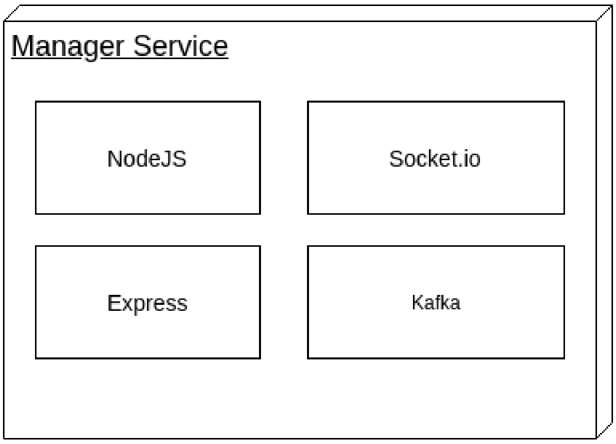

Figure 4.3: Manager service framework and main libraries used.

Besides of being a RESTful service it also does requests to APM platform API to manage the agents lifetime. This is the way of communication that the service uses to create agents when a simulation is started and also to delete them when a simulation is finished. During a simulation, Manager service uses Kafka to communicate with the agents. This communication is established to inform agents about the transfers that are occurring, more specifically, inform about the status of the trucks that are performing a transport, for example, a truck arrived to an info point or a truck arrived to a destination. Figure 4.4 shows the different communications between Manager service and the APM platform, with a more detailed representation.

4.2.3 Data service

With the definition of a service that takes care of the frontend side and another that is responsible directly for the management of the resources used and the simulations flow, we had the need to have another that would be responsible for the management of the data that is used in our simulation platform. In this sense, we built a RESTful service in **Python** with **Flask**, a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. The main goal of using Python is to take advantage from libraries like **Pandas** that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive in order to deal with data sets in a efficient and effective way. Given this, Data service manages the data sets that contains the schedules for each simulation, generating them providing the plans for each warehouse,and also the data sets that contains the results of each simulation that was made. All the data sets are

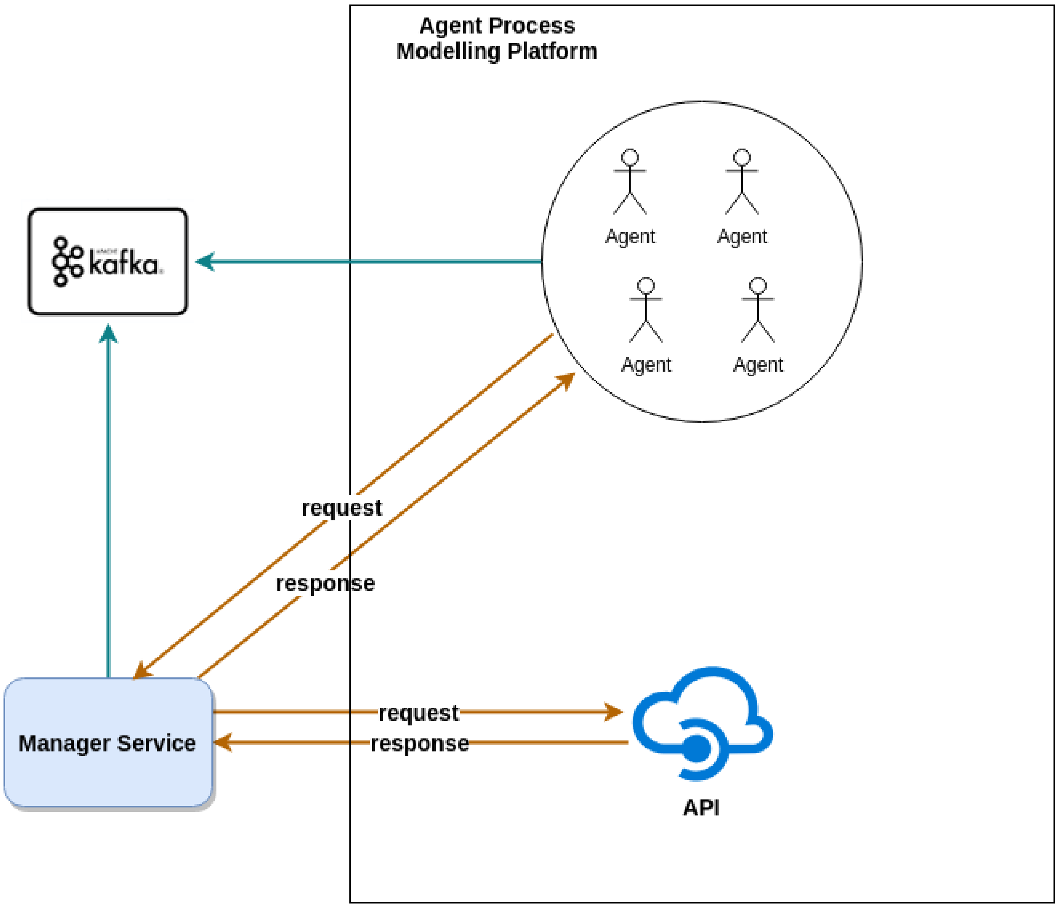

Figure 4.4: Manager service and APM platform interactions.

represented in CSV files and stored in a S3 Bucket. Figure 4.5 shows the main libraries used that have impact in the service capabilities.

A simulation can have more than one schedule, which means that the same scenario can be used with different plans for the warehouses. The schedules represent the transfers that are planned to be made. Table 4.1 contains the schedules file structure, where it shows in the first row that at tick 1 a truck should depart from origin with id 1 and with an ETA (estimated time arrival) at tick 4 to destination with id 2, carrying a load of 1010 units. Since Data service deals with all the data that is used to achieve knowledge from the environment, it also represents agents capabilities in terms of choosing trucks and verifying delays, by accessing the results of the Data Mining model trained previously. A better description on how this capabilities are accessed by the agents is provided in the next section.

4.3 Agent Process Modelling platform

Given the definition of the simulation tool developed, we used Agent Process Modelling (APM) [Rúbio et al., 2019] platform to develop the agents that would interact with the simulations. APM allows to model agents behaviours through Business Process Modelling (BPM), which means that

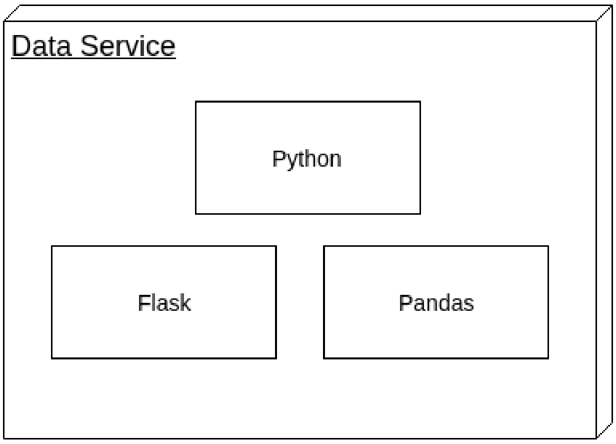

Figure 4.5: Data service framework and main libraries used.

each agent has a process that will represent all the capabilities and different states of the it. Figure 4.6 shows how agents are defined in the platform.

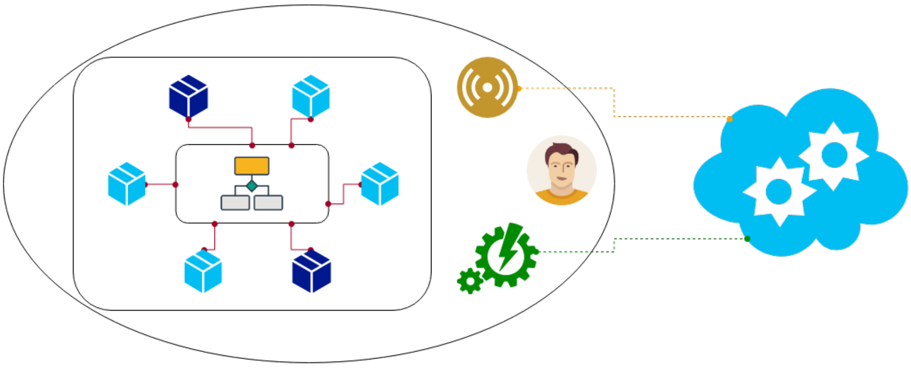

Figure 4.6: APM agent is autonomous process that orchestrates microservices as capabilities.

This approach has the goal of the agents don't be responsible for heavy computation processes, but instead, they orchestrate the flow and interact with other services (by HTTP or other mechanisms) to compute some tasks on the way. The illustration on how this is achieved is represented in Figure 4.7.

In order to allow agents to communicate between them, they are able to exchange messages through **Kafka**, a distributed streaming platform that allows to publish and subscribe to streams of records, similar to a message queue or enterprise messaging system. This is the method agents use to communicate with each other, where they have the ability to either do a broadcast of a certain message or send to a unique receiver. For the message to be delivered, the agent that is responsible to receive a message, should be in a state of catching the message before the another agent sends it. The messages are identified by name, support a payload field where data is sent and then we can specify if we want to send to a specific agent by placing the agent id in the field **receiver**, or it can do a broadcast to a certain types of agents by specifying the role with the field **receiverRoles** and finally if the agent does not specify any of this fields the message is sent to all the agents in the environment. In Figure 4.8 we see Agent 1 sending a message to Agent 2. In the moment Agent

departure_tick	arrival_tick	origin	destination	load
1	4	1	2	1010
2	4	2	8	2000
3	8	2	1	1400

Table 4.1: Schedule file representation of its structure.

1 sends the message, Agent 2 should already be waiting to receive it, or else the message will not be captured by agent 2.

As we described previously, an agent is modelled through Business Process Modelling which means that each agent has a process that represents its behaviour. In the APM platform, a process has the following fields:

- **Name**: represents what we want to call to the process. Although, it is not an identifier which means that we can have more than on process with the same name. The identifier is given and it's a number;
- **Description**: field used to do a description of the process. It is a optional field;
- **Version**: each process has a version in order to identify and use previous versions. When a process is modified, a new version is created;
- **Model**: it is where the behaviour of the process is created. After modelling the process with the canvas provided, it generates a BPMN file with the information need, related to the model;
- **Environment**: APM allows to create environments in order to have a better definition of the agents that interact with each other and reduce the scope of usage;
- **Role**: each process must have a role. This is used to identify the type of agents/processes and define scopes on groups of agents. The role is associated to the Environment so a process can only use a Role that was added to the Environment selected;
- **User**: represents the permissions the process will have, the user can be system, admin or user.

Once we have our processes defined, we can then create our agents. To create an agent we must provide the following fields:

- **Name**: represents what we want to call to the agent. Similar to the name of the processes, it is not an identifier which means that we can have more than on process with the same name. The identifier is given and it's a number;
- **Description**: field used to do a description of the process. It is a optional field;

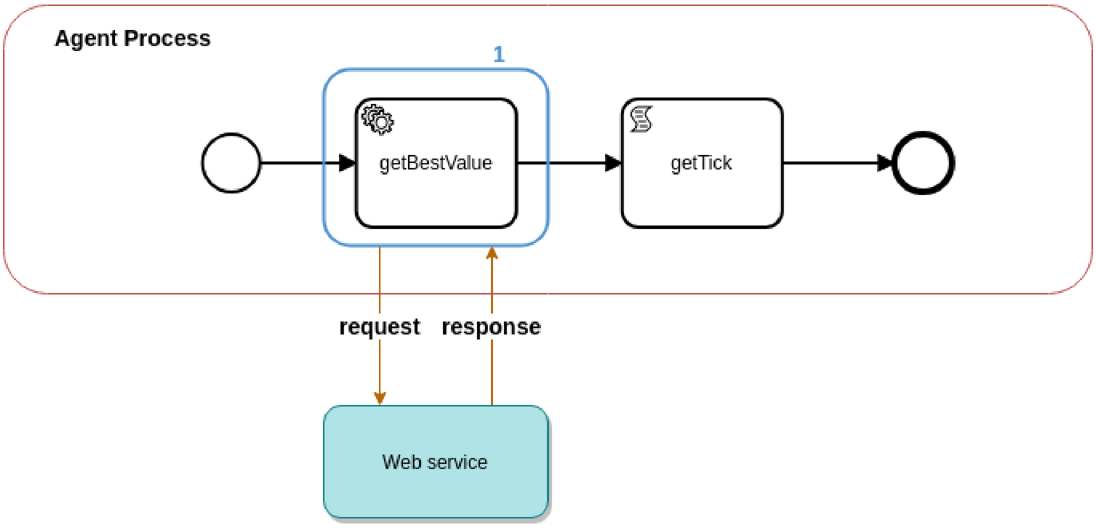

Figure 4.7: Agent process interaction with external service capabilities

- **User**: represents the permissions the process will have, the user can be system, admin or user. This overrides the User defined in the process;
- **Process Model**: in here we select the process we want the agent to have. This will be responsible for the agent behaviour;
- **Role**: each agent must have a role. This is used to identify the type of agents/processes and define scopes on groups of agents. The role is associated to the Environment so a agent can only use a Role that was added to the Environment selected. This is inherited from the process;
- **Variables**: agents can have initially variables defined. A variable is a key-value pair. The key is a text field and the value can be text or expression;
- **Goals**: agents can have also their own goals. These are defined as a representation of what agents want/have to achieve in the environment.

Given the details provided about the connection between the agents, processes, environments, users and roles, we can see a diagram that has the relations between these in Figure 4.9.

4.4 Simulation agents

Having a good knowledge of what Agent Process Modelling platform could offer to our simulation tool and how they would interact with each other, we put it all together in order to determine what would be the agents needed to our simulations. Approaching the context of 2PL inventory management we conclude that would be necessary to have two agents responsible to represent the two objects that must be managed in the simulation, the **warehouse** and the **truck**. Then, being the focus the warehouse transfers, we needed to have an agent that would be responsible for the

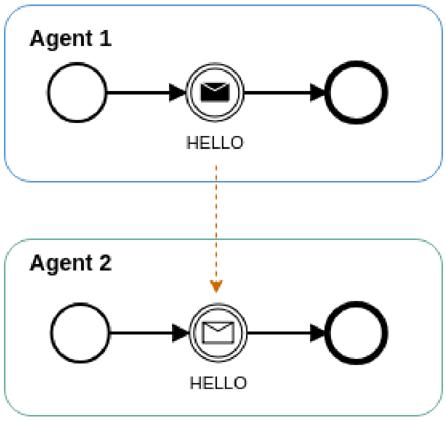

Figure 4.8: Agent communication scheme using message events

time counting of the simulation, which we identify as **ticker** agent. In the end, we defined an agent that will listen the counting of the ticker agent and inform the simulation tool when the simulation time is updated. With this implementations, we will have per simulation, one informer agent, one ticker agent, the number of warehouse agents is equal to the number of warehouses created in the simulation and the number of truck agents is equal to the number of warehouse agents times the number of warehouse agents. Each warehouse agent manages a single warehouse and the same applied to the truck agents, each one manages a single truck.

