POLITECNICO DI TORINO

Master's Degree

in Engineering and Management

Master of Science Thesis

Simulation of regional logistics systems with Agent-Based Modelling: a Dubai case study

Supervisor

Prof. Alberto De Marco

Candidate

Alessandro Fabbri

Co-Supervisor

Prof. Hussein Fakhry

A.A 2019/2020

Table of Contents

List of Figures

List of Code Snippets

Introduction

We live in a complex world.
An ever more complex world that seems to be more volatile and uncertain than ever before.

Nothing is fixed and it's impossible to predict the future. No one knows what's going to happen, and there's definitely not a crystal-clear path along the way.

Our world is getting more complex day by day. Technology is advancing at a speed that even scares some of us. What was true yesterday doesn't work today anymore and we now live in a new world with new rules.

Digitalization, big data, artificial intelligence, robotization, globalization, terrorism, financial crises, climate change, pandemics and global shifts in power all contribute in making the world around us more uncertain, heightening risks for organizations and threatening their ability to achieve their business objectives.

Change is happening at a massive rate, faster and often, carrying by a multipored effect. If we change one thing, we may affect a whole lot of other things. If we reorganize an organization, we'll affect culture, leadership and how innovation is driven. If we change how innovation is driven, we may affect organization direction, purpose and culture again.
It's all interconnected – making it all complex.

However, due to technological enhancements, we have also become much more capable of dealing with this increased volatility and complexity. Computing power and speed, the omnipresence of information, the development of complex algorithms and the digital connectedness of the world, enable us to make dramatically deeper analyses and respond much faster than even a decade ago. Thus, even though volatility and complexity probably have increased, our increased ability to deal with them does not directly mean that this has made doing business also more difficult.

Everybody agrees that decision making should be based more on data and analysis than intuition and gut feelings. But there are two problems here: data may be difficult to gather, and data tells about the past but gives no indication about the future. So how to create and use data that will help predict what is going to happen?

Modelling and simulation, techniques used to support scenario analysis and optimal decision making, can be used for undertaking opportunities or defensive pessimism, helping organizations to take practical steps to improve their ability to deal with disruption and to provide tools to prepare and survive.
The use of software models makes it possible to avoids actual experimentation, which can be costly and time-consuming. As such, they can facilitate the understanding of a system's behavior, without actually testing it in the real world.

In the field of study of complex systems, Agent-Based Modelling has made a significant impact in the recent years. It focuses on the individual components of a system, in contrast to both the more abstract System Dynamics approach and the process-focused Discrete Event. With Agent-Based Modelling, agents must be identified and their behavior defined. Connections between them are established, environmental variables set, and simulations run. The global dynamics of the system then emerge from the interactions of many individual agents, whose behavior and characteristics are heterogeneous.

The objective of the thesis is to provide a detailed analysis of the Dubai Logistics Corridor, leveraging on advantages provided by Agent-Based Modelling. In specific, the paradigm enables deepening into individual resources and is used with the aim of studying bottlenecks, investments allocation and policy implications. The resulting frameworks then aims to be used as a support tool for decision making in regional logistics systems. As a secondary implication, the thesis also examines the possibility of expanding an existing System Dynamics model with Agent-Based and gives recommendations about when the latter is most suitable.

This thesis originated thanks to the mutual contribution of Dr. Alberto De Marco (Polytechnic of Turin – College of Management Engineering) and Dr. Hussein Fakhry (Zayed University – College of Technological Innovation), building up on the previous thesis *System Dynamics approach to assess the investments impact of the Dubai Logistics Corridor* (Postorino, 2018).

The work lasted approximately eight months and was carried out through different phases. First of all, the study of the topic of complex systems and the methodology to deal with this kind of problems. In fact, management of complex system is not a teaching subject in the Master course of Engineering & Management, that instead is only taught at doctoral level. Hence, a study from scratch of topics concerning complex system was necessary to obtain a more accurate view on the subject.

Having acquired the basic knowledges, the second part was carried out throughout a more specific study of academic papers to assess the effective use of Agent-Based Modelling to solve Logistics problems and get valuable insights on individual behavior, interactions among agents and relevant indicators, with a focus on management and strategic decision making. Research was also carried out on papers and documentation about the integration between System Dynamics and Agent-Based Modelling, the newly available opportunities and the possibility to create a multi-paradigm model.

Lastly, a period of approximately three months was spent in Dubai thanks to the hospitality of Zayed University, in order to study the local processes and build a consistent model to be validated.

The thesis is structured as follows:

Chapter 2 presents the theory underlying the approaches to complex systems and simulation modelling. A particular emphasis is drawn on Agent-Based Modelling, its potentialities as well as its drawbacks according to the state of art of the field.

Chapter 3 is dedicated to the Emirate of Dubai, the evolution of the logistics sector and the opportunities enabling country development. In addition, a general overview of the Dubai Logistic Corridor is given, described in detail each of its components.

Chapter 4 constitutes the core element of the present work and is dedicated to the modelling part. After drawing the focus on the technique to transform a System Dynamics model into an Agent-Based, issues and available solutions are highlighted. Subsequently, the in-depth explanation of the newly developed model is presented and in particular how it is created starting from a preliminary System Dynamics model.

Chapter 5 shows the simulation of the newly developed model, highlighting its functioning and features. Moreover, the model is tested in comparison with the System Dynamics model in order to be validated.

Chapter 6 contains Conclusions, which include the results, recommendations on which approach to use, limitations and available opportunities for future development.

Chapter 2

Simulation Modelling

2.1 Why Simulation

People frequently underestimate how often they use models. In fact, mental models are built by people every day, in order to understand and analyze how things work in the real word. However, when tasks become more complicated, digital models become by far superior to mental models, and this is why building models using computer software tools has become standard in business and engineering processes.

The most popular digital modeling approach is obviously to use spreadsheets. Spreadsheet models are commonly used for financial analysis, investment research and business planning and using user-defined formulas, spreadsheets can perform mathematical, statistical and organizational transformations on sets of data. Spreadsheets also help to present data in an organized way that can help inform future planning decisions. However, they struggle to handle complex problems since they lack the possibility of dealing with easiness topics like feedback, delay, interdependencies and non-linearity. Simply, when dealing with complex systems, they fall behind simulation models, that may be considered as a set of rules (equations, flowcharts, state machines, cellular automata) that define how the system being modeled will change in the future, given its present state (Borshchev and Filippov, 2004).

Why though, do people create models at all? First of all, to experiment without having to use real-life objects, since working directly with them is often too expensive, dangerous or even impossible while, on the contrary, the virtual modeling world is completely risk-free. They enable to analyze systems and find solutions where traditional methods fail largely because they can take into account more complex interdependencies and the behavior of the system over time, even if it appears counter intuitive or controlled by a large number of variables. Given an appropriate level of abstraction, a simulation model is a more straightforward process than analytical modeling, since it is typically scalable, modular and allows to introduce measurements and statistical analysis at any time. Moreover, simulation considers the randomness and non-obvious influences which characterize real-life problems. Managers, researchers, and engineers can create and test models of various system designs and answer hundreds of “what-if” questions, all by virtually experimenting in a safe environment.

Simulation modeling stands apart within the range of modeling tools and technologies because it allows the introduction of dynamics into models. Every simulation model evolves over time, discretely or continuously changing its state according to a set of given rules. Simulation modeling is a very powerful instrument that is widely used for solving numerous business challenges.

This is mostly because conceptually it is practical and easy to understand: there is no need guess the system a priori, just run a simulation model and see the result emerge. Lastly, animation is a significant advantage that simulation modeling can provide over static spreadsheet models. When people see a dynamic dashboard or an animated model, similar to a movie or a computer game, where a system can be seen working as specified, it is easier to visually validate the model and understand better its results (Lyle Wallis, 2016).

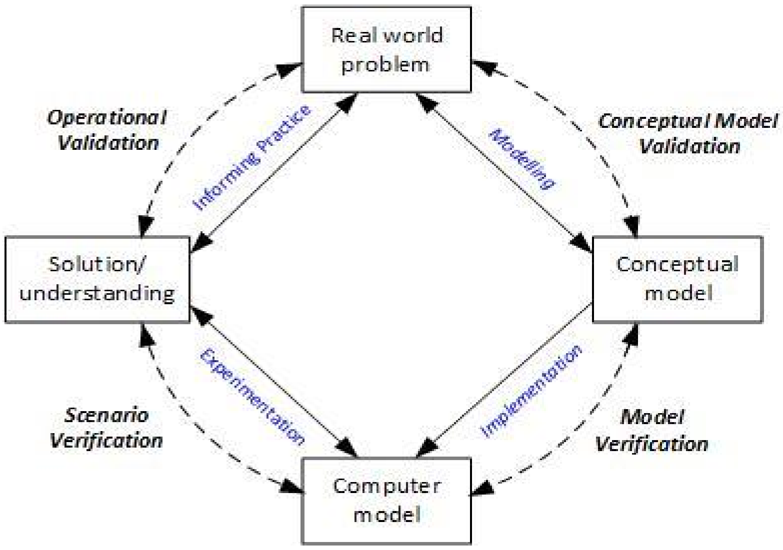


Figure 1 - Stages of a simulation study

As shown in the figure proposed by Brooks and Robinson (2000), the typical lifecycle of a simulation modelling is based of five iterative stages: Stage 1 - Real World Problem Definition, Stage 2 - Conceptual Modelling, Stage 3 - Computer Modelling, Stage 4 - Verification and Validation, Stage 5 - Solution and Understanding.

In this chapter, different approaches to simulation modeling will be briefly covered and multimethod (also called hybrid) modelling defined. Moving from theory to practice, AnyLogic software will be introduced together with an in-depth look at Agent-Based Modelling. Lastly, academic literature will be reviewed with a dual focus: the theories underlying Software Agents and the use of Agent-Based Modelling to solve problems in the field of logistics.

2.2 Main Paradigms

Method, in simulation modeling, means a general framework for mapping a real-world system to its model. A method suggests a type of language, or "terms and conditions" for model building. In modern simulation modeling for complex systems there are three main methods, each of them serving a particular level of abstraction.

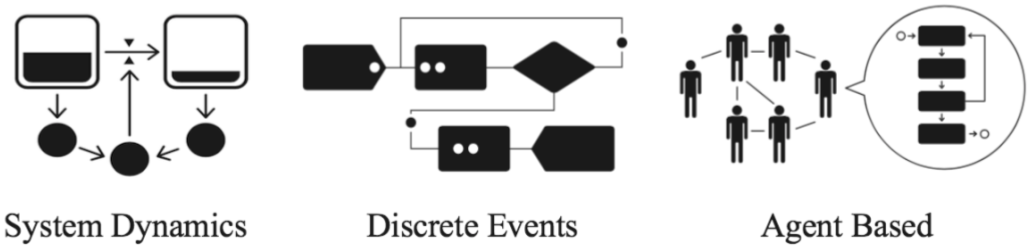


Figure 2 - Main Paradigms

- System Dynamics is the oldest simulation approach, with roots in work done by Massachusetts Institute of Technology’s professor Jay Forrester in the 1950s. From the very beginning, System Dynamics stayed focused on the management and organizational layers of business. This approach operates at a high abstraction level and is mostly used for strategic modeling. The basic System Dynamics element is the stock and flow diagram. A stock is a digital representation of something that is modeled, such as a number of people, a set of requirements or a number of products. Any process in System Dynamics is modeled as a flow between stocks. System Dynamics is most commonly used for modeling strategic management, marketing and macroeconomic issues, ecological and social systems. It makes extensive use of the notion of feedback and is based on differential equation: at every point in time, flows determine the passage of a determined quantity of a variable from one stock to another.

- Discrete Event is based on a process-centric approach, thus supporting medium and medium-low abstraction. In October 1961, IBM engineer

Geoffrey Gordon introduced the first version of GPSS (General Purpose Simulation System, originally Gordon's Programmable Simulation System), which is considered to be the first software implementation of discrete event modeling. Hence, Discrete event modeling is almost as old as System Dynamics but even in our days discrete event modeling is supported by many software tools. The main idea of discrete event simulation is to consider a system as a process, i.e. a sequence of operations being performed across entities. A model is specified graphically as a process flowchart, where blocks represent operations. The flowchart usually begins with "source" blocks, that generate entities and inject them into the process, and ends with "sink" blocks that remove entities from the model. This type of diagram is familiar to the business world as a process diagram and is widely used to describe business processes. Discrete event simulation modeling is widely used in the manufacturing, service systems, and healthcare fields.

- Agent-Based Modelling is the most recent of the major simulation methods. From 1990-2000, Agent-Based Modelling stayed a purely academic topic, but the 21st century, with its boom in computer processing capability, made Agent-Based modelling commercially applicable for solving high-scale business tasks. Moreover, compared with the other modeling techniques, it is now showing the fastest growth in use. Agent-Based Modelling uses a bottom up approach, where the system is described as interacting objects with their own behaviors. System behavior emerges as a summary of the individual actions of agents. Agent-based models can vary, from very detailed, where agents represent physical objects, to highly abstract, where agents are competing projects or assets. There are some specific fields where Agent-Based Modelling is particularly effective to solve problems, such as population, pedestrian, road traffic, and epidemiology modeling. But in fact, Agent-Based Modelling is used for modeling almost everything, from the markets, to supply chains and logistics, in cases where we need to focus on individual objects and describe their local behavior and interactions.

2.2.1 Abstraction Level

In one way or another, building a model requires a level of simplification. Can a broad view be taken, or should fine details be captured? It all depends on the system being modeled and the problem in need of a solution.

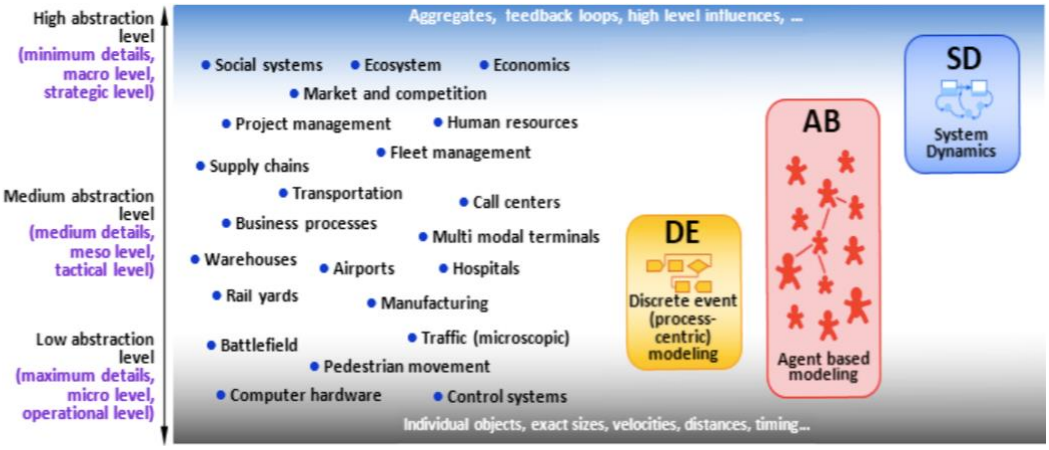


Figure 3 - Abstraction Level

The figure above shows a number of simulation applications, all sorted by the abstraction level of the corresponding model. At the bottom are the physical level models that use highly detailed representations of real-world objects. At this level, the focus is on physical interactions, dimensions, velocities, distances, timings. On the contrary, models at the top are highly abstract, and they typically use aggregates rather than individual objects, thus allowing to understand relationships without requiring modelling intermediate steps. Lastly, in the middle are Agent-Based models, which can vary from very detailed models where agents represent physical objects to the highly abstract models where agents represent whole systems.
Choosing the right abstraction level is critical to the model success and occasionally reconsider the abstraction level could be normal. In most cases, modelers start at a

high abstraction level and add details as soon as they need them, neglecting whatever remains below the desired level of abstraction.

Using a single method, it can be difficult to model at the appropriate level of abstraction. It may be possible to model the actions of autonomous entities via System Dynamics, but unnecessary when Agent-Based tools avoid the need for additional abstractions and assumptions. Similarly, discrete methods are inefficient for modeling continuous variables when System Dynamics methods are available. Most real-world cases are complex, and it is convenient to describe different parts of a system with different methods. The ability to capture business systems with their real complexity and interactions can be seriously limited using only one method.

Sometimes, in particular cases, it might be helpful to model different part of the simulation system with different methods; this is usually referred as multimethod model or hybrid model (Borshchev, 2013). The idea of multimethod modeling is simple: to seamlessly integrate different methods of modeling and simulation to overcome the drawbacks of individual approaches and get the most from each one. Combining different methods leads to efficient and manageable models without using workarounds. Historically, the first commercial software to support this kind of integration was AnyLogic.

2.3 Agent-Based Modelling

Compared to System Dynamics and Discrete Event, Agent-Based modelling is a relatively new method, mainly due to the fact that remained largely an academic topic until simulation practitioners began using it in early 2000s.
It was triggered by three main elements: First, a desire to gain deeper insights into systems that traditional approaches don't capture well. Second, advance in modeling technology made possible by computer science, such as object-oriented modeling, UML (Unified Modelling Language) and Statecharts. Last, the rapid growth of CPU power and memory due to Agent-Based models being more demanding than System Dynamics and discrete events models.
With the appearance of StarLogo in 1990, Swarm and NetLogo in the mid-1990s and RePast and AnyLogic in 2000 as well as some custom-designed code (Java or C++ libraries from academic community), modelling software became widely available and the range of domains that Agent-Based Modelling was applied to grew. Later on, AnyLogic evolved into an industrial grade solution with many applications, including markets and competition, healthcare, manufacturing, retail, social and ecoSystem Dynamics, defense, pedestrian dynamics and road traffic, aerospace, supply chains and logistics, business processes, project management. Nowadays, even if Agent-Based Modelling is integrated in others commercial solutions for practitioners as Rockwell Automation's Arena and Simul8 Professional, AnyLogic remains the leading solutions in industries like consulting, banking, automotive, telecom, transportation, pharmaceutical, government and defense.