4.4.1 Warehouse agent

The warehouse agent is responsible to manage the transfers where the warehouse is the start point, the origin. More specifically, it manages what trucks realise a certain transport that starts from it and tracks those transports, taking decisions when needed. Figure 4.10 shows the process model of a warehouse agent for the first experience to be made. Initially the agent performs the task **getSchedules** to load all the transfers, through a HTTP request to Data service, where it is the starting point. After that, it waits for one of the three events shown, catch a END message in the **endTick** event and finish the process and consequently the agent stops, catch a TICK message in the **ticker** event which informs the warehouse the new tick of the simulation and from there the truck enters in the task **checkTickSchedules**, which gets the transport that should start at that tick from the warehouse and then it verifies if there's any or not in the gateway **hasSchedule**. If there is no transport to be made the warehouse goes back to wait for one of the three events. In case it has, the agent does a broadcast of a NEW_LOAD message, in the **newLoad** event, to all the truck agents with the details about the transport (origin, destination, start_tick, estimated_arrival, load) and waits for a reply. When it receives a reply, it gets the offer that was sent in the message, in the task **getOffer**.After that, the agents send a START_TRANSPORT message to the truck agent that made the offer through the **startTransport** event and goes back to the initial state.

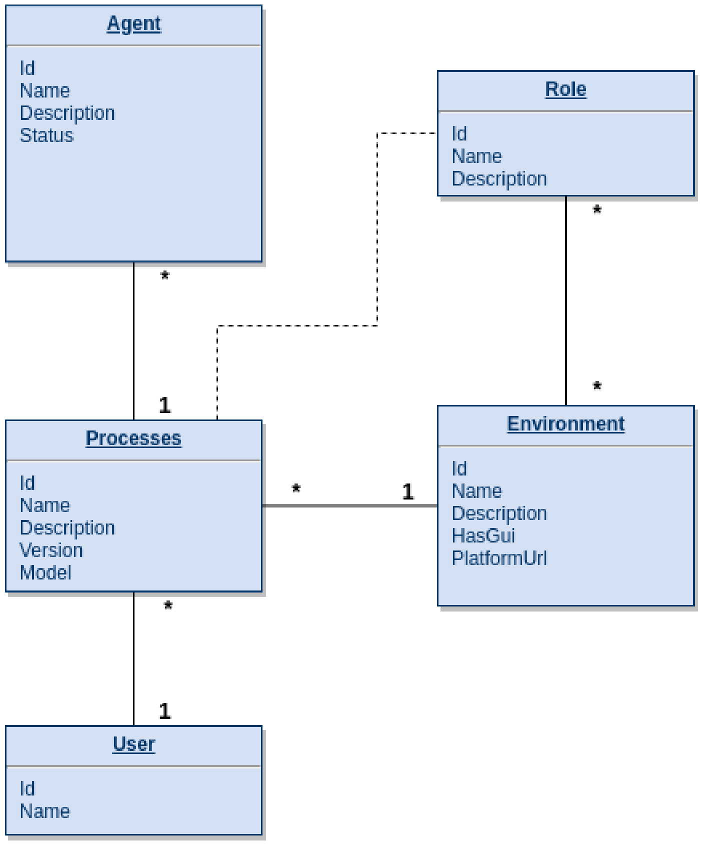

Figure 4.9: Agent Process Modelling relationships

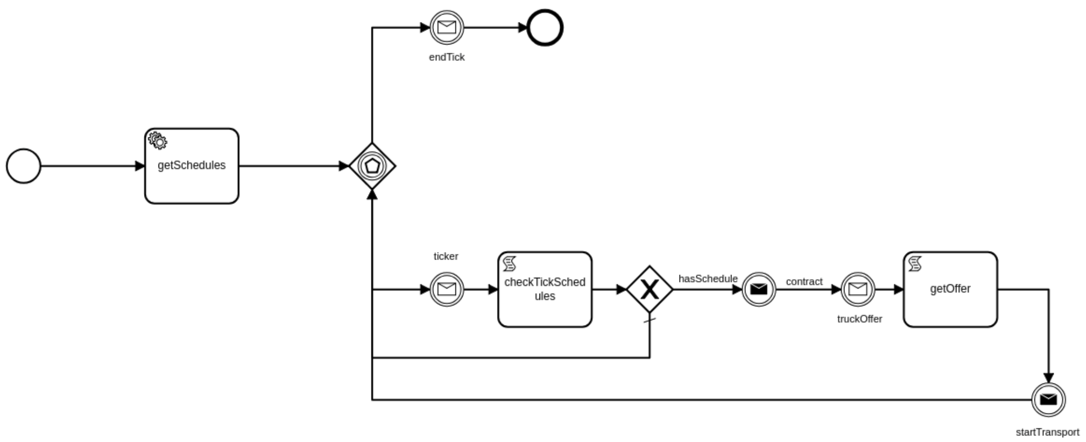

Figure 4.10: Warehouse agent process model for simulation 1.

For the second simulation, the warehouse agent has additional capabilities. See Figure 4.11 that has the process model of the warehouse agent for the second simulation. When it starts, the agent loads the trucks predicted delays through a HTTP request to the Manager Service in the task **getTrucksAgents**, which returns an array, ordered, according to the predicted delays of each truck. For the picking of the trucks, the process now is defined in order to pick the best truck that is available Instead of doing the broadcast, it get the first truck from the array in the task **getTruckId** and sends a message asking for its availability. The answer from the truck is processed and then it is verified if the truck is available in the gateway **truckIsAvailable**. If it is, the warehouse sends a message to the truck, to start the transport, otherwise, it goes back and picks the next truck on the list.

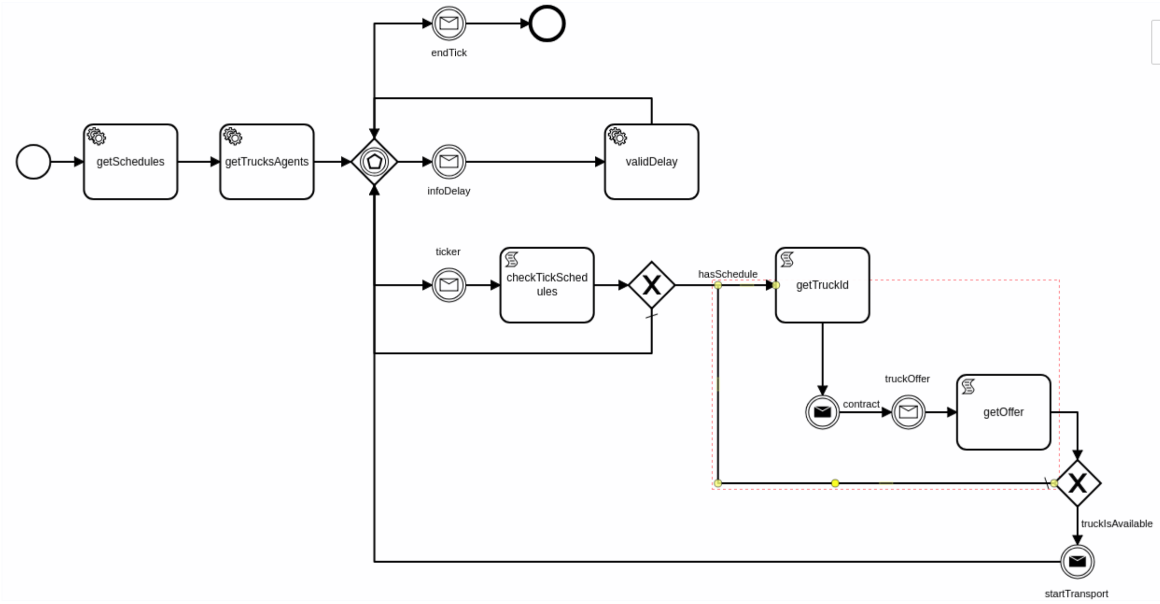

Figure 4.11: Warehouse agent process model for simulation 2.

When a warehouse agent is created, it has assigned some start variable that are mentioned in Table 4.2. These are needed for the simulation.

Start variables	**Description**
id	Identifier of the warehouse in the Canvas. Each warehouse agent has an id of the warehouse they represent in the Canvas
simulationid	Identifier of the Canvas simulation. When a simulation is created it has an id
runsimulationid	Identifier of the simulation that is running. When we run a simulation, this id is generated to store information related to what is running (agents, canvas layout, etc)
schedule	Identifies the schedule used in the simulation that is running. When a simulation is created, a schedule is generated containing the transfers that have to be realised. The agent uses it to load the transfers initially

Table 4.2: Start variable of warehouse agent.

4.4.2 Truck agent

With the definition on an agent to orchestrate the behaviour of warehouses, we defined the agents responsible to manage the trucks in simulation. A truck agent represents one truck, it leads with the interaction with warehouses when a new transfer is announced and also manages the trip of the truck during the transfers. Figure 4.12 shows the process model for the first simulation, of a truck agent. When the agent starts it waits for one of three events: catch a END message in the **endTick** event and finish the process and consequently the agent stops, catch a NEW_LOAD message in the **newLoad** event and check in the **verifyAvailability** gateway if it is available to realise a transport at that tick. It it is not available, goes back to initial state. In case it is available, the agent calculates the proposal in the **calculateProposal** task, sends it to the warehouse that announced the new transport and goes back to the initial state. Finally, catch a START_TRANSPORT message in the **startTransport** event which informs the agent to start a transfer in the simulation. When the agent receives this message, it does a HTTP request to the Manager service to place the truck in the simulation in order to perform the transport. Then, the agent waits for two possible messages that might come from the simulation tool during the transport. These messages can be a INFO_POINT message that is caught on the **infoPoint** event where the agent is informed that the truck arrived to an info point and the respective tick when arrived. The agent then verifies if it is late and if so, it notifies the warehouse responsible. It can also receive a END_TRANSPORT message through the **endTransport** event which informs the agent that the truck arrived to the destination and it goes back to the initial state.

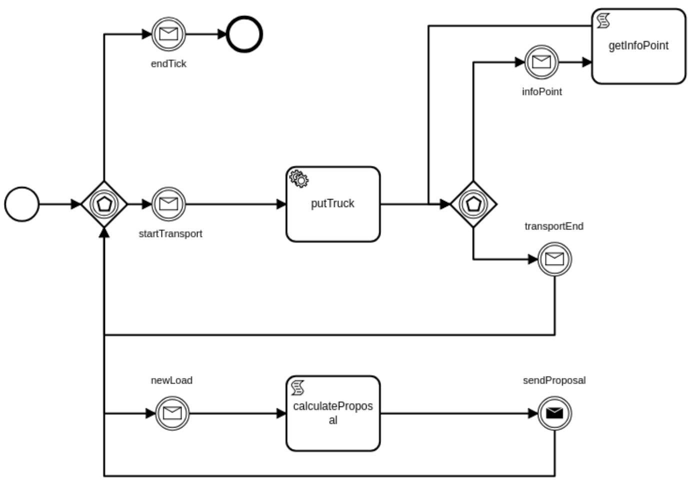

Figure 4.12: Truck agent process model for simulation 1.

For the second simulation, the truck process has new capabilities to interact with the warehouse. Figure 4.13 shows the process of the truck agents in the second simulation. First, it needs to be available to answer warehouses when they ask for their availability and track the truck status during a transfer. For this, we used a parallel gateway, which allows the trucks to listen for the

startTransport and **endTick** event, while it is catching and processing the **newLoad** event. Also, when the truck starts a transport it updates its availability in the task **availableToFalse** and when it finishes a transport, it also updates its availability in the task **availableToTrue**.

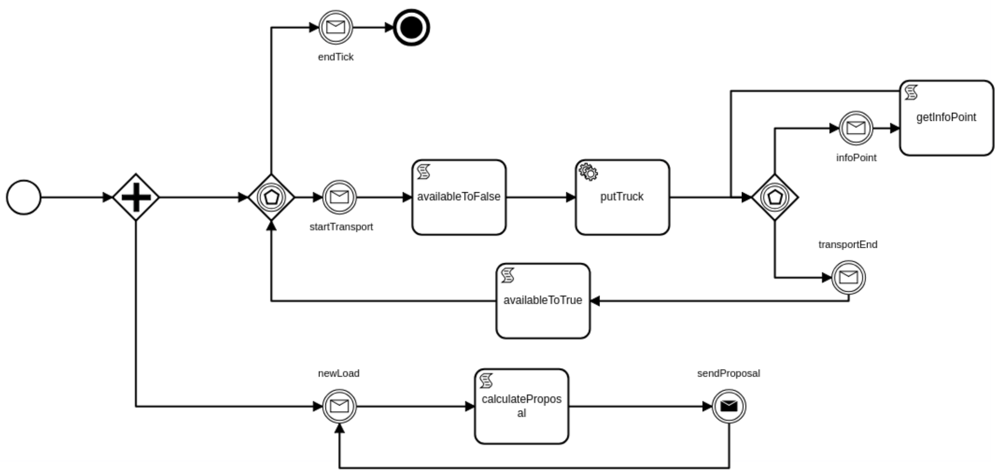

Figure 4.13: Truck agent process model for simulation 2.

A truck agent when is created is assigned with some start variables that are specified in Table 4.3.

Start variables	**Description**
id	Identifier of the truck in the Canvas. Each truck agent has an id of the truck they represent in the Canvas
simulationid	Identifier of the Canvas simulation. When a simulation is created it has an id
runsimulationid	Identifier of the simulation that is running. When we run a simulation, this id is generated to store information related to what is running (agents, canvas layout, etc)
companyid	Truck belongs to a company. This is used as an identifier of the company
price	Value the truck covers to realise a transport
force	Force that is applied in the truck movement
volatility	Value used to calculate the vector generated outside of the right trajectory, to vary the truck movement

Table 4.3: Start variable of truck agent.

4.4.3 Ticker agent

Having defined the agents responsible for the entities that are inserted in the simulation (warehouse and truck) we had to create another responsible for the timing of each simulation that is made. For this we created the ticker. It is responsible for counting the ticks and inform the agents present in the environment every time the time is updated. Figure 4.14 shows the process model of the ticker agent. When the agent starts, it increments the tick counter and verifies if the tick is the last of the simulation. If it is not, the informer sends a message to all the agents in the environment, by not specifying neither a receiver or a receiverRole, with the tick value and waits the timestamp defined to update the tick. In case it is the last tick, it sends a message to all the agents in the environment informing that they got to the end of the simulation.