AnyLogic historically was developed not by simulation modelers but by people with background in distributed systems, concurrency theory and computer science. Therefore, none of the classical simulation modeling paradigms was used as a foundation. Instead, approaches and languages designed to handle complexity and adopted in the software engineering world were implemented. It turned out that stock-and-flow diagrams and flowcharts are naturally expressed in the object-oriented core language of AnyLogic, and there is a lot of value added even for the classical modeling styles: compact structured representation, flexible data definition, etc. But the most exciting thing is the ability to rapidly compose industrial-strength Agent-Based models within the same visual environment. AnyLogic supports ready-to-use constructs for defining agent behavior, communication, environment model, and has rich visualization capabilities. (Borshchev, Filippov, 2004)

2.3.1 Modelling in AnyLogic

Academics still debate which properties an object should have to be an "agent": proactive and reactive qualities, a spatial awareness, an ability to learn, social ability, "intellect" etc. (Schieritz and Milling, 2003). In opposition to this debate, the practitioner's community faded away from strict definition inherited from computer science and just took advantage of these powerful instruments to apply them to real-world problems. In fact, in applied Agent-Based Modelling all kind of agents can be found: some communicate while others live in total isolation, some live in a space while others live without a space, and some learn and adapt while others never change their behavior patterns. Everything can be an agent, not just people: a vehicle, a piece of equipment, a project, an organization, or even an investment. In fact, agents don't need to be necessarily cognitive or active (simple goals), but can also be passive objects such as a piece of equipment; even if it seems counter intuitive, in applied Agent-Based Modelling it could be a very effective way to associate more properties to the object, such as costs, maintenance, replaced schedules and breakdown events. At last, there are also Agent-Based models in which agents don't interact at all between them, as the case of health economics where individual dynamics depend only on personal parameters and the environment.

From an architectural point of view, Agents are main building blocks of AnyLogic model but practically speaking they don't differ much from simple classes of object-oriented programming. In fact, AnyLogic is based on Java programming

language but an intuitive user interface makes it possible to avoid writing long blocks of codes and put the essential information in pre-formatted boxes. There is no need for professional software development skills to create a good model, but small actions like parameter initialization, sending messages between agents or the movement of agents are defined by adding a few lines of code scripts.
A typical AnyLogic Agent-Based model would have at least two agent types: *Main* type for a top-level object that represents the environment and another type of agent that would be contained inside *Main*.

From the standpoint of practical applications, Agent-Based Modelling can be defined as an individual-centric approach to model design. When designing an Agent-Based model, the modeler identifies the agents and their attributes, defines their behavior using statecharts and events, puts them in an environment, establishes any necessary connections, and runs the simulation. The system-level behavior then emerges as a result of the interactions of the many individual behaviors. Statecharts are one of the most powerful and commonly used Agent-Based Modelling tools for modelers, allowing users to define an object's behavior as a sequence of states. On the other hand, events allow actions to be executed in a model depending on time-conditions or conditional triggers. Since each agent in the environment is a separate unit that operates independently, communications with other agents take place by sending messages.

Statecharts are the fundamental tool used to model agents in AnyLogic, since they are one of the most advanced constructs used for describing time-based and condition-based behavior. Exactly as previously described for agents, statecharts are an adaptation from UML state machines that, however, relax the strict definition of computer science to rather focus on the practical applications on a wider spectrum.

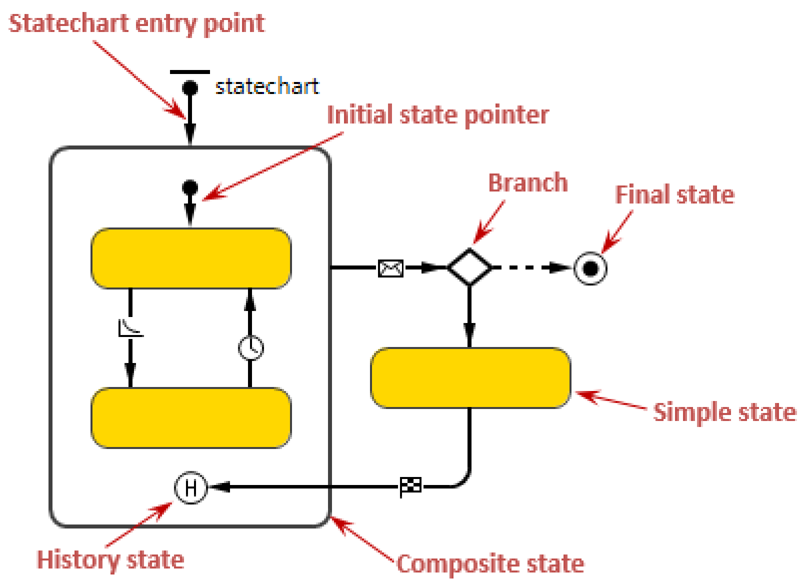


Figure 4 - AnyLogic Statechart

The statechart entry point marks the initial state and defines the name of the statechart. In general, an agent may have multiple statecharts, but usually just one is required.

The states in the statechart are alternative, so an agent can only be in one simple state at a time; however, simple states can be included in composite states, which can have a hierarchical structure and be included in other higher-level composite states. Entering in a composite state requires the use of an initial state pointer, since agent must always be in a single state; history state can also be used to point to the most recently visited state inside a composite one.

When entering or exiting states, either simple or composite, action can also be performed, unless a safeguard known as guard-after-trigger is evaluated false.

Transitions define how the agent changes its states, hence making a new set of transition become active. Transition can be triggered by various types of event occurred, as listed below:

- Timeout: After a specified time, interval counted from the moment the statechart enters the direct state. Commonly used to model delays.
- Rate: Acts in the same as timeout, but the time interval is drawn from an exponential distribution parameterized with the given rate.

- Condition: Monitor a given condition and react when it becomes true. The condition is an arbitrary boolean expression and may depend on the states of any agents in the whole model with continuous as well as discrete dynamics
- Message: Upon reception of a message by the statechart or by the agent from outside. Message template can be specified in the transition properties and only a message matching criterion will trigger transition.
- Arrival: used only for a moving agent, when it arrives at the specified destination

There is a special type of transition called internal transition, in which both start and end points of the transition lie on the border of this state. Since an internal transition does not exit the enclosing state, neither exit nor entry actions are executed when the transition is taken. Moreover, the current simple state within the state is not exited too. Therefore, the internal transition is very useful for implementing simple background jobs, which should not interrupt the main activity of the composite state.

Branches can be used together with transition that have more than one destination states, according to a condition to be evaluated. Control never stays in a branch state and it always passes through. Therefore, triggers cannot be specified for transitions exiting branches and the existence of a default transition, taken if all others are false, is always required. Finally, final state represents a termination point of a statechart and it can optionally be used to remove an agent from the environment by calling *main.remove_agent(this)*.

An event is the simplest way to schedule some action in the model; thus, events are commonly used to model delays, timeouts and condition check. Events can be activated by three types of trigger:

- Timeout: It is used when in need to schedule an action at some particular moment of time or at a date. Timeout triggered event has even more features: you can specify that it expires either once, cyclically or is fully controlled by the user (using event methods)

- Condition: It is used when you want to monitor a certain condition and execute an action when this condition becomes true.
- Rate. It is used to model a stream of independent events, following a Poisson distribution. It is frequently needed to model arrivals following queueing theory

There is another type of event defined with another model element - dynamic event - used to schedule any number of concurrent and independent events. A model can have several instances of the same dynamic event scheduled concurrently, and they can be initialized by data that are stored in the event's parameters.

Given the fact that agents are independent objects, the mechanism to communicate information is made available thanks to messages. In this way, an agent can send a message to an individual or to a group of other agents, no matter if they are agents of the same type or not. A message can be an object of any type of complexity, including plain text, integers, structures with multiple fields or even a reference to an object. Specific functions are used to implement this mechanism, such as *sendToAll(msg)* for agents in the same population, *sendToRandom(msg)* for one randomly chosen agent in same population and *send(msg,agent)* for generic messages where the selected recipients are passed as argument and can also be filtered.

2.4 Literature Review: Software Agents

As any human innovation, Agent-Based thinking did not develop in an empty space. There had been related discussion in several areas, but the most important fields of research for the birth of Agent-Based thinking were artificial intelligence, object-oriented programming and human-computer interface design (Jennings, Sycara & Woolridge, 1998). The emergence of Agent-Based design in artificial intelligence took place in 1985, named after “autonomous agent research”, “behavior-based research”, “bottom-up AI” or “animat approach”. These names all describe the characteristics of new kind of modelling: it is an approach, where systems are designed bottom-up, and thus the ‘autonomous’ components, namely agents, act according to their internal schema instead of external commands. This produces emergent behavior, which to some extent resembles life, and thus is closer to natural intelligence than the behavior the previous approaches produce (Vuori, 2011).

Thus, like the ability to form thoughts is an emergent property of the interaction between different brain cells and consciousness is an emergent property of the interaction of thoughts, emergent properties take place on the system level and are not a property of a single agent. The existence of emergent properties is the motivation also for Agent-Based Modelling and simulation. Often Agent-Based Modelling is aimed at revealing some emergent properties and the process leading to them. The simulation of interaction between agents is often “surprising, because it can be hard to anticipate the full consequences of even simple forms of interaction” (Axelrod, 1997).

The important aspects of these systems are complexity and decentralization. It is slightly difficult to find a clear difference between Agent-Based software design and more general object-oriented design. The notion of agent, however, is controversial even inside the restricted community of computer scientists dealing with research on agent models and technologies. Franklin and Graesser conclude that “An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" (Franklin and Graesser, 1996). This definition separates autonomous agents from other pieces of code, because an object does not have e.g. an own agenda. It is also stated that agents differ from other programs because they are autonomous, flexible, are considered to have their own thread of control and exhibit a control over their behavior. One of the most commonly adopted definition of an agent (Wooldridge and Jennings 1995) specifies a set of properties that must characterize an entity to effectively call it an agent, and in particular autonomy (the possibility to operate without intervention by humans, and a certain degree of control over its own state), social ability (the possibility to interact employing some kind of agent communication language), reactivity (the possibility to perceive an environment in which it is situated and respond to perceived changes) and pro-activeness (the possibility to take the initiative, starting some activity according to internal goals rather than as a reaction to an external stimulus). The authors consider this a weak definition of agency, even though it is generally already too restrictive to characterize as agents most of the entities populating Agent-Based models for simulation in different fields (Bandini, Mazzoni, Vizzari, 2009).

Although, even if the distance between the context of research on intelligent agents and Agent-Based simulation cannot be neglected, in practical application those strict definitions are often relaxed to just focused on the resulting behavior of relatively simple agents. From a practical modelling standpoint, based on how and why agent-models are actually built and described in applications, Macal and North (2011) consider agents to have certain essential characteristics:

- An agent is a self-contained, modular, and uniquely identifiable individual. The modularity requirement implies that an agent has a boundary. One can easily determine whether something is part of an agent, is not part of an agent, or is a shared attribute. Agents have attributes that allow the agents to be distinguished from and recognized by other agents.
- An agent is autonomous and self-directed. An agent can function independently in its environment and in its interactions with other agents, at least over a limited range of situations that are of interest in the model. An

agent has behaviors that relate information sensed by the agent to its decisions and actions. An agent's information comes through interactions with other agents and with the environment. An agent's behavior can be specified by anything from simple rules to abstract models, such as neural networks or genetic programs that relate agent inputs to outputs through adaptive mechanisms.

- An agent has a state that varies over time. Just as a system has a state consisting of the collection of its state variables, an agent also has a state that represents the essential variables associated with its current situation. An agent's state consists of a set or subset of its attributes. The state of an Agent-Based model is the collective states of all the agents along with the state of the environment. An agent's behaviors are conditioned on its state. As such, the richer the set of an agent's possible states, the richer the set of behaviors that an agent can have. In an Agent-Based simulation, the state at any time is all the information needed to move the system from that point forward.
- An agent is social having dynamic interactions with other agents that influence its behavior. Agents have protocols for interaction with other agents, such as for communication, movement and contention for space, the capability to respond to the environment, and others. Agents have the ability to recognize and distinguish the traits of other agents.

Agents may also have other useful characteristics:

- An agent may be adaptive, for example, by having rules or more abstract mechanisms that modify its behaviors. An agent may have the ability to learn and adapt its behaviors based on its accumulated experiences. Learning requires some form of memory. In addition to adaptation at the individual level, populations of agents may be adaptive through the process of selection, as individuals better suited to the environment proportionately increase in numbers.
- An agent may be goal-directed, having goals to achieve (not necessarily objectives to maximize) with respect to its behaviors. This allows an agent

to compare the out- come of its behaviors relative to its goals and adjust its responses and behaviors in future interactions.

- Agent may be spatial-aware, living in a discrete or continuous space, thus interacting with other agents and the environment.
- Agents may be heterogeneous. Unlike particle simulation that considers relatively homogeneous particles, such as idealized gas particles, or molecular dynamics simulations that model individual molecules and their interactions, agent simulations often consider the full range of agent diversity across a population. Agent characteristics and behaviors may vary in their extent and sophistication, how much information is considered in the agent's decisions, the agent's internal models of the external world, the agent's view of the possible reactions of other agents in response to its actions, and the extent of memory of past events the agent retains and uses in making its decisions. Agents may also be endowed with different amounts of resources or accumulate different levels of resources as a result of agent interactions, further differentiating agents.

2.5 Literature Review: Use in Logistics

Applications of Agent-Based Modelling span a broad range of areas and disciplines, ranging from modelling agent behavior in the stock market and supply chains to predicting the spread of epidemics and the threat of bio-warfare, from simulation of socio-economic systems to understanding consumer purchasing behavior, from logistics optimization to modelling the engagement of forces on the battlefield or at sea, and many others. Some of these applications are large scale in nature, in which a system is modelled in great detail, meaning detailed data are used, the models have been validated, and the results are intended to inform policies and decision making. These applications have been made possible by advances in the development of specialized Agent-Based Modelling software, new approaches to Agent-Based model development, the availability of data at increasing levels of granularity, and advancements in computer performance.

In contrast to System Dynamics which majorly focus on relationship among macro-variables of a system in top-down manner, Agent-Based Modelling tries to model how local and predictable interactions among micro-components of a system can generate a complex system-level behavior.

In the recent years, Agent-Based Modelling has been adopted as the main tool to solve complex problems in the fields of logistics and transportation. This is due not only to the reason discussed above but also to several features that make this approach more valuable than System Dynamics, which majorly focus on relationship among macro-variables of a system in top-down manner, as opposed to Agent-Based Modelling local interactions among micro-components of a system, thus generating a complex system-level behavior. In fact, Agent-Based Modelling is particularly effective in case of interactions between the agents that are nonlinear, discontinuous, or discrete, when space is crucial and the agents' positions are not fixed, when the population is heterogeneous and each individual shows differences, when the topology of the interactions is complex, when the agents exhibit complex behavior, including learning and adaptation.

Hereafter some academic papers are presented and discussed, showing successful implementation of Agent-Based Modelling in real world cases.

- in Agent-Based Simulation for Supply Chain Transport Corridors, Pal (2015) presents the use of multi-agent systems to model and simulate some aspects of transportation corridors, focusing on costs of container unloading, reloading, and waiting time using data sets. In this way, agents provide an appropriate mechanism to model and simulate corridor traffic management since it offers a personalized way to abstract and describe every autonomous entity on the individual level, analyzing bottlenecks and improving the operating efficiency.
- in Modeling logistic systems with an Agent-Based model and dynamic graphs, Démare, Bertelle, Dutot and Léveque (2015) reproduce through simulation the behavior of logistic systems defined as corridors established between a gateway port, where goods are imported, and urban areas, where the final distributors are located. An Agent-Based model coupled with dynamic graphs has been chosen to design the model, showing how macro-patterns emerge from micro rules.
- in Agent-Based Simulation for Yard Management in Container Terminal Operations, Brito (2016) considered Agent-Based simulations to hold high promises for developing complex logistics systems, based on the fact that it is able to build analysis of systems whose behavior is associated to emergent properties, answering more complex questions not treated by traditional simulation methodology. Agent-Based simulations fulfill the high flexibility, representability and promises for logistics systems, even extending its contribution to some of the conceptual frontiers of the simulation methodology, such as generic modeling methodology discussion and simulation-optimization integration
- in Agent-Based Container Terminal Optimization, Winikoff et al. (2011) describe an emulation platform of a container terminal involving the operation of distributed entities (e.g. quay cranes, straddle carriers). This is

obtained using an agent framework, along with two optimizations related to negotiation and physical allocation of resources.

- In Building Agent-Based Models of Seaport Container Terminals, Vidal and Huynh (2010) provided a powerful tool that terminal operators could use to assess the performance of various contemplated crane service strategies as well as the effect of having additional cranes or fewer cranes due to mechanical problems and/or scheduled maintenance.
- in Evaluation of Airport Capacity through Agent-Based Simulation, Peng, Wei, Junqing and Bin (2014) proposed Agent-Based architecture to model aircrafts' movement in order to predict the airport capacity and satisfy the domestic airport property.
- in Agent-Based Simulation Study for Improving Logistic Warehouse Performance, Ribino, Cossentino, Lodato and Lopes (2016) performed an Agent-Based simulation to analyze the behavior of automatic logistic warehouses under the influence of specific factors, thereby obtaining indicators to supporting decision making during warehouse performance improvement
- in Simulation of operational policies for transshipment in a container terminal, Henesey, Persson and Davidsson (2006) Agent-Based simulation is applied to the problem of analyzing operational policies for transshipping containers in a port container terminal. The managers involved in transshipment operations, i.e., the terminal manager, the port captain, the stevedores and the ship captains, as well as some of the operators of the physical resources, such as cranes and straddle carriers, are modelled as agents. Policies concerning allocation sequencing are then compared.