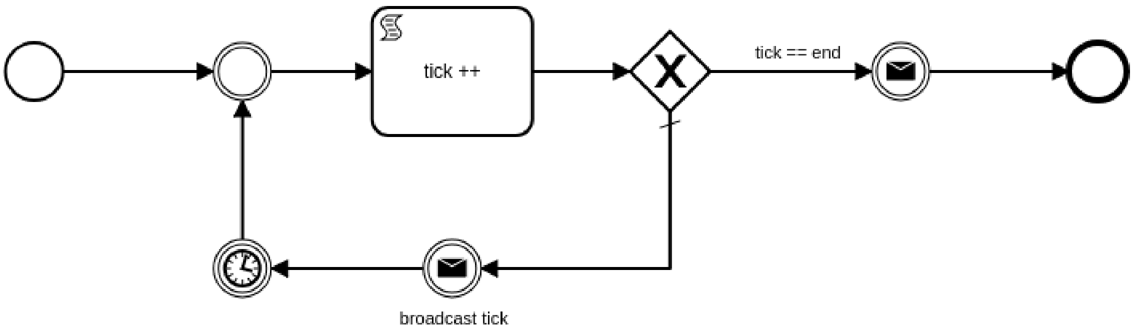

Figure 4.14: Ticker agent process model.

When a ticker agent is initialised, it needs to have the start variables mentioned in Table 4.4.

Start variables	**Description**
tick	Variable used to store and count the ticks time
end	Represents the value of the last tick in the simulation. When tick is equal to this, the simulation stops
interval	Duration between the tick update, imposing how long,in time, the tick takes

Table 4.4: Start variable of informer agent.

4.4.4 Informer agent

With our agent responsible for the timing, we had to create another that would be responsible to inform the simulation tool about the tick update during the simulation. For this we created an informer agent. Figure 4.15 shows its process model. The agents starts and waits for one of two possible events: catch a TICK message through the **tick** event, get the tick value that comes in the payload of the message and do a HTTP request to the Manager service informing the new tick of the simulation, or catch a END message through the **endTick** event, do a HTTP request to the Manager service that the simulations has ended and finish the process.

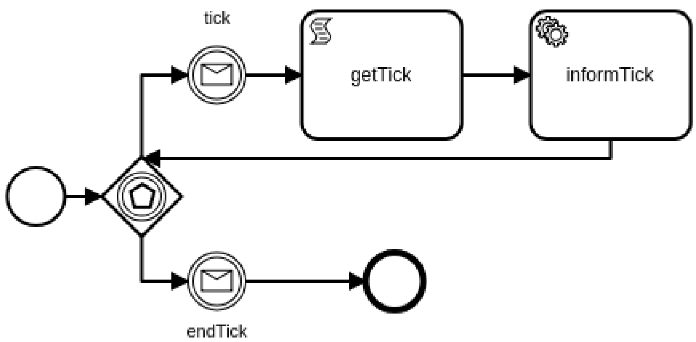

Figure 4.15: Informer agent process model.

The informer agent is assigned with the start variable described in Table 4.5.

Start variables	**Description**
runsimulationid	Identifier of the simulation that is running. When we run a simulation, this id is generated to store information related to what is running (agents, canvas layout, etc)

Table 4.5: Start variable of informer agent.

4.5 Data Mining as a Service (DMaaS)

The data that is collected in the first experiment is processed, so we can have a clear analysis of the data, and then is applied in multiple Data Mining models, in order to calculate which one gives better results. To analyse multiple models, we use RapidMiner AutoModel, a platform that allows to build predictive models through the browser using automated machine learning. This platform works as a Data Mining as a Service (DMaaS), which is described as a software and computing infrastructure that allows interactive mining of scientific data in the cloud [Tejedor et al., 2016]. The platform allows to deploy the models that were trained, which makes them available as a service, providing predictions given the inputs needed, through HTTP requests. To build a predictive model in RapidMiner AutoModel, we have to do four steps. First, we need to upload our data, which in our case is a CSV file (see Table 4.6, which contains a example of the file). Second, we need to choose, from the data uploaded, the column we want to predict. In our case it is the delay column. Third, we need to choose the inputs that will be needed to predict, which will be the truck_id and the company_id. Finally, the fourth step is to choose the models we want to run

(Decision Tree, etc). After all the processes, when the models finish to run we can analyse the results of each one and deploy the model we want to use.

truck_id	company_id	departure_tick	eta_tick	arrival_tick	delay	i_p_1	i_p_2	i_p_3
1	1	15	19	21	2	18	19	20
1	1	40	42	43	1	41	42	42
2	1	10	12	11	-1	10	10	11
4	1	1	4	4	0	2	3	4
7	2	25	28	34	6	26	29	29
10	3	2	4	7	3	4	6	7

Table 4.6: Example of data structure uploaded to RapidMiner as a CSV file.

For our case, we manually go through the processes described above and after deploying the model, we instantiate, as an environment variable, the URL of the service that provides the description in our Manager service. Then, when the user wants to run the simulation with the capabilities of the warehouses to consult the delay predictions of the trucks, the Manager service does the requests to get the predictions and each warehouse, when initialised, requests to the manager the predictions. See Figure 4.16 to have a clearer vision on how this is achieved.

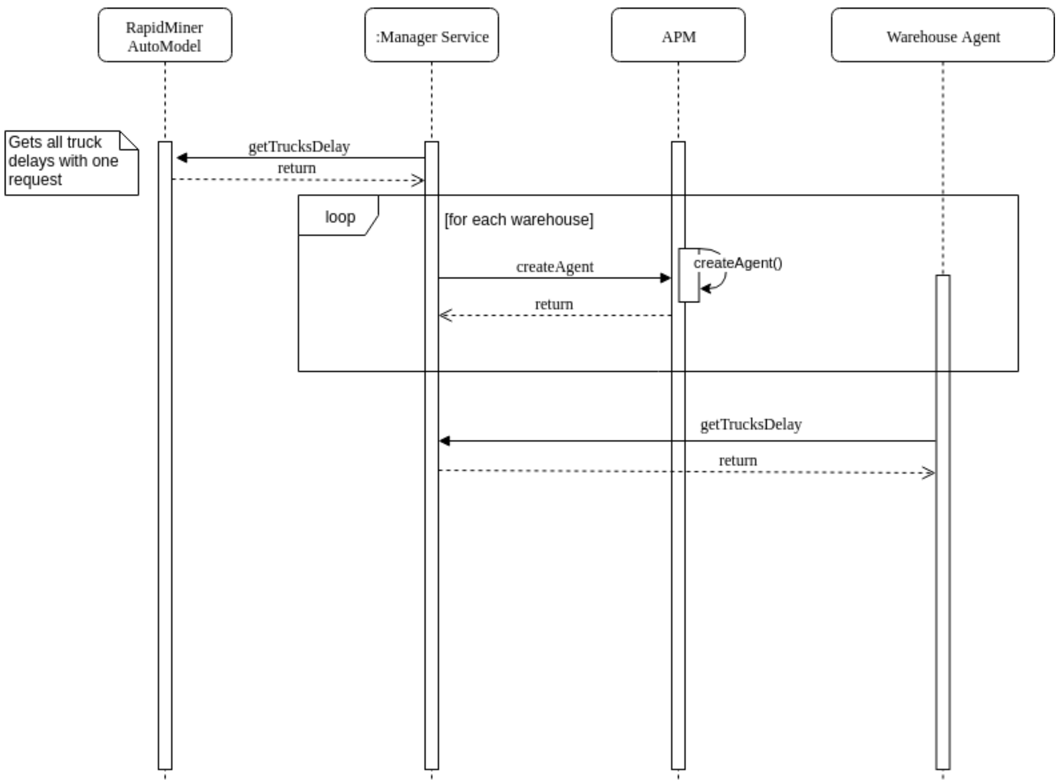

Figure 4.16: Agent with IoT results in a agnostic cyber-physical system.

4.6 Simulation Workflow

The simulation tool created had three main requirements: to enable the development of different environment layouts on a warehouse transfers context, have the ability to run different schedules for the same simulation and then run a pair simulation-schedule as many times we want. For this, we built three menus: a menu that would be responsible for the initial interaction with the user in the landing page, a menu specified for the creation of simulations and a menu where the user runs the simulations. In the landing page we designed a simple interface that has all the main actions provided with the click of a button as shown in Figure 4.17. In the figure we identify the different scopes that the user can interact. (1) User can create a new simulation, (2) the user starts a new simulation where the simulation layout and respective schedules are represented below and (3) the user can switch the simulation and/or the schedule that wants to run. In order to analyse the schedules, the user is able to download them. The schedules come in a CSV file format.

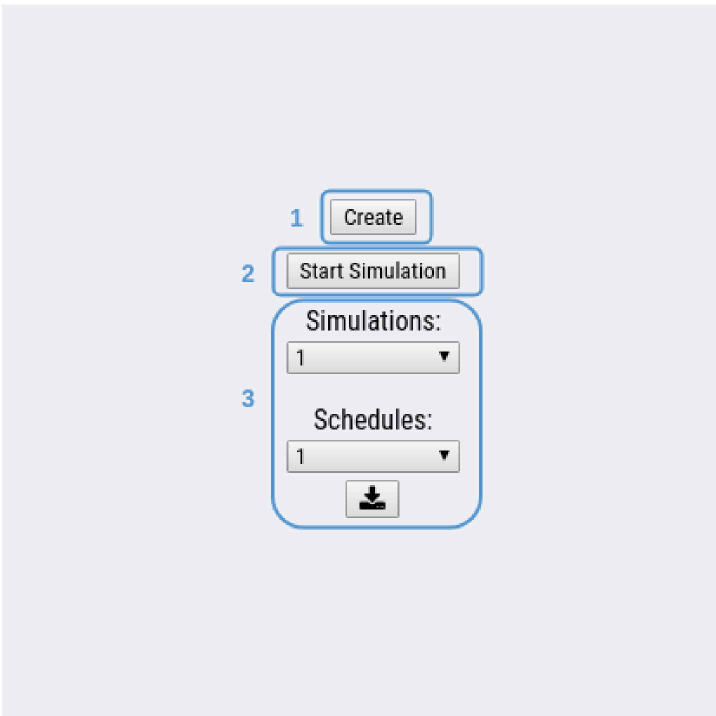

Figure 4.17: Simulation options

Given the scopes present in the menu present on the simulation platform landing page, create and run a simulation contains more interactions with the user and also technical details about how both processes work. In the following sections we explain how they work, defining the user work flow and interaction with the platform and also what happens in the platform during each process, referring the actions that are made by the services in the background.

4.6.1 Create simulation

When the user clicks in the **Create** button a new clear canvas is shown with the instructions that the user should follow in order to create a simulation. During the process of creating the simulation there are two phases, one is to place the warehouses and the other is define the info points between them. Figure 4.18 shows a state diagram with the flow on how to create a simulation.

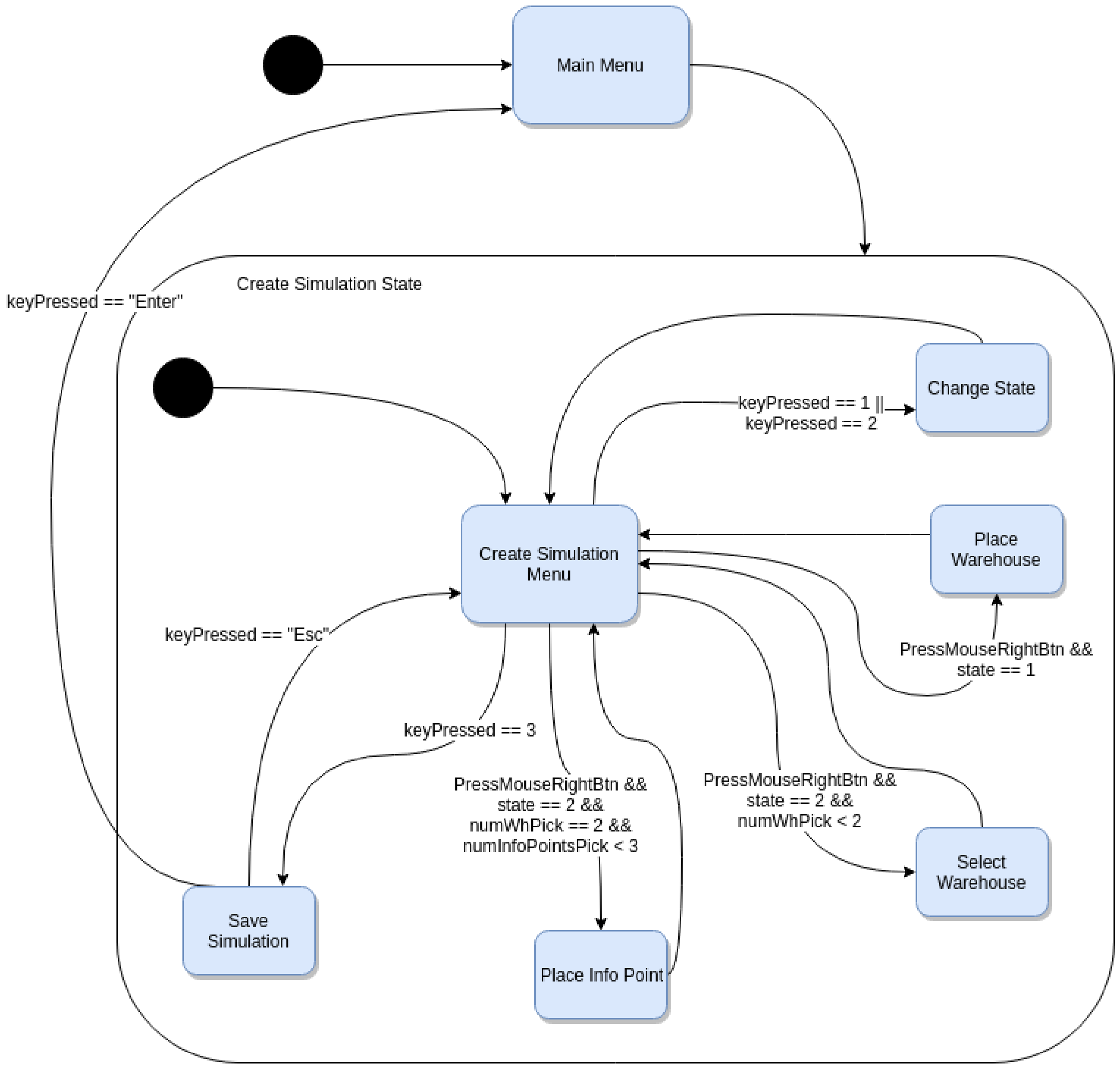

Figure 4.18: State Diagram representation of respective flow to create a simulation.