Chapter 3

Dubai Overview

3.1 Evolution of Logistics

The process of ever closer economic integration across the world which is synthetically referred to as "globalization" is mainly propelled by two powerful engines: telecommunication and logistics. The technological advances in ICT have attracted most of the public's attention because they have transformed the life of billions of individuals. But the other engine of globalization has probably been even more momentous. By virtue of an impressive drive to standardize, connect and integrate procedures and physical handling of thousands of transport operations across industries and continents, the economies of distant areas can exploit today synergies unthinkable even two decades ago.

The innovations at the end of last century have intensified global interactions, modified the economic geography, reshaped the relative attractiveness of routes and slashed entry barriers. But they have also deeply altered the fundamentals of the trade business and the logistics sector raising the bar in terms of organizational scale and investment needs. For example, containerization has led to a massive increase in capital intensity of ports, airports and intermodal facilities because mechanized transfer systems require wide space for maneuvering, warehousing and docking. Before the advent of containers, the typical seaport berth was 50m wide and required one hectare of storage space. By the late 1990s, the typical size had increased to 300m and a terminal required 50-100 hectares of storage to accommodate economically meaningful operations.

The revolution in transport systems brought about by container shipping, jumbo aircrafts and adoption of IT in transport management, has consolidated equilibria

and created a new spatial structure of transport routes worldwide. Dubai has taken advantage of this process, and actually anticipated it with foresight through the construction of Jebel Ali Port that was launched in the late 70s and is currently the biggest port in the Middle East by far and the 10th busiest in the world.
With the beginning of freight operations at Al Maktoum International Airport in June 2010 another milestone has been reached for the Emirate and the global logistics industry.

Lower intermodal transport costs have strengthened the links between distant areas, with Dubai taking advantage of this transformation thanks to the completion of state-of-the-art infrastructure built in anticipation of a large jump in global trade.
Dubai has indeed been one of the focal points of this transmutation, becoming in a very short time one of the pivotal trading hubs in the world, thanks to its location along the ancient silk road. In fact, it is positioned at the southeast of Arabian Peninsula, making it the crossroad of international trade between the East and the West.

3.1.1 The Masterplan

Most people believe Dubai's revenues came primarily from oil but, in the past, the Emirate only used a moderate amount of oil reserves to generate the infrastructure for trade, manufacturing and tourism, in order to build up its long-term economy.
It is true that the most important natural resources of the United Arab Emirates (UAE) are oil and natural gas, which contribute for the major part to country GDP. In 1975, oil contributed around 60% to the UAE's GDP, but fell to 37% in 2005 and further down to 30% in 2017.
However, it is also true that roughly 91% of the total reserves are concentrated in Abu Dhabi, with the remaining part distributed among the other six emirates: Dubai, Sharjah, Ajman, Umm Al Quwain, Ras Al Khaimah and Fujairah.
Moreover, in 2014 oil prices fell sharply, bringing a contraction in Gulf Cooperation Council (GCC) countries' GDP. In the aftermath, many GCC nations sought to reduce their dependence on oil, following UAE as a model for

diversifying the economy. In fact, a very large part of their income depends on the production and sale of oil to other countries, which means that their standard of living sometimes fluctuates wildly in correlation with the price of oil. Even Saudi Arabia looks up to it for inspiration, recently pushing for foreign investments and opening up for tourist visa. The UAE comes to the forefront with two of its emirates, Abu Dhabi and Dubai, being global marquees. Abu Dhabi focuses on the investments of aerospace, nuclear power, defense, information technology and petrochemical industries, whereas Dubai has heavily made investments for the development of tourism, logistics, exhibitions, trade, banking and a port for manufacturing and service industries. Quite unexpectedly, more than 95% of Dubai's GDP is generated from non-oil-based sectors.

3.1.2 Logistic Trends

Over the years Dubai has grown as a leading logistics hub providing unprecedented logistics facility and extending numerous opportunities to any foreign investor for setting up business with 100% ownership holding and tax-free income.

Most of the logistics activity in the country takes place through the utilization of ports and free zone areas. Having increased the number of warehouses and developed infrastructure and freight facilities, the city's excellence in logistics service sector has been a major factor for most of the development in the country.

This idea was envisaged by His Highness Sheikh Mohammed Bin Rashid Al Maktoum, Ruler of Dubai, Vice President and Prime Minister of the United Arab Emirates. The original Master Plan of Dubai logistics city revealed in 2006, was limited to the geographical boundary of Dubai World Central and was hence revamped in 2014 by a revised master plan to combine the greater Jafza and DP World into the Dubai Logistics Corridor. The Dubai Logistics Corridor now connects the Al Maktoum International Airport (DWC) at Dubai World Central, the Jebel Ali Port (DP World) and the Free Zone into a much bigger business model connected by three important modes of transportation together.

In addition, the transportation sector would expect a major change with the implementation of upcoming infrastructure development which mainly includes the GCC wide railway network with dedicated freight lines. The road network across emirates is another factor that could improve efficiency in logistics activities as they are expected to be transformed into freight line.

Apart from historical reasons, the logistic sector is crucial for Dubai economy. Not only it constitutes roughly 14% of the emirate GDP, but it enables many other sectors, of which wholesale and retail trade is by far the most important. In particular, freight forwarding contributes the largest share of 63% along with transportation which is the second largest contributor with about 18% of total logistics revenues. Two other significant factors contributing to the growth of UAE logistics market include warehousing which is 14.2% and Value-Added Logistics Services such as package and crossdocking which contributed about 4.1%.

As a natural geographic gateway between East and West and an ideal strategic crossroad between developed and emerging markets, the opportunities for global trade are vast for UAE based firms. According to the data provided by Dubai Statistics Office, foreign trade is classified in 3 main categories: Import, Export and Re-export. The latter, also known as entrepot trade, consist of foreign goods exported in the same state as previously imported, from the free circulation area, premises for inward processing or industrial free zones, directly to the rest of the world. It may occur when one member of a free trade agreement charges lower tariffs to external nations to win trade, and then re-exports the same product to another partner in the trade agreement, but tariff-free. Eventually, it can be used to avoid sanctions by other nations, as the case of Iran avoiding United States trade sanctions. From a quantitative point of view, Dubai ranks among world's largest re-export hubs, with a double-digit increasing trend in the last three years. A closer look at trade type shows a progressive increase of 4% annually in export and re-export in Free Zone Trade, compared to an almost stationary outlook in the ones of Direct Trade. Imports, net of re-export, are slightly diminishing in the same timeframe considered.

From a qualitative point of view export commodities are crude oil, natural gas, dried fish, dates, pearls, precious and semi-precious stones, gold, pulp and wastepaper, sulphur and unroasted iron pyrites, metalliferous ore, metal scrap, chemicals and – of course – Re-export products. On the contrary, the main import commodities have been sorted by the same source as machinery, instruments and transport equipment, chemicals, food, cotton, accessories, gems and jewelry, man-made yarn, fabrics, base metals, cotton yarn, marine products, plastic and linoleum products and tea.

3.2 The Country Development

Through substantial investments in transport infrastructure and logistics capabilities Dubai has become a major regional trade gateway and a re-export zone on the EMENA trade route. The Dubai Strategic Plan 2015 or DSP 2015 was originally launched in 2007 by Sheikh Mohammad bin Rashid Al Maktoum. Soon after, in 2010, the United Arab Emirates' Government followed with the UAE Vision 2021, which "sets the key themes for the Social and economic development of the UAE" and calls for "a shift to a diversified and knowledge-based economy". The final and comprehensive goal of this vision is to give the United Arab Emirates the status of one of the best countries in the world by 2021, the nation's Golden Jubilee Year. It was a public strategy document that put forward a 'vision' for the city-state's development – Dubai Plan 2021 – describing in the "The Economy" section the transformation of Dubai into 'A Pivotal Hub in the Global Economy' thanks to an enrichment in Dubai's maritime port and airport operators. The goals in the UAE vision 2021 could be compared with other development plans in GCC countries, as Kuwait Vision 2035, Saudi Vision 2030 and Qatar National Vision 2030.

Indeed, another well-known milestone for Dubai development will be Expo 2020, taking place between October 2020 and April 2021, thus taking advantage of the mild climate. It will be organized under the theme "Connecting Minds, Creating the Future", which reflects Dubai's history of new initiatives, pioneering ideas,

innovation and future vision. Dubai Expo is expected to attract about 25 million visitors, 70 per cent of whom will come from overseas, making it the most global event in the history of exhibitions in the emirate. After April in 2021, more than 80 per cent of the Expo-built structures will be retained and repurposed for District 2020.
District 2020 will see the urbanization of the Expo 2020 site into a smart and sustainable micro city-within-a-city that aims to provide an innovation-driven business ecosystem, while being a diverse and thriving urban community.
District 2020 is curating a diverse mix of tenants that include global corporations, SMEs, and startups, as well as enablers and social and educational platforms. This huge trade and economic event, as well as the following development, will have a positive impact on all sectors and activities, but retail, tourism, logistics and real estate are expected to get the most benefits.