When the user is in the **Create simulation State**, the canvas has a own state (state 1 is for the user place the warehouses and state 2 is to define the info points) and starts listening both keyboard and mouse inputs. In state 1, the user should click in the right button (*PressMouseRightBtn*) of the mouse and a warehouse will be drawn in the position where the click was made. In state 2, the user should initially click in two distinct warehouses that wants to define the info points and after than, place the info points according to his preference. To change the states, it is just necessary to type the state that the user wants, key with the number 1 or 2 (*keyPressed*). The user is allowed to make the transition of states whenever he intends, but once it starts the definition of the info points related to a certain origin-destination he must finish it before define others. Finally, if everything is set, the user must click the key with the number 3 to finish creating the simulation. When the key is pressed, a pop up appears to confirm or cancel the creation. The user has to press "Enter" to conclude or "Esc" to keep the development of the simulation. In Figure 4.19 we see how the interface looks during the process to create a simulation and the interactions in both phases.

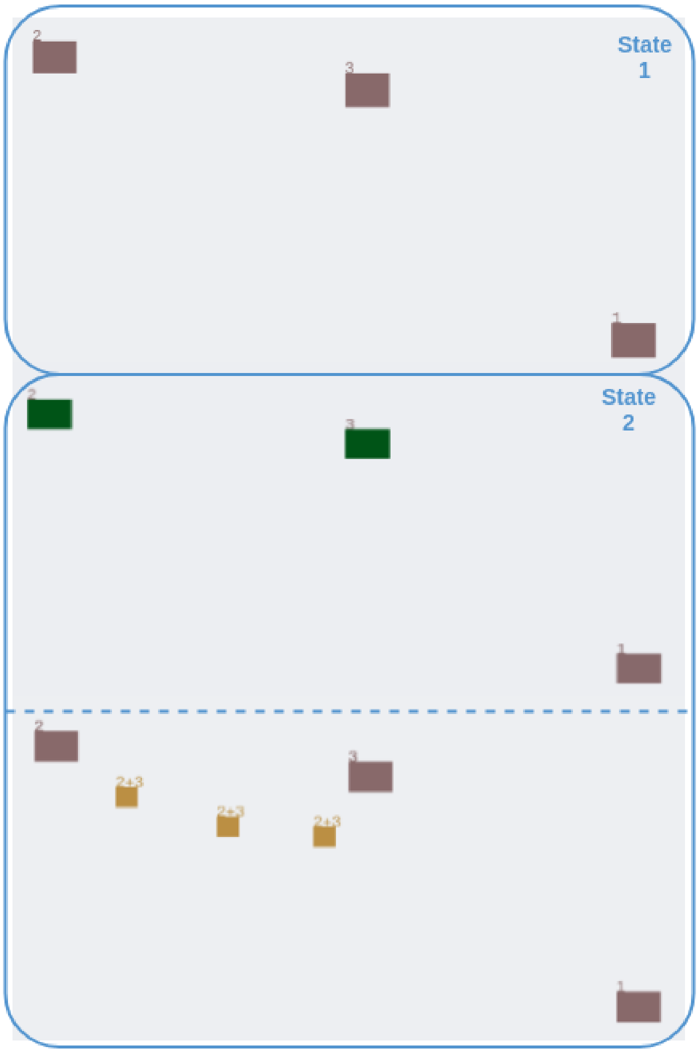

Figure 4.19: Create simulation menu with the different states. 1 - Place the warehouse. 2 - Select the warehouses and then define the info points between them.

In the backend, once the user creates a simulation, the configurations are sent from the Canvas service to the Manager service. Then, they are stored in a table on DynamoDB and the Manager service triggers the Data service to generate a schedule for the simulation created. The data stored are the canvas dimension (*height* and *width*), the canvas id, a list with the warehouses configuration (*id*, *x coordinate* and *y coordinate*), a list with the info points configuration (*origin*, *destination* and a list with the points coordinates), a list with the trucks configuration (*id*, *companyId*, *price*, *volatility* and *force*) and a list with the file keys of the schedules.

4.6.2 Run simulation

After creating the simulation it will be available to run in the dropdown input presented in the landing page. Associated to the simulation is the schedule with the plan of transports for each warehouse, as we said before. Once the user starts a simulation by clicking in the **Start simulation** button, the canvas is cleared and drawn with the simulation layout. Behind he curtains, a lot is going on. By steps the platform loads the simulation to the canvas, creates the agents that will represent the warehouses, trucks and the informer, and finally when all are initialised it creates the ticker agent that will trigger the start of the simulation. In order for this to happen, the processes must be already created in the Agent Process Modelling platform. Manager service is responsible to create the agents through API and it uses the ids and versions of the processes and the ids of the agent roles that has stored in the environment variables. Figure 4.20 shows the initial process to start a simulation.

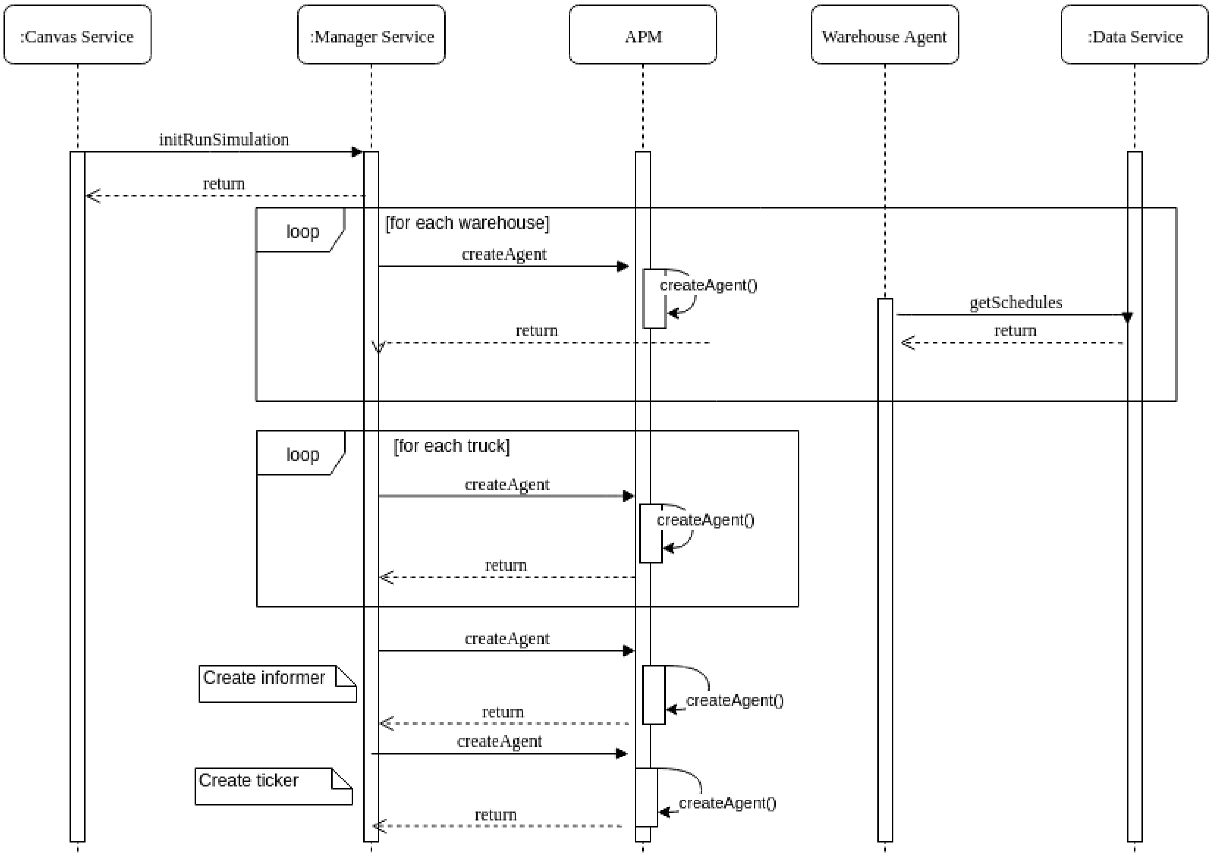

Figure 4.20: Diagram with the sequence of messages exchanged between the services, APM and agents to start a simulation.

Once the agents are initialised and the simulation starts the user just needs to wait while watches the transfers occurring in the canvas. At every tick, warehouses consult their schedules and negotiate with trucks to realise the transfers planned. When a truck does a offer and it is accepted, it stops listening the events about new transports and goes to another state to do the transport and track what happens on the way. Initially, when a truck starts a transfer, does a post request to start the simulation of the movement in the simulation platform and waits for notifications from the simulation about the movement, arrive to info points and destination. In Figure 4.21 we show the sequence of interactions between the truck agent, the manager service and the canvas service.

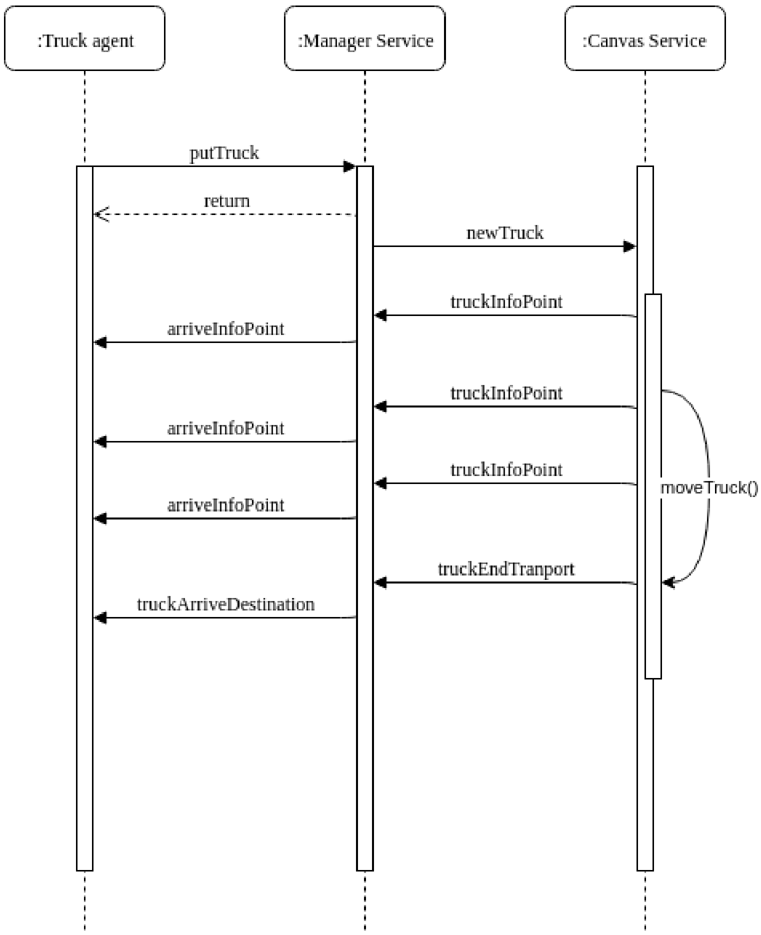

Figure 4.21: Diagram with the sequence of messages exchanged between the services and truck agents during a simulation of a transport.

In order to have a good management of the simulation that is running, information about it is stored in the Redis instance. In this way, we control the agents that are being used in the simulation and the respective objects in the simulation layout that they are responsible. In Table 4.7 we have a representation of the storage on Redis during the simulation with the data related to the simulation and the agents used.

runSimulationId	Identifier of the simulation that is running
canvas	Canvas configurations (height and width)
warehouses	Warehouses visual configurations (coordinates and id)
trucks	Trucks configurations (id, volatility, force, companyId and price)
infoPoints	Info points visual configurations and relations (origin, destination and list of points with coordinates)

agentsSimulationId	Identifier of the agents used in the simulation that is running
warehouses	Object with warehouse agents ids and names and respective warehouse id
trucks	Object with truck agents ids and names and respective truck id

Table 4.7: Storage on Redis with the configuration of the simulation in the left and with the mapping of the agents to the objects in the right.

During the simulation the user is able to see the trucks moving, going from the origin warehouse to the warehouse on the destination and passing through the info points on the way. Figure 4.22 is an example of what the user sees. The canvas also shows the tick count.

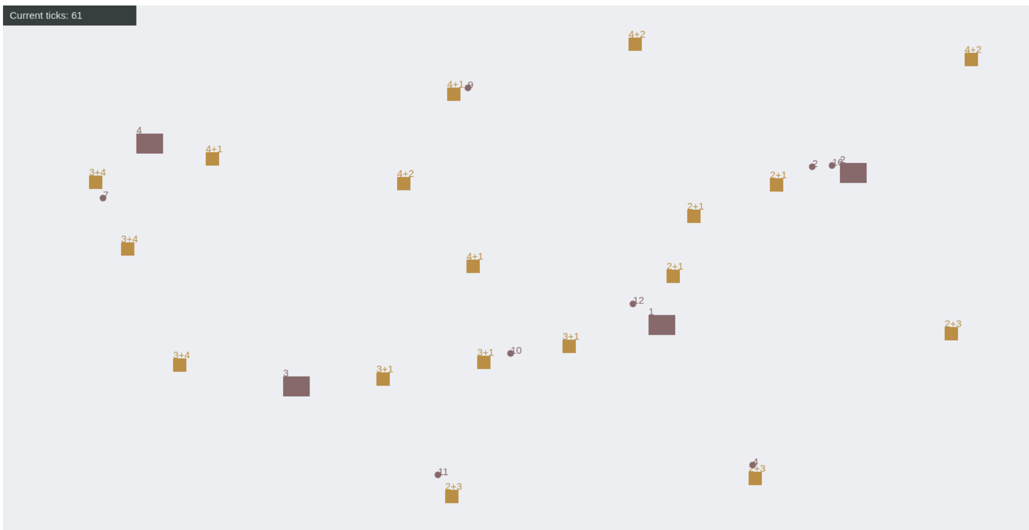

Figure 4.22: Simulation running with trucks transporting units between warehouses.

When the ticker gets to the end, it broadcasts the message to all the agents in the environment informing that they reached the end of the simulation and all stop. The simulation platform is also informed which then finishes drawing the simulation and redirects the user to the initial menu.