3.2.1 Favorable Factors

The aforementioned strategic plan for long-term development wouldn't be even conceivable without proper conditions and favorable factors, that are briefly explained hereafter: location & connectivity, infrastructure & regulations, politics, free economy & business opportunity, critical mass of logistics companies.

As already stated above, the main factors that makes Dubai a logistical hub is its strategic geographical location, as it is at the crossroads of important shipping routes. It is located midway between Asia and Europe, linking the Pacific, Atlantic and Indian Oceans. This strategic location coupled with other logistical advantages gives Dubai tremendous opportunities to be one of the leading logistical hubs worldwide. Dubai connections with Red Sea, the Indian sub-continent and East Africa increase the success in the free zone of Jebel Ali. The goods manufactured in the Asian countries are re-exported through Dubai to various parts of the world including Africa, the Middle East, Russian Commonwealth and to the eastern Europe within two weeks of transit time.

The use of multimodal transportation ensures a smooth and efficient stream of the logistical flows positioning DLC as a tri-modal option in the present and a quadra-modal system in the future (sea, air, road, rail):

- Jebel Ali Port (DP World) is considered to excel in connectivity, having the best infrastructure across sea freight ports and also providing multiple sailings to most ports across the region.
- The road network, exceeding 168000 km, makes it possible to transit across the Arabian Gulf within a period of just 2-3 days.
- Dubai International Airport (DXB) represents a hub of critical importance both for passenger traffic and for cargo operations. The expansions plan currently taking place at Al Maktoum International Airport will further enhance Dubai's global attractiveness, increasing trade, industrial and logistics capacity.
- The Etihad Railway is a new initiative designed to connect the UAE, west of Saudi Arabia and East of Oman, further strengthening UAE's importance and claiming to remain the major logistics hub in the years to come. With a 1200 km network it is expected to carry approximately 50 million tons per annum of cargo.

Apart from physical infrastructure, ICT enables companies to enhance their speed of operation, communication, document processing and transparency in logistics operation. Prior to 2009, this technology which is essential to provide value-added services was not readily available within the UAE, whereas it is highly available and affordable in the current scenario. The Logistics District became the first free zone in Dubai to invest in an electronic system that regulates the process of issuing entry permits, in line with city commitment in applying smart solutions across all of its districts, thus transforming the Emirate into a smart city.

To derive full advantage of the logistical infrastructure and services of the city of Dubai, the regulatory and administrative system needs to ensure seamless and efficient flow, for instance, making the administrative procedure operate 24/7. Another aspect that requires to be addressed, concerns the simplification and integration of various steps of procedures into one or two single steps, hence decreasing the lead times needed to clear the merchandise making the logistical system more efficient and thereby, more competitive.

Stability or instability in the political environment has an important role to play in attracting the foreign investors, since it is a relevant element in deciding whether to invest capital or not. For free zones to be successful, political as well as economic stability is of prime importance, as foreign investors need consistency in the host country when political environment is concerned. From this perspective, UAE as a whole and Dubai in particular enjoy a very stable political system. The country has also been instrumental in bringing peace among the Arab Spring regions.

The free zone business model was set up with the intention of streamlining trade and logistics activities across the countries, as well as boosting the country's economic growth by extending various tax exemptions. Dubai currently accommodates more than 30 free zones and each one is governed by its own Free Zone Authority (FZA). Jebel Ali Free Zone Authority (JAFZA) was established in 1985 and became UAE's first free zone. Since then, JAFZA has remained a landmark zone of the UAE and a thriving center to over 7,500 businesses, including 100 global conglomerates from the list of fortune 500 companies. Some of the strategic advantages for logistics companies operating in Dubai Free Zones are: 100% foreign ownership, 100% import and export tax exemption, 100% repatriation of capital and profits, corporate tax exemption, no taxes on personal income, liberal visa policies, abundant availability of space, energy, offices, warehouses, workforce etc.

In this perspective, availability of blue collars at a competitive cost is considered to be one of the most important factors for logistics in attracting foreign investors, allowing to lower the cost of production, transportation and other overhead costs affecting the logistics. For instance, in Dubai, workers from different parts of the world such as Sri Lanka, Bangladesh, Pakistan, Nepal, Ethiopia and Philippines are available at a very competitive salary for the prosperity of the sector.

In addition, prices of commercial real estate – either rent or buy – are seeing a progressive contraction in the last couple years due to excess supply. That is caused by Dubai's long-term expansion plan, but in the short run it translates in an opportunity for foreign investment.
While Al-Maktoum International Airport is not yet fully operational, it is expected to function optimally in the near future, which in turn will drive up the demand for commercial real estate close by and inevitably push up prices. Savvy investors know that in a few years when prices will rise again, they will reap an excellent return on their initial investment.

Since the recovery of Dubai financial crisis in 2012, the establishment of companies in UAE has been growing significantly. In fact, the cluster of logistics companies and its various types have been instrumental in uplifting the JAFZA's growth. There has been a tremendous increase in the amount of imports and exports as well as in the number of organizations that are setting up their business within the UAE from 2010. Major international logistics players are preparing for this extensive investment in the logistics operations by increasing their capacity of their shipment and warehousing capabilities along with other value-added services.

3.2.2 Inhibitors

Over the years, Dubai has undergone tremendous transformation to position itself as one of the most preferred international logistics hubs. Some of the key issues that the industry is exposed to relate to Dubai being a small geographical location facing extreme weather conditions. In addition to this, Dubai continues to face other eco-geo-challenges in an attempt to sustain this position in the long run. The followings are other major challenges affecting the growth.

Inevitably, state-of-the-art transportation infrastructure is the backbone of a modern competitive international logistics hub. However, the efficiency and seamless connectivity between the administration and customs are equally important, such as fast and accurate clearance of goods. In UAE, trade facilitation and shipment handling services are not available 24/7 and are highly complex compared to benchmarking cities, like Singapore and Hong Kong. Transportation of goods by trucks is also restricted during peak hours in order to avoid roadblocks, thus increasing the lead. The clearance time without inspection takes almost one day and the numbers of agencies involved can be as high as three, as opposed to Singapore (same days and 1 agency). Furthermore, it takes a longer time to import and export goods in UAE (2 days) than in Hong Kong (1 day) and is more. Moreover transparency, timings and adequate provision of information from customs clearance and other border agencies is relatively lower.

Extreme temperature during summer, high humidity and sandstorms negatively affect the products in the warehouse or while being transported. Temperatures above 40°C can easily damage perishable goods necessitating controlled ambient while high humidity and sandstorms can damage the packing, exposing the product to further damage. These natural factors cause an increase in the costs of warehouse and shipment in general, which is bound to have a negative impact on the shipper's profit.

The logistics sector faces a lack of skilled workforce which is of utmost importance for the various projects undertaken to enhance Dubai's position as a logistics hub. This draws light to the fact that there is a gap in the skills required in the logistics industry at the managerial level and what is currently available in the market. Furthermore, Dubai is highly dependent on expat workforce and therefore it is difficult to retain existing talents.

The factors that hinder Dubai from being one of the most preferred logistics hubs are the high rents as well as the cost of operation. Increased rents will lead to demand for higher wages by employees, thus increasing the overall cost of engaging in business across borders. Currently, middle income earning expats find it difficult to cope up with the high rental price in the city. As a result, heavy migration takes place to less expensive cities like Sharjah or Ajman, which are well connected by expressways and are within an hour's drive from the logistics corridor.

The sustainability and preference of Dubai can be influenced by the development of ports and free zones in surrounding zones. Other GCC countries like Qatar, Kuwait, Saudi Arabia, Bahrain and Oman, as well as Djibouti and Egypt, are creating some degree of competition to the Dubai Logistics Corridor, intensifying and upgrading their own logistics industry to emerge as alternative regional hubs, posing threats to the UAE's position as the preferred logistics hub in the Middle East for the future.

Upcoming Expo 2020 has also a great impact on UAE logistics companies, which are redesigning their transportation network by investing in public transport, transit accommodation, warehouse facilities and a strategic positioned site for high scale supply chain.

Potential risks associated with cargo security necessitate more stringent screening, which poses a tradeoff between cargo security and international trade growth. This entails the risk of delay to the consignment as the screening of cargo may take time. Another problem that is prevalent is that since most of the manufacturing players have only trading operations within UAE the logistics sector become more focused on freight forwarding. Since Dubai's logistics sector is not as advanced as Singapore, it lacks the cutting-edge innovation and technology that can cut down logistics delays in order to speed up the business process. This can lead to certain systematic risks as the technology in use is not kept up to date which may in turn, lead to systemic errors and cause further delay. Since the supply chain activity relies heavily on coordination between several parties for successful execution, the industry needs to constantly strengthen ties between all players so as to avoid procurement and receivable risks, ensure standardization across the operating bodies, stakeholders and institutions.

3.3 Dubai Logistics Corridor

As already introduced above, the present thesis main focus is on the Dubai Logistic Corridor, constituted with the purpose of harmonize and streamline logistics operations, no matters whether it involves air or sea transport.