After the simulation is finished, the user can consult the flow and interaction of each agents, in order to consult the results about the transfers made and also the times on each transfer, not just between origin and destination, but also between info points.

4.7 Summary

In this chapter we reviewed all the implementation made during our research. We separated our development in two phases, first we defined and built a simulation platform related with the context of warehouse transfers and then we created the processes for each entity present in our simulations, integrating the simulation platform with the different communication mechanisms existent in the APM platform. We described both technical and scientific approaches made, presenting the reasons on why we have chosen the resources, technologies and languages used in our study. After the development made, we did experiments to analyse the impact of using data initially collected from the simulations and use it to learn and improve decisions in future simulations with the same configuration. The next chapter describes the experiments made and the results and conclusions achieved.

Chapter 5

Experimental Evaluation

5.1 Introduction

After the definition and development made, we now have to evaluate our case of study by running different simulations related to the context of the research. We separated the experiments in two types: in the first experiment, the picking of trucks by the warehouses is made according to their availability and which one answers faster to the announcement of a new load and in the second, warehouses have knowledge achieved from the first experience about trucks, picking trucks with the order from the most successful to the least. We then, evaluate the results and compare them, in order to conclude the impact of having knowledge about partners and also to analyse the prediction of delays, according to the info points. In the next section we detail the problem that we simulated.

5.2 Problem modelling

With the context for our application being 2PL transfers management, in our scenario warehouses have a plan to follow and interact with trucks in order to perform the transfers that are in the plan. The objective is to run an initial simulation where warehouses initially select trucks who are available without taking in consideration their performance and in a second simulation, warehouses have the ability of choosing the trucks according to their estimated delay, by learning from the first experience. In this sense, our simulation contains two main entities that arrepresented by agents: the trucks and the warehouses. Trucks can be associated to different partners (carriers) and warehouses are represented by a single company. In the following list there is the detailed responsibility/ownership of each agent:

- **Truck**, it is responsible to interact with warehouses to propose offers for the transfers requests, when it is available. Track its status during a transfer and collects information in the departure, info points and arrival. The information collected is the following:

1. Departure tick
2. Estimated arrival tick
3. Info point number
4. Tick at each info point
5. Arrival tick

- **Warehouse**, The warehouse consults the schedules that it loads initially and pick the trucks to realise a transfer. It it responsible for the management of the trucks that depart from it.

In Figure 5.1 we can see a representation of the transfers in our problem domain. There are info points for each pair of warehouses where trucks must pass. The info points are divided in the route according to users preference.

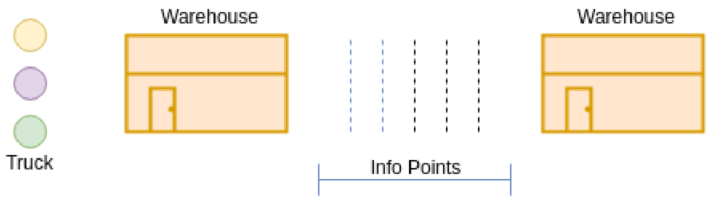

Figure 5.1: Representation of the entities in the scenario.

5.2.1 Evaluation Metrics

In order to evaluate the simulation and the results, we defined two metrics: estimated travel time (ETT) and delay. This allows us to validate important measures for the environment which are the fulfilment of the estimated delivery times and evaluation of trucks performance in terms of delay. In the following list we explain how each metric is evaluated:

- **Estimated travel time**: For each transportation between warehouses, an estimated travel time is defined. With this, we also calculate (diving the estimated travel time, proportionally by the number of info points) an estimated travel time for each info point, in order to evaluate the performance not just between the origin and destination, but also between the info points. If the truck departs on time, the transportation time made should be smaller or equal to the estimated;

- **Delay**: Given the estimated travel time and the actual travel time made, we calculate the delay, also between origin, info points and destination, in order to evaluate the performance of the trucks.

These metrics will be calculated and analysed in the two simulations. The goal is to, in the end, conclude about the results achieved by comparing both simulations, analysing the impact of having knowledge to pick trucks and also the possibility of predicting delays according to the information that comes from the info points.

5.3 Simulation without Data Mining

The simulation created for both experiments, in our platform, contains **4 warehouses**, **3 info points** for each pair of warehouses and **16 trucks**, with 4 companies and 4 trucks each. In Figure 5.2 we can see the simulation scenario used in the experiences made.

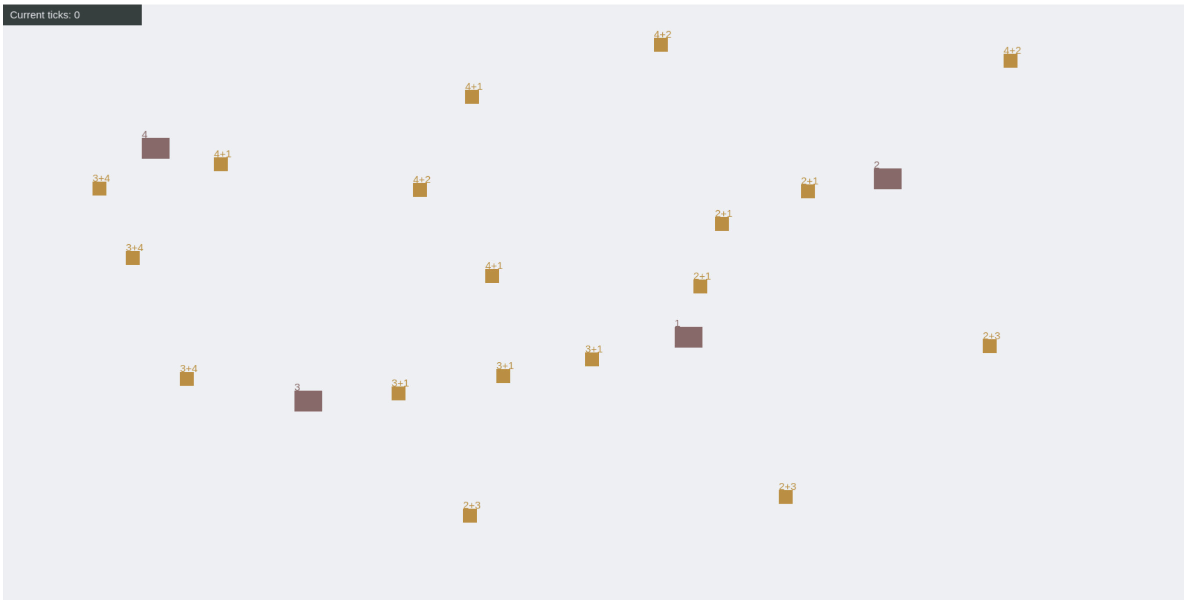

Figure 5.2: Simulation scenario used in the experiences.

For the trucks, we gave different configurations that impact their performance (force, volatility and company_id) in order evaluate the results in the end. For the same company, trucks have the same configurations. In Table 5.1 we have the configurations for each truck.

company_id	id	force	volatility	price
1	[1,4]	40	4	800
2	[5,8]	60	4	1000
3	[9, 12]	80	4	1000
4	[13,16]	100	4	900

Table 5.1: Trucks configurations for the experiments.

Following the trucks configurations we expect that the performance of trucks in company 1, will be better than company 2, company 2 will be better than company 3 and company 3 will be better than company 4. We ran ten times a simulation with 60 transfers, giving a total of 600 transfers.

After running each simulation, we collect the CSV files present in the trucks agents and store them in a folder, so we can process in the end of all the simulations, for this first experiment. In the end, when we have all the ten simulations made, we run a jupyter notebook script to process the data from the files collected and save the results in a single CSV file. When we run the script and get the result file, we analyse the results achieved and we came with the analysis expected. Since in our first experiment, the picking of trucks is based in the truck that is available and answers first to the announcement (warehouse agent broadcasts a message, informing a new transfer to all the trucks), there are no order and any truck can be selected to realise a transport if it is available. Given this, we can see in Figure 5.3 that company 1 made more transfers than company 2, company 2 made more transfers than company 3 and company 3 made more transfers than company 4. Since the configurations of trucks gets worst with the companies order, this results are due to the fact that, since trucks in company 1 are faster than the others, they are more available. This applies for the comparison of the other companies. Taking in consideration the delays that were calculated, we got **44 transfers without delay**, which means that from the 600 transfers simulated, only 44 arrived in the time that was planned or before.

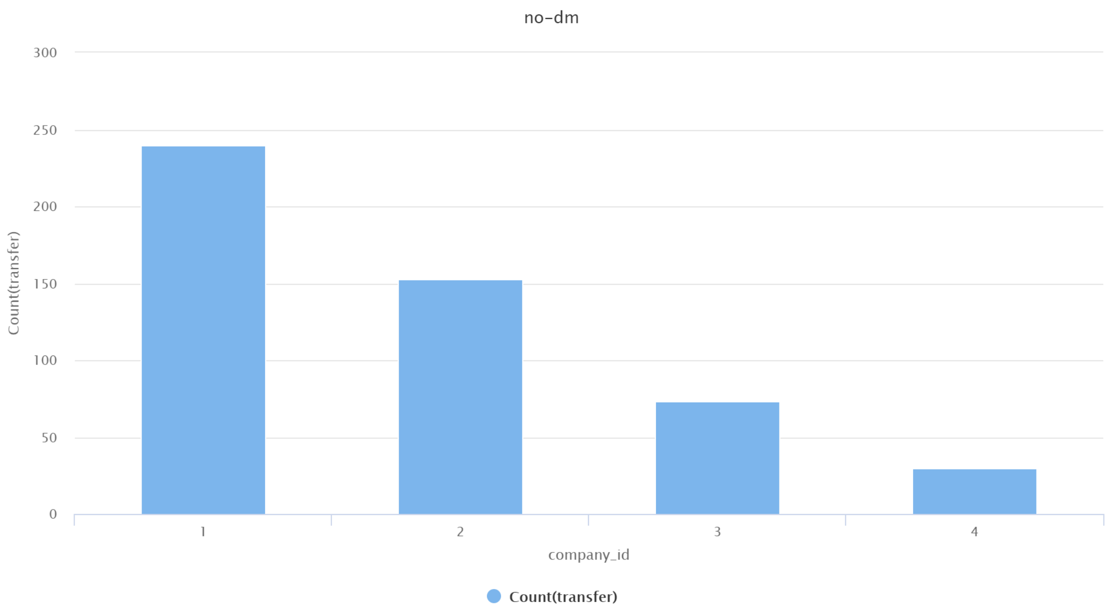

Figure 5.3: Number of transfers made by company, in first experiment.

Now, looking to the information gathered about the info points, the results are also the expected. Figure 5.4 shows the median of the percentage delay of trucks for each info point. We can see that the percentage of delay of the trucks from company 1 and 2 is smaller than company 3 and 4.The percentage of delay is normally smaller in info point 1 than in the others. This can be related to the fact that, since the trucks travelled only in first part of the course, the performance of trucks does not have a lot of impact because there was not that must field to evaluate the trucks speed. This theory can be reinforced, given that the difference between the delays of each info point, is bigger in the trucks who have bad performance. Having a perspective at company level, in Figure 5.5 we can see the relation of the delays for each info point, per company. We ca see that the median of delay in company 1 does not differs too much between the info points, while in company 4 the difference is more accentuated.

Figure 5.4: Median percentage of delay, per trucks for each info point, in first experiment.

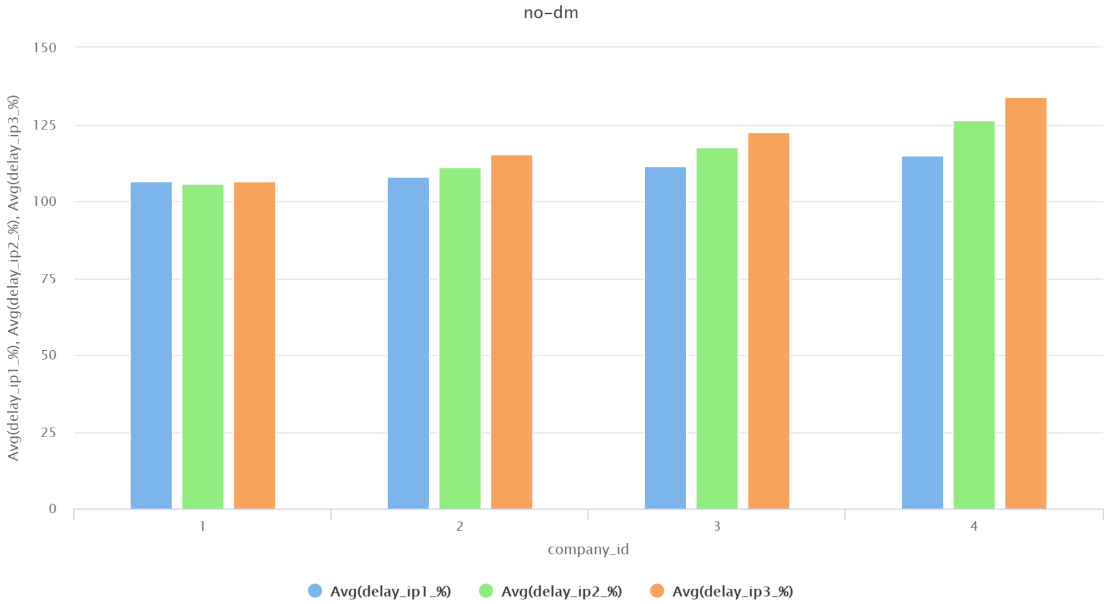

Figure 5.5: Average percentage of delay, per company for each info point, in first experiment.

After the analysis of the results for the experiment, we had to deploy our data, that was processed from our jupyter notebook script, in RapidMiner AutoModel, as we referred in Section 4.5. With this, we intend to see what are the best models to use for our second experiment and then, deploy the model to be accessible through HTTP requests. In Figure 5.6 we can see the results for each model trained. Looking to the results, we believe the best to choose would be the model with the least **Root Mean Squared Error** (RMSE), the average error of predictions where larger errors have a disproportionately large effect, i.e. outliers are penalised more. The lower the error, the better the predictive model. In Figure 5.7 we can see the RMSE for each model. Given this, we opted to use the **Decision Tree**, a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Model	Root Mean Squared Error	Standard Deviation	Total Time	Training Time (1,000 Rows)	Scoring Time (1,000 Rows)
Generalized Linear Model	0.511	± 0.094	1 s	2 s	35 ms
Deep Learning	0.423	± 0.036	2 s	2 s	35 ms
Decision Tree	0.151	± 0.240	299 ms	10 ms	~0 ms
Random Forest	0.775	± 0.104	8 s	161 ms	283 ms
Gradient Boosted Trees	0.457	± 0.232	21 s	369 ms	71 ms

Figure 5.6: Auto ML results for each model.

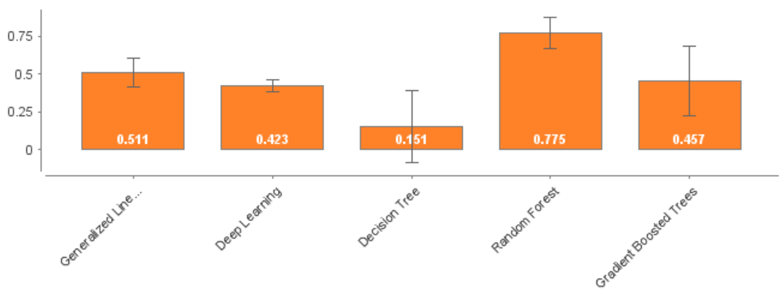

Figure 5.7: Auto ML Root Mean Squared Error results for each model.

With the usage of Decision Tree model, we have the ability to produce two different results: predict the delays or classify. For the second experience we get the prediction of delays, but we could add the capability to trucks, to check during a transfer if they were going to arrive on time. In Figure 5.8 we have decision tree generated, with the predicted delays and in Figure 5.9 we have a example of a decision tree classification.

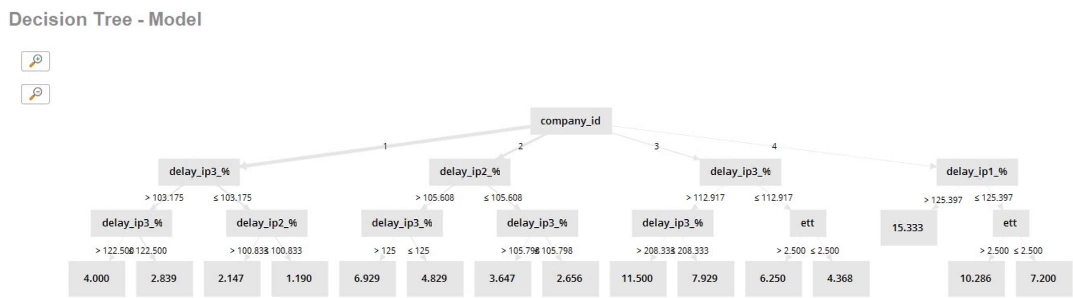

Figure 5.8: Decision tree with the delays that are predicted.

5.4 Simulation with Data Mining

In this experiment, we used the model trained and got the prediction of delays for each truck and company. After that, we ordered them, with the best in first, so warehouses would pick always the

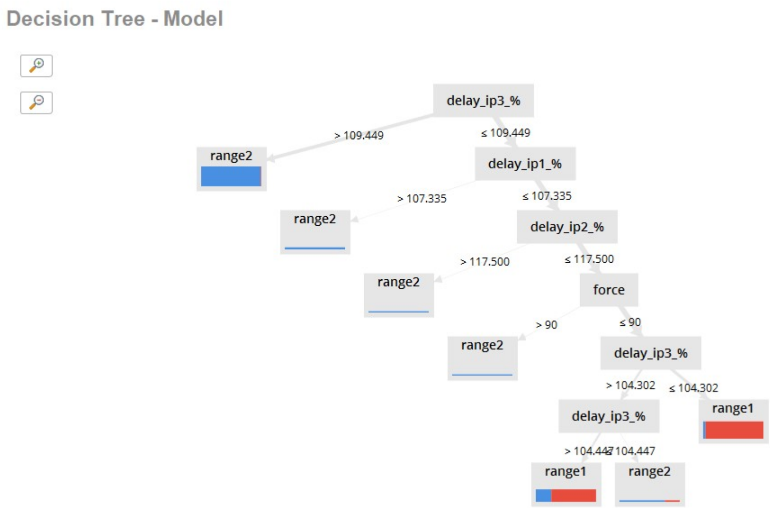

Figure 5.9: Example of a decision tree to classify a delay.

trucks that have better delay and availability. As we expected, the order came with the trucks from company 1 in first, then the trucks form company 2, company 3 and company 4. In order to have more solid results, in this experiment we used the same simulation from the first experiment. The same warehouses, trucks and transfers planned.

So, once we ran the simulation with 60 transfers, ten times, we could notice considerable differences in the results. The first big impact, is that the warehouses stopped choosing trucks from company 4. Since the demand of transfers, did not required more than twelve trucks at the same time, only company 1, 2 and 3 did transfers. Although, even company 3 only made one transfer, the rest was performed by company 1 and 2. In Figure 5.10 we have a chart with the number of transfers made, per company. Taking in consideration the number of delays that were calculated, we got **89 transfers without delay**, meaning that we achieved the double of successful transfers then in the first experiment. Notice that, although it is still a small number, this is due to the configurations of trucks, meaning that, even company 1 and 2 does have a good performance to deliver most of transfers successfully.

Focusing in the info points, we could analyse a big impact in the delays. Figure 5.11 has the median of percentage delay, per truck, for each info point. We can notice that, looking to axis if the Median and comparing with the same chart in the first experience, now the values very between 100 and close to 115, while in the first experiment the values very between close to 100 and 140. We can also see, that the median of delays between info points are closer. In Figure 5.12 we have a better vision in the difference of averages between info points in the same company. This provides more trust when predicting delays during transfers in the early info points, meaning that since the average of delays in all info points is similar, then if a truck is late in first info point, there are more probabilities to be late too, in the following info points.

Having the results from both experiments, we can see that having knowledge about the environment, has advantages in terms of decision-making and adds reliability to decisions made. In

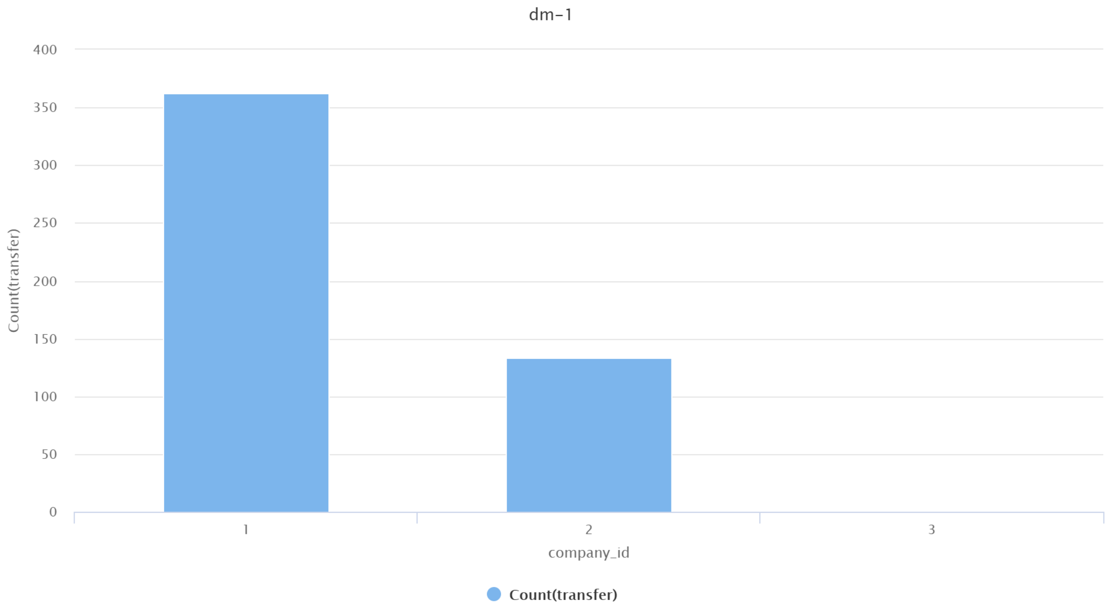

Figure 5.10: Number of transfers made, by company, in second experiment.

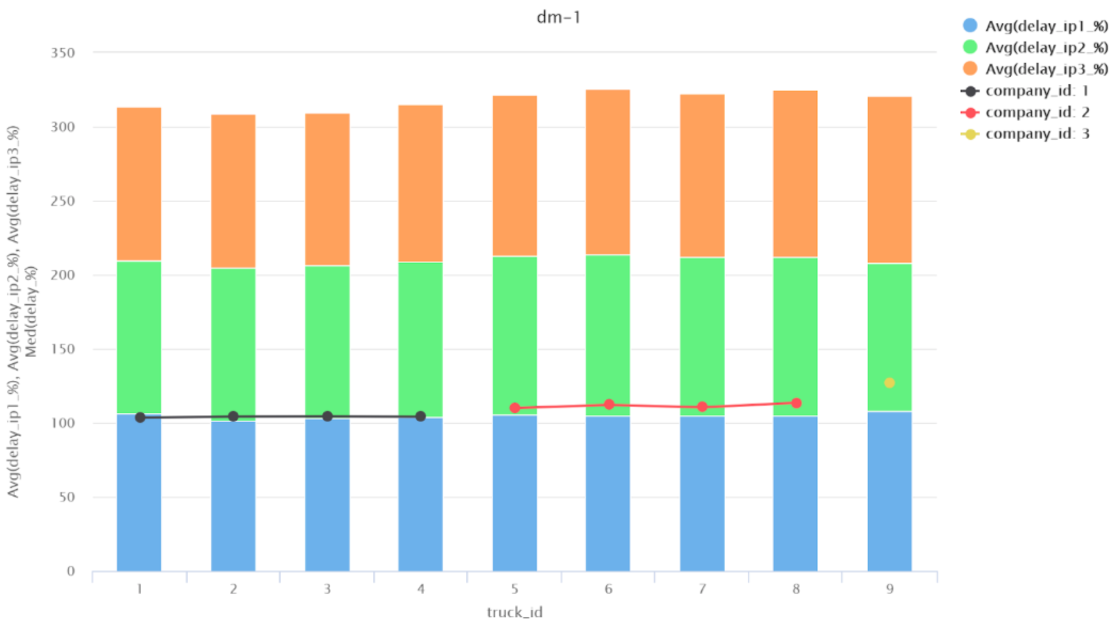

Figure 5.11: Median percentage of delay, per trucks for each info point, in second experiment.

our case, we could conclude, from data collected what are the best partners that warehouses could have, in order to improve the transfers success. Also, we can classify a transfer during the process, in order to prevent late alerts of delays and be able to act in time, so a single transfer would not impact the rest of the plan made, previously. In our case, companies were configured with force and volatility, that impact their movement. In the real scenario, companies may have different configurations, whose values are not visible to warehouses, but with the data that can be collected,

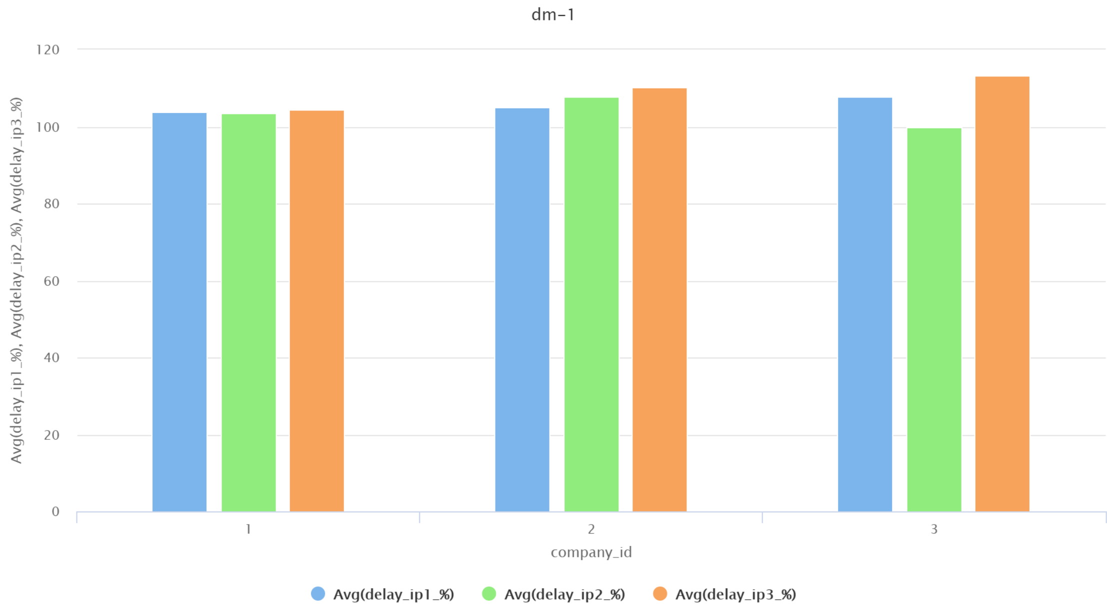

Figure 5.12: Average percentage of delay, per company for each info point, in second experiment.

companies can be classified, just like we did, in our case.

5.5 Challenges and limitations

The fusion of concepts between IoT and Multi-agent systems can bring the interoperability that is needed in cases where devices interact or participate in human actions. Adding intelligence to this systems has the challenge of defining where it should be introduced. A deeper study on Intelligence of Things is required in order to have a formalisation of possible levels of intelligence, regarding the enhancement of the capacities in the devices and in the agents. Also, due to cyber-physical systems becoming more complex, the simulation of the environment has a big weight in order to map the problems and solutions to future systems and new application domains related to logistics.

5.6 Summary

We realised experiments, based in our platform and approaches, in order to evaluate the impact of intelligence in autonomous environments. In this chapter, we analyse the experiments made and results achieved, defining what was assumed and considered for each. In the next chapter, we present our conclusions and future work to realise.

Chapter 6

Conclusions and Future Work

In this chapter we summarise the outcomes of this project as well as the discussion regarding our initial hypothesis, methodology, main contributions and challenges. Finally, we explain some possible future work related to our research.

6.1 Overview

This thesis presents a novel approach for analysing logistic problems through agent-based distributed modelling and simulation. We have discussed the role of the main entities present in the context of the 2PL inventory transfers to understand the behaviours of the two main agents: warehouses and trucks. 2PL warehouse transfers represent the whole process of negotiating the best partner to do the transfer as well as monitoring delays and costs in order to assure delivery. Since the loads are often big volumes of items, delays should be the point to tackle and selecting the best company could benefit from a data mining approach. The more information is shared between logistic company (represented by the trucks), the faster should be the response of the warehouse in case of a disruption. Remediation actions should comprise, for example, interrupting a transfer or observing the delay until the next information point. In this sense, CPS as well as a live monitoring system could help gathering real time information and increase the power of decision. Since we do not usually have that, a set of info points is defined so the trucks could inform (or not) their delivery status as arriving to such point.