Located in the southern part of Dubai (left portion on the map below), it is connected to Dubai downtown, International Financial Centre and Dubai International Airport (DXB) through Sheik Zayed Road and the Metro Line

It is composed essentially by three main entities that will be briefly explained hereafter: JAFZA, Jebel Ali port (DP World), Al Maktoum International Airport (DWC).

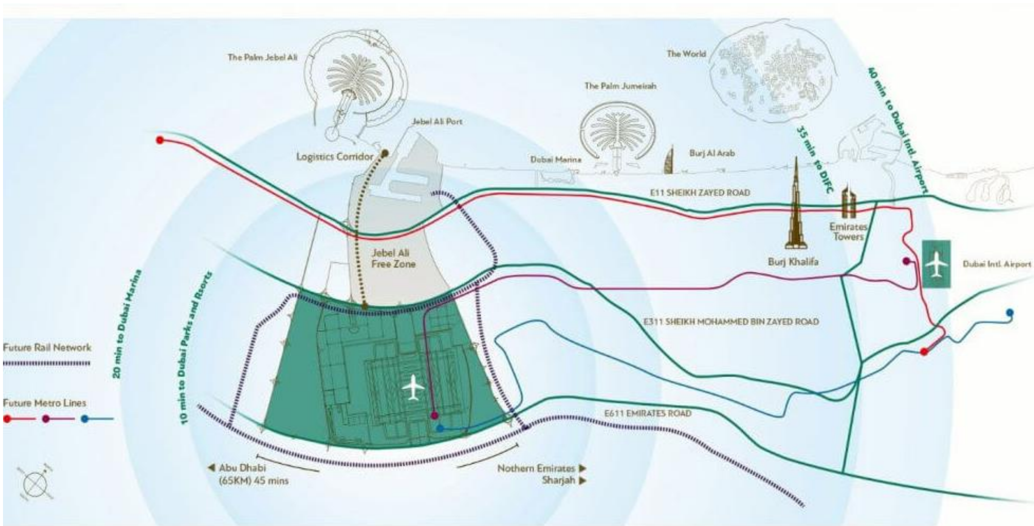


Figure 5 - Dubai

3.3.1 JAFZA

Jebel Ali Free Zone is one of the world's leading free trade zones, originally created in 1985 under a Ruler's Decree to promote trade and support container throughput at the Jebel Ali Port. Located in southern Dubai, right behind Jebel Ali Port, Jafza is spread over 57 square kilometers on both sides of Sheikh Zayed Road, and is accessible via road, Danube and Metro Red Line.

Today Jafza accounts for almost 23.9% of total FDI flow into the country and contributes 23.8% to Dubai's GDP, hosting more than 8,600 companies from over 100 countries, including approximately 100 Fortune 500 companies. Moreover, it sustains the employment of more than 150,000 people in the United Arab Emirates and generates trade worth US$93 billion.

It is one of the largest free zones globally and the region's most efficient logistics hub, linking the port, free zone and airport, creating one of the most efficient sea-air logistics links in the world. In fact, both Jebel Ali and JAFZA are linked to the Dubai World Central complex, a planned mega residential, commercial and logistics zone that also encompasses Al Maktoum International Airport. In line with concepts of integrated logistics space, the various infrastructural elements are part of one unified customs zone with an airport–seaport corridor, allowing cargo to be moved from port to airport within 20 minutes, ultimately leading to a smother process and reduced costs. Above all, having a unique customs area makes it possible to avoid various procedures of customs, duties and legal compliance that cargo must fulfill when moving across zones, process that is both time and cost incurring.

Prior to the formation of the DLC, a lot of documentation and customs work had to be complied once the goods left JAFZA for the DWC, leading to a complete clearance in about 4 days. Thus, the DLC business model would help the companies to reduce their lead times and be able to enjoy more responsive logistics, while not compromising on maintaining operational efficiency. This innovation, of course, is a win-win solution for all of the stakeholders involved in the process, since at each and every custom point not only documentation but also financial guarantees needed to be produced for the go through. Ultimately, companies need not open branches in both free zones but can easily operate through a single branch with the same level of service and operational easiness, as opposed to the past where they needed to open branches both in the JAFZA for the DP world and at the DWC in case of frequent requirement for sea-air or air-sea operations.

3.3.2 Jebel Ali Port

Three main ports currently serve Dubai's foreign merchandise trade activities: Port Rashid, Al Hamriya Port and Jebel Port, in addition to traditional piers along the Dubai Creek. Foreign trade has developed through free zones as a result of the massive infrastructure built in Jebel Ali Port. As an integrated multi-modal hub offering sea, air and land connectivity, complemented by extensive logistics facilities, the Port plays a vital role in the UAE economy. It is a premier gateway for over 90 weekly shipping services, connecting more than 150 ports worldwide. Being DP World's flagship, it is the largest non-Asian seaport and ranks among the top 10 container ports of the world, with a handling capacity of 19.3 million TEU. Container Terminal 1 is the busiest terminal, and ever since the creation in 1979 has allowed Jebel Ali Port to achieve its position as one of the top ten ports globally. It reaches a capacity of 9 million TEU, thanks to 15 berths and 51 quay cranes. In comparison, Container Terminal 2 is smaller with 32 quay cranes, 8 berths and has a capacity of 6.5 million TEU.

Container Terminal 3, inaugurated in 2014, is known for its remarkable technological achievements since it is one of the largest semi-automated terminals in the world. It has 5 berths, a capacity of 3.8 million TEU and features 19 automated quay cranes and 50 automated rail-mounted gantry yard cranes. In addition, it is capable of handling Ultra Large Container Vessels with capacities exceeding 18,000 TEU. Container Terminal 4 will be the next benchmark for the world of trade with capabilities designed to serve the current and future market requirements. On its completion it will take the port capacity to 22.4 Million TEU, even if it can seem a bit overestimated given the utilization of the recent years. In fact, data provided by Statista and Dubai statistics office, show that the volume of handled container progressively grew until 2015 and almost stabilized since then. This is in line with the other top 10 ports, that show a contraction in 2009 and in 2016 due to macroeconomic factors. However, in general terms, the outlook for Dubai in the mid-long term looks positive, as opposed to the example of Hong Kong that has not recovered from that situation. Jebel Ali handled 14.95 million TEUs in 2018, compared to the 15.59 of the 2015 peak; however, it is almost double than what handled in 2005, showing the impressive development in the long run. In comparison, Hong Kong port lost utilization, moving from 22.43 million TEUs of 2005 to 19.6 of 2018.

Apart from containers, Jebel Ali also handles vehicles, break-bulb and bulk. Again, vehicles and break-bulk are diminishing after reaching the peak of 2015; on the flip side, the quantity of bulk goods handled remains almost stationary, between 4.5 and 5.0 million metric tons.

3.3.3 International Airports

The air transport sector is one of the most important success stories in the Emirate of Dubai. The sector's progress began in the mid-1980s and continues to this day constituting one of the cornerstones of the emirate's economy. The exceptional growth of the air transport sector over the past two decades has been due to several factors, including most prominently the strategic location, the air transport infrastructure, and the operating airline companies, above all Emirates Airlines. Dubai is home to two of the world's best airports in terms of efficiency and quality of service: Dubai International Airport and Al Maktoum International Airport.

Dubai International Airport (DXB) was officially open back in 1960 and has continuously expanded ever since, most importantly with the construction of Terminal 2 in 1998 and Terminal 3 in 2008. After that, smaller expansions and re-organization of concourses began in order to fulfill the capacity needed to handle the continuous growth. In the list of world's busiest airports (2018), DXB ranks 3rd considering passenger traffic and 6th considering cargo traffic. Passenger traffic totaled 89.15 million, resulting in the first airport in the world per international flights, since in comparison Hartsfield–Jackson Atlanta and Beijing Capital handle a relatively high number of hub-and-spoke domestic flights. The average annual growth rate in the last 8 years is around 8.4%, despite a small contraction in the last year; compared to the 47.18 million passengers of 2010 the airport has almost doubled its traffic. Cargo traffic on the contrary remains around 2.64 million tons, with a much more moderate growth rate in the recent years (+11% between 2014 and 2018).

The Al Maktoum International Airport (DWC) is part of Dubai World Central, which is one of Dubai's largest air transport projects and world's first fully integrated airport city, including a residential city, a commercial city, a golf city and an aviation city. The impressive total cost of the project is around 32.7 billion USD and is being executed in phases, with a completion date estimated in 2050. Upon completion of the first phase, no earlier than 2030, DWC it will feature 3 parallel runways for simultaneous operations and a planned capacity of 120 million passengers and 12 million tons of cargo per year; with the completion of the second phase the capacity will be increased to 250 million passengers and 16 million tons. The passenger terminal was opened in October 2013 with a number of passengers increasing from about 433,000 passengers in 2015 to about 1 million in 2018. On the other hand, the cargo terminal was opened in June 2010 and cargo movements totaled 189,000 tons in the same year. Since 2014 it is the main hub for Emirates SkyCargo, as a plan to move freight operations from DXB. Ultimately, attention should be drawn on the strategic location of DWC inside the UAE, since it is positioned quite half-way between Dubai DXB and Abu Dhabi AUH, making it appealable not only to Emirates but also to Ethiad.

In the 16th and 17th century seaports and waterways defined the location of cities and in the 18th century roads began to develop inward based on trading opportunities. In the 19th century cities followed railroads and canals and in the 20th century cities built airports. It is expected that in the 21st century airports will determine the location of cities and economic activities.

The Aerotropolis concept, which represents the evolution from city airports into airport cities, is the brainchild of John D. Kasarda, Professor of Strategy and Entrepreneurship at the University of North Carolina, who documented and researched how airports are evolving from transportation and supply chain-focused areas into mixed-use commercial centers.
The opening of part of Dubai World Central development will significantly increase Dubai's trade, industrial and logistics capacity. Dubai World Central will constitute the foremost example of an Aerotropolis, providing an optimal response to the

changing supply chain management requirements in global markets and distribution systems in a key geographical area.

Chapter 4

Model Development

4.1 From SD to ABM

System Dynamics and Agent-Based Modelling are two widely used methodologies in simulating complex real-life systems characterized by dynamic nonlinear relationships. Even if the internal structure is different, both aim at increasing the understanding of the system and testing scenarios with the objective to support decision-making and policy implementation.
The building blocks in specifying a System Dynamics model are stocks, flows and auxiliary variables. Stocks represent the accumulation of material and information, caused by the action of inflows and outflows. While stocks are mathematically described by integral equations, flows are described by differential equations. The solution of these sets of equations describes the aggregated state of the system, which changes continuously over time and depends on the previous state of the system.
On the contrary, the main building blocks of AB are autonomous agents, their decision rules and actions, and the environment in which they interact. Although agents' decision rules usually govern agents' behavior to achieve individual benefits, collective intelligence may also emerge when agents coordinate their decisions to achieve common goals. Therefore, analyzing solely the internal mechanism of agents does not explain the macro level observations. Moreover, agents' decision making is typically based on limited observed knowledge rather than on complete knowledge of the entire state of the system.

Generally speaking, it is possible to derive a corresponding Agent-Based model starting from a System Dynamics one, since the two are not mutually exclusive paradigms. Then, the Agent-Based model can be further enhanced to capture much more complicated behavior, dependencies and interactions, thus providing deeper insight in the system being modeled.

Compared to System Dynamics models, there is no such place in Agent-Based Modelling where the system behavior or global dynamics would be defined. Instead, the modeler defines behavior at individual level, and the global behavior emerges as a result of many individuals, each following its own behavior rules, living together in some environment and communicating with each other and with the environment. Therefore, Agent-based Modelling is a mindset more than a technology, describing the system using a bottom-up approach rather than a top-down as in System Dynamics.

Hence, the natural consequent advantage is that it provides for construction of models in the absence of the knowledge about the global interdependencies: one may know nothing or very little about how things affect each other at the aggregate level, or what is the global sequence of operations etc., but if there is some perception of how the individual participants of the process behave, it is possible to construct the Agent-Based model and then obtain the global behavior. Thus, even if there exists, say, a System Dynamics model that answers the question, it might be much easier to build the Agent-Based model.

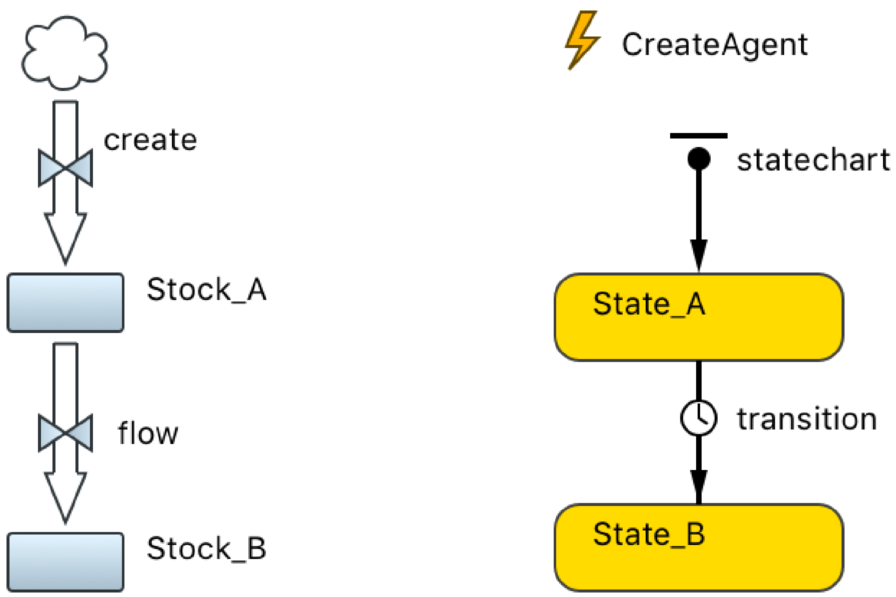


Figure 6 - Converting SD into ABM

The key starting point is to "disaggregate" the stocks (SD) by looking at them as if they are not tanks with liquid but boxes with balls. These balls will become agents, being in states corresponding to the original stocks they were in; consequently, the flows (SD) will be translated into transitions. The moment of time at which a transition will happen intuitively depends on the original flow rate and can thus be modeled by feeding the transition with appropriate parameters. Lastly, sources (SD) can be replaced with an event that creates a new agent according to the timeout or rate provided.

4.1.1 Bass Diffusion Model

The Bass diffusion model, a classic textbook model for product diffusion based on current and potential adopters, is proposed as a trivial example of the above-mentioned conversion. (Borshchev, Filippov, 2004)

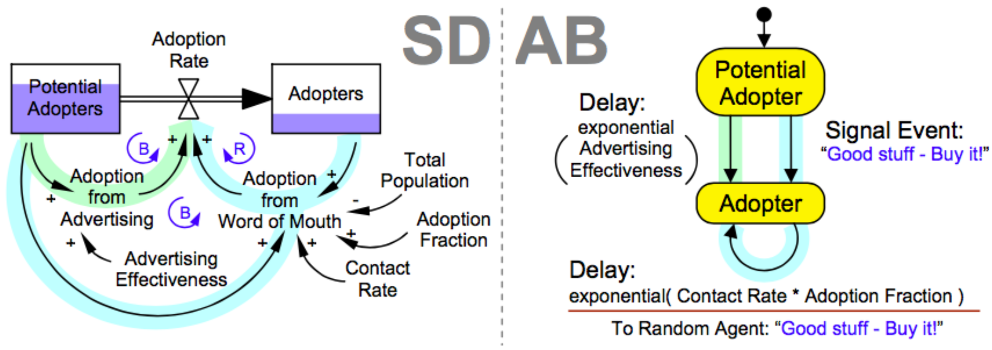


Figure 7 - Bass Diffusion Model

As shown in the figure, an agent can be either in *Potential adopter* or in *Adopter* state at once, following the corresponding stocks in the System Dynamics model. The two summands affecting the adoption rate, namely adoption from advertising and adoption from word of mouth (WOM), are translated into two separate transitions. The former is simply modelled as a rate transition, the latter as a message transition. In fact, once an agent becomes an *Adopter* it starts sending messages to other agents, in turn triggering a transition if a *Potential Adopter* receives one. The results obtained, i.e. the well-known S-shape, are virtually indistinguishable with a sufficiently high number of agents involved. However, the Agent-Based model development is more natural and straightforward than composing a system of differential equations, in particular if there is no previous knowledge of the whole system behavior.

Agent-Based models are also typically easier to maintain with model refinements normally result in very local, not global changes. For example, suppose that the WOM influence of a particular person depend on how long ago he purchased the product, such that the person promotes it actively soon after the purchase and less frequently after a while. To capture this phenomenon only a small modification is needed: the introduction of a variable that track the time of purchase and its introduction in the transition to send a message, so that its frequency depends on the time passed. Is it possible to capture such behavior in System Dynamics? The WOM contribution into the *Adoption Rate* may now be different for any two adopters and it also changes over time for an adopter. Therefore, aggregating the adopters into one (or any reasonable finite number) of stocks will distort the results.

Of course, System Dynamics have faced that kind of problems long ago, and have even suggested a certain partial solution, to be more precise – a workaround: a stock containing objects with sufficiently different properties is decomposed into an array of buckets, and objects move between the buckets as their properties change. Consider however objects having not one, but several such properties. The array of buckets grows as a product of dimensions and, after a few steps the number of cells in the array may easily exceed the number of individual objects in the real world (Keenan and Paich 2004). This obviously makes such System Dynamics model senseless and slow, while the Agent-Based model will always contain as many objects as needed.

4.1.2 Predator Prey

Furthermore, in some cases System Dynamics cannot even provide a solution to deal with certain problems, as for the example of another well-known case: the predator prey model. Based on a pair of differential equations developed by Lotka and Volterra, the model is characterized by oscillations in the population size of both predator