We have opted for agent-based simulation because of the distributed nature of the problem, entities are represented as autonomous agents that run some behavioural processes. In that sense, the Agent Process Modelling methodology helped us to take advantage from a standard notation for defining the agent processes while the microservices architecture helped the scalability specification of the system. We have developed a flexibly technological architecture capable of running different simulation configurations and collect the data to be analysed. Given the introduction of

APM and the simulation tool developed, we believe this can be a first step to have a broader adoption of multi-agents in the business side, since both are cloud based and are flexible, allowing to use for different cases and applications.

Our hypotheses comprised the evaluation of simulation regarding the improvement on partner selection and the possibility of leveraging data mining application in order to achieve better selection. In such sense, we have designed two process models, one that do not consider data mining when selecting the partners and the experimental results showed that some trucks were selected because of their price, but not considering their delays. On the other hand, the results of the first experiment were used to train a machine learning model used in a second process to decide over the predicted delay.

The results from running different simulation scenarios showed that the delay between info points is crucial to the overall delay of the transfer, so the best trucks (chosen more frequently) had less variance between info points delays. We also have seen the agents to disregard a whole company when using the data mining approach, since that company (company 4) would never fit to the expected delay thresholds.

Summarising, we can improve the system to be more productive with the information that is provided but also be more accurate. A warehouse transfers problem used as a application domain, allow us to map the solutions provided to other logistics problems. The focus is to reduce disruptions by having a constant motorisation of the system and use it to possibly learn what are the best decision-making. There are already approaches in using agents to coordinate these systems but there is no definition on how can we take more advantage of the capabilities of agents and devices. Thus, the reduction of disruptions and learning from the environment have still a gap to be filled.

Finally, the contributions of this work comprised the development of a distributed simulation tool as well as the design of process models for warehouses and trucks that could be used and improved in any other warehouse transfer scenario. Moreover, we have extended the capabilities of warehouse decisions by using a data mining as a service approach.

6.2 Future work

From an application point of view, there is a wide range of opportunities opened with this work: multiple features could be added to our simulation platform to improve realism, more user capabilities and better tracking of trucks' path or other inter-agent interaction such as the messages exchanged. Moreover, the automation of the simulations pipeline, providing the data results of each simulation could be a nice improvement. Automate the Data Mining process, training the models automatically, instead of manually upload the data to the platform and start the training. Nonetheless, the simulation platform can be improved in order to be used for future researches related to transfers and inventory management, even to be used tracking real-life trucks if considering tracking devices such as connected GPS or RFID attached to the trucks or items.

From a scientific point of view, other research directions to explore include more complex study cases and more variables to affect decision, such as the capacity of the trucks or even the capability of accounting the items that are already in transfer, the possibility to get them back in case of disruption or even the existence of intermediary points (distribution centers) where the items are dropped and other truck/company becomes responsible. More realistic metrics could be assessed with some domain experts from logistics companies and considered in the simulations. Taking in consideration the number of docks in each warehouse and the trucks with cost/performance relation.

The focus of this work was the study of warehouse transfers considering the 2PL aspect: transportation made through logistics partners. There is a lot of space to study other types of transfers and with different resources (different vehicles, vehicles that are owned by the company or different partnerships), etc.. More importantly, there are new ways of improving the manual processes that currently are made and other concepts to add the capability of decision-making. We used Data Mining to add these capabilities, although we did not perform an in-depth analysis of the algorithms and how they worked. We believe improving the data mining part could be a important field for future work, studying the best models and ways to use data about transfers in order to achieve better results and becoming more reliable so we can apply it in a real scenario.

References

[Bellifemine et al., 2007] Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons.

[Bordini et al., 2009] Bordini, R. H., Dastani, M., Dix, J., and Seghrouchni, A. E. F. (2009). *Multi-Agent Programming*. Springer.

[Bordini et al., 2006] Bordini, R. H., Fisher, M., Visser, W., and Wooldridge, M. (2006). Verifying multi-agent programs by model checking. *Autonomous agents and multi-agent systems*, 12(2):239–256.

[Borshchev and Filippov, 2004] Borshchev, A. and Filippov, A. (2004). Anylogic—multi-paradigm simulation for business, engineering and research. In *The 6th IIE annual simulation solutions conference*, volume 150, page 45.

[Bresciani et al., 2004] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236.

[Broy and Schmidt, 2014] Broy, M. and Schmidt, A. (2014). Challenges in engineering cyber-physical systems. *Computer*, 47(2):70–72.

[Byers and Wetterwald, 2015] Byers, C. C. and Wetterwald, P. (2015). Fog computing distributing data and intelligence for resiliency and scale necessary for iot: The internet of things (ubiquity symposium). *Ubiquity*, 2015(November):4:1–4:12.

[Cao, 2009] Cao, L. (2009). *Data mining and multi-agent integration*. Springer Science & Business Media.

[Chiarini, 2012] Chiarini, A. (2012). Business process reengineering. In *From Total Quality Control to Lean Six Sigma*, pages 25–27. Springer.

[Chinosi and Trombetta, 2012] Chinosi, M. and Trombetta, A. (2012). Bpmn: An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.

[Davarzani and Norrman, 2015] Davarzani, H. and Norrman, A. (2015). Toward a relevant agenda for warehousing research: literature review and practitioners' input. *Logistics Research*, 8(1):1.

[Dragoni et al., 2017] Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., and Safina, L. (2017). Microservices: yesterday, today, and tomorrow. In *Present and ulterior software engineering*, pages 195–216. Springer.

[García et al., 2011] García, E., Rodríguez, S., Martín, B., Zato, C., and Pérez, B. (2011). Misia: middleware infrastructure to simulate intelligent agents. In *International Symposium on Distributed Computing and Artificial Intelligence*, pages 107–116. Springer.

[Gehrke et al., 2007] Gehrke, J. D., Ober-Blöbaum, C., et al. (2007). Multiagent-based logistics simulation with plasma. In *GI Jahrestagung (1)*, pages 416–419. Citeseer.

[Giannakis and Louis, 2016] Giannakis, M. and Louis, M. (2016). A multi-agent based system with big data processing for enhanced supply chain agility. *Journal of Enterprise Information Management*, 29(5):706–727.

[Grzybowska and Kovács, 2017] Grzybowska, K. and Kovács, G. (2017). The modelling and design process of coordination mechanisms in the supply chain. *Journal of Applied Logic*, 24:25–38.

[Harukawa et al., 2001] Harukawa, S., Toyama, M., Tokisato, Y., Hirakawa, T., and Iijima, A. (2001). Automated warehouse and automated warehouse management method. US Patent 6,315,513.

[Helsinger et al., 2004] Helsinger, A., Thome, M., and Wright, T. (2004). Cougaar: a scalable, distributed multi-agent architecture. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 2, pages 1910–1917. IEEE.

[Hofmann and Rüsch, 2017] Hofmann, E. and Rüsch, M. (2017). Industry 4.0 and the current status as well as future prospects on logistics. *Computers in Industry*, 89:23–34.

[Jayaram, 2017] Jayaram, A. (2017). An iiot quality global enterprise inventory management model for automation and demand forecasting based on cloud. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 1258–1263. IEEE.

[Jazdi, 2014] Jazdi, N. (2014). Cyber physical systems in the context of industry 4.0. In *Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on*, pages 1–4. IEEE.

[Jeston, 2014] Jeston, J. (2014). *Business process management*. Routledge.

[Kato et al., 2015] Kato, T., Chiba, R., Takahashi, H., and Kinoshita, T. (2015). Agent-oriented cooperation of iot devices towards advanced logistics. In *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, volume 3, pages 223–227. IEEE.

[Luke et al., 2005] Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., and Balan, G. (2005). Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527.

[McFarlane et al., 2016] McFarlane, D., Giannikas, V., and Lu, W. (2016). Intelligent logistics: Involving the customer. *Computers in Industry*, 81:105–115.

[Naim and Gosling, 2011] Naim, M. M. and Gosling, J. (2011). On leanness, agility and leagile supply chains. *International Journal of Production Economics*, 131(1):342–354.

[Pavón et al., 2006] Pavón, J., Gómez-Sanz, J., and Fuentes, R. (2006). Model driven development of multi-agent systems. In *European Conference on Model Driven Architecture-Foundations and Applications*, pages 284–298. Springer.

[Pokahr et al., 2013] Pokahr, A., Braubach, L., and Jander, K. (2013). The jadex project: Programming model. In *Multiagent Systems and Applications*, pages 21–53. Springer.

[Railsback et al., 2006] Railsback, S. F., Lytinen, S. L., and Jackson, S. K. (2006). Agent-based simulation platforms: Review and development recommendations. *Simulation*, 82(9):609–623.

[Rúbio et al., 2019] Rúbio, T. R. P. M., Lopes Cardoso, H., and Oliveira, E. (2019). Agent process modelling. In Demazeau, Y., Matson, E., Corchado, J. M., and De la Prieta, F., editors, *Advances in Practical Applications of Survivable Agents and Multi-Agent Systems: The PAAMS Collection*, pages 277–280, Cham. Springer International Publishing.

[Russell et al., 1995] Russell, S., Norvig, P., and Intelligence, A. (1995). A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25.

[Sanchez-Rodrigues et al., 2010] Sanchez-Rodrigues, V., Potter, A., and Naim, M. M. (2010). Evaluating the causes of uncertainty in logistics operations. *The International Journal of Logistics Management*, 21(1):45–64.

[Sarkar et al., 2014] Sarkar, C., Nambi, S. N. A. U., Prasad, R. V., and Rahim, A. (2014). A scalable distributed architecture towards unifying iot applications. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 508–513.

[Scholz-Reiter and Stickel, 2012] Scholz-Reiter, B. and Stickel, E. (2012). *Business process modelling*. Springer Science & Business Media.

[Sik Jeong and Hong, 2007] Sik Jeong, J. and Hong, P. (2007). Customer orientation and performance outcomes in supply chain management. *Journal of Enterprise Information Management*, 20(5):578–594.

[Tejedor et al., 2016] Tejedor, E., Lamanna, M., Moscicki, J., Mascetti, L., Piparo, D., and Mato, P. (2016). Data mining as a service (dmaas). In *J. Phys. Conf. Ser.*, volume 762, page 012039.

[Thönes, 2015] Thönes, J. (2015). Microservices. *IEEE software*, 32(1):116–116.

[Tisue and Wilensky, 2004] Tisue, S. and Wilensky, U. (2004). Netlogo: A simple environment for modeling complexity. In *International conference on complex systems*, volume 21, pages 16–21. Boston, MA.

[Trappey et al., 2017] Trappey, A. J., Trappey, C. V., Fan, C.-Y., Hsu, A. P., Li, X.-K., and Lee, I. J. (2017). Iot patent roadmap for smart logistic service provision in the context of industry 4.0. *Journal of the Chinese Institute of Engineers*, 40(7):593–602.

[White, 2004] White, S. A. (2004). Introduction to bpmn. *Ibm Cooperation*, 2(0):0.

[Wooldridge, 2009] Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

[Xiong et al., 2015] Xiong, G., Nyberg, T. R., Dong, X., and Shang, X. (2015). Intelligent technologies and systems of material management. In *Intelligent Techniques in Engineering Management*, pages 295–330. Springer.

Appendix A

Agents Processes

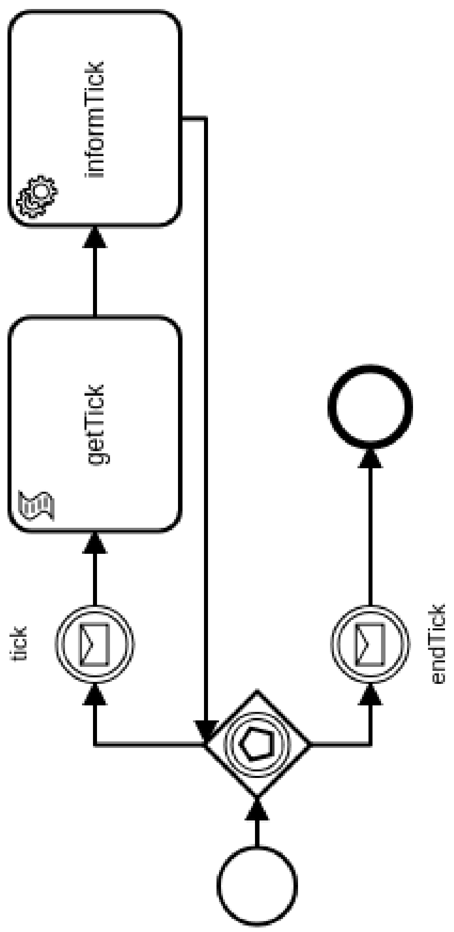

Figure A.1: Informer agent process model.

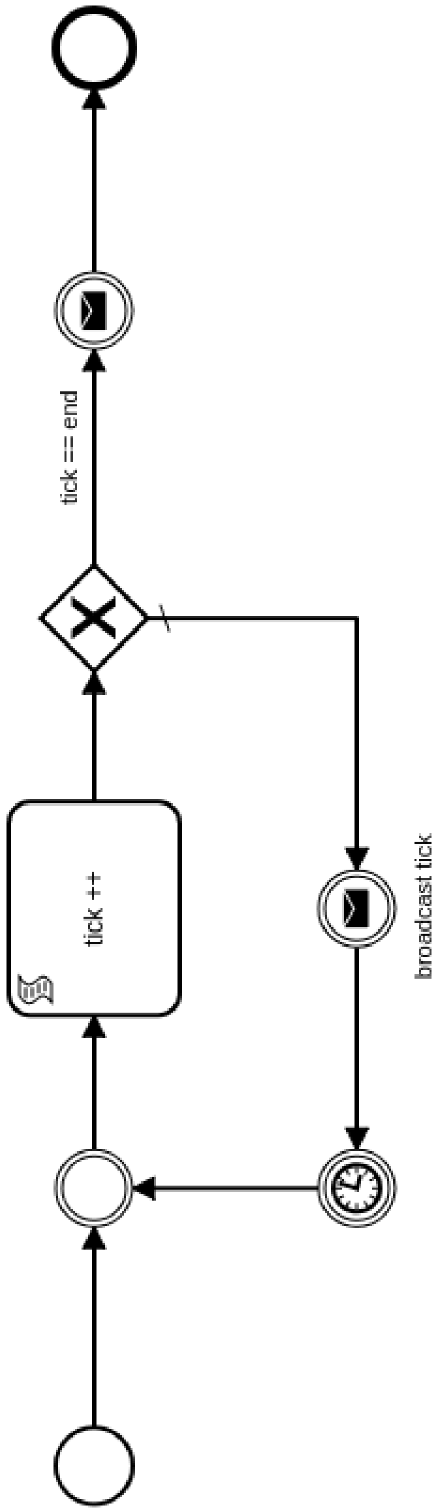

Figure A.2: Ticker agent process model.

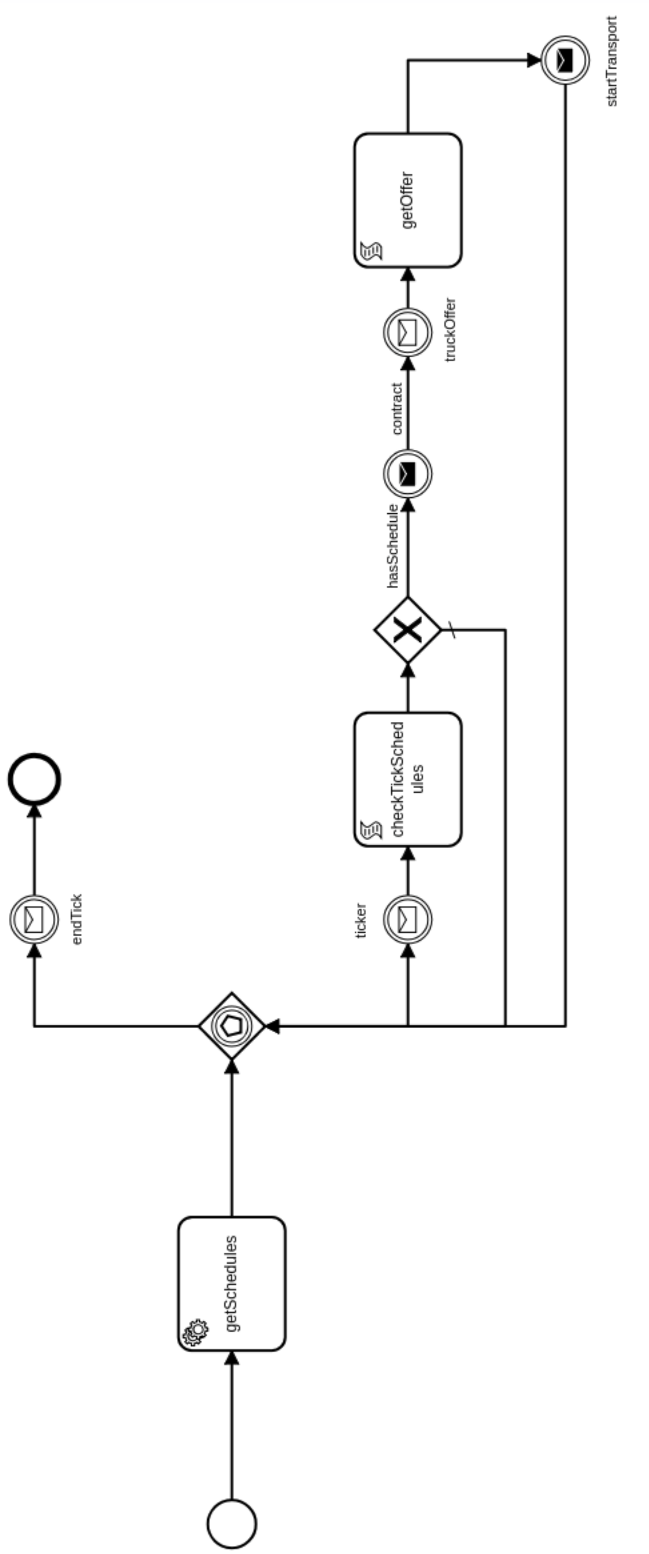

Figure A.3: Warehouse agent process model for simulation 1.

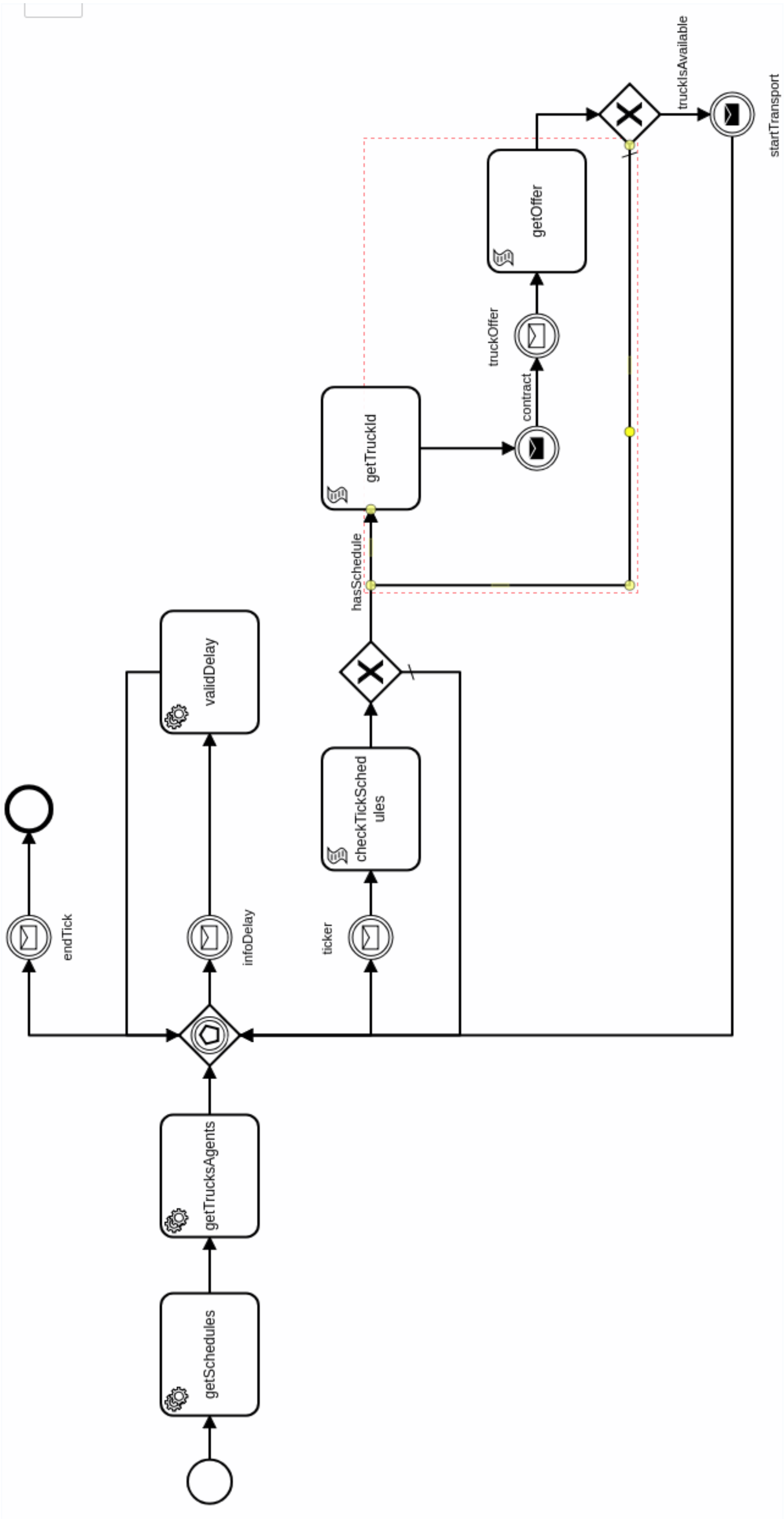

Figure A.4: Warehouse agent process model for simulation 2.

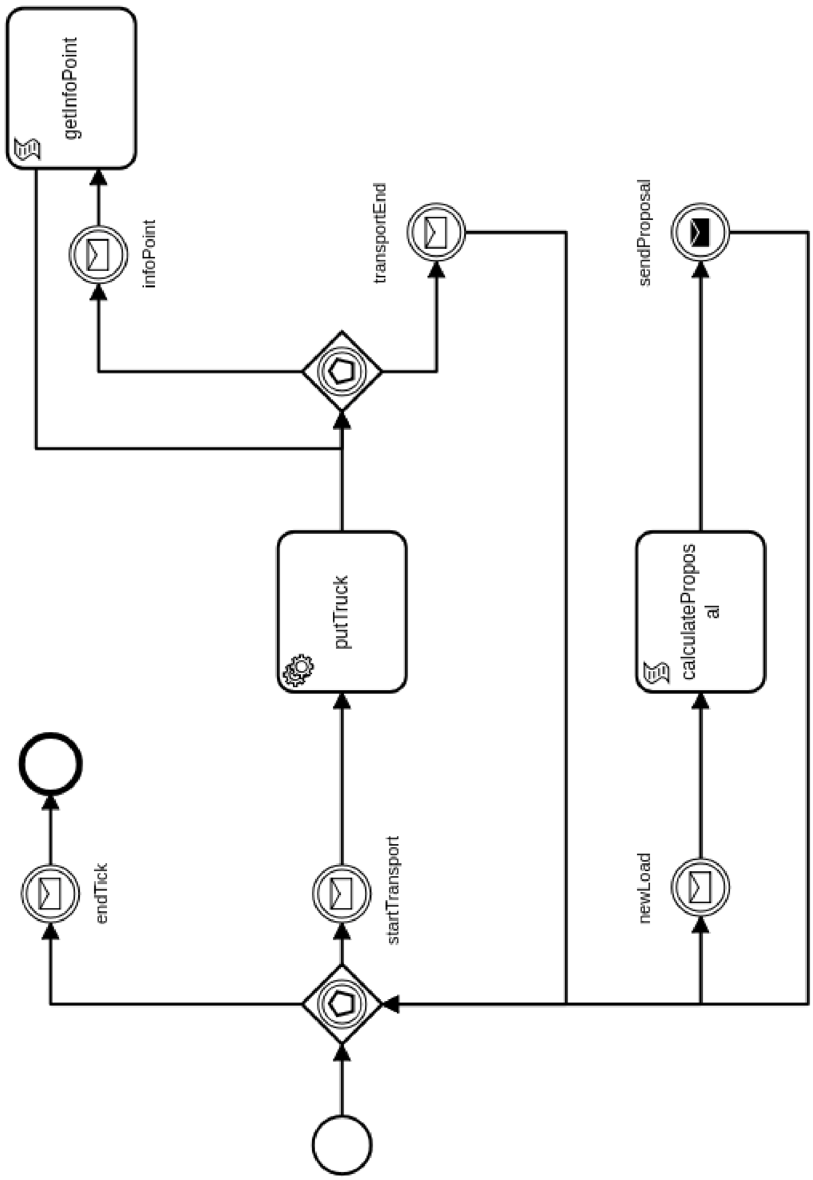

Figure A.5: Truck agent process model for simulation 1.

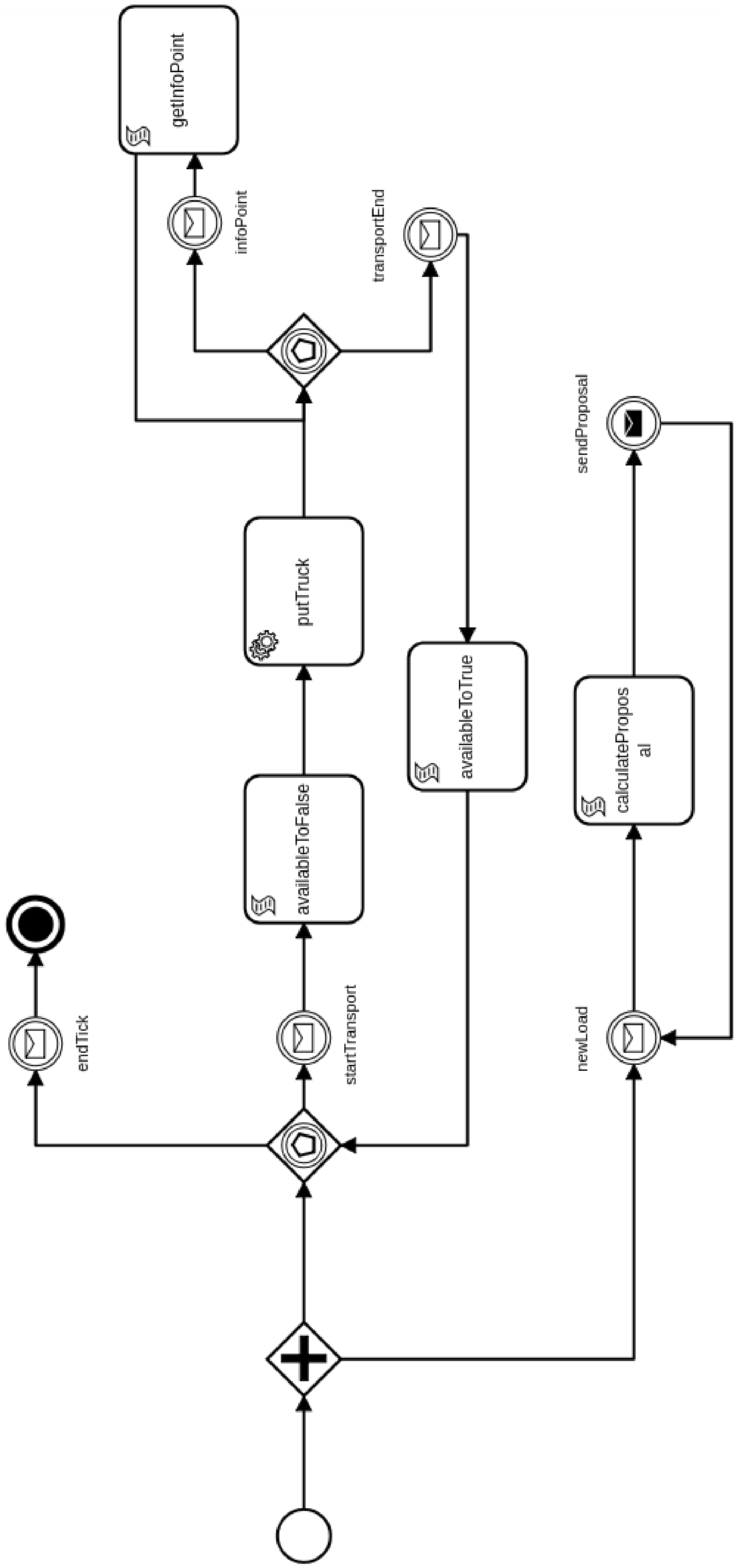

Figure A.6: Truck agent process model for simulation 2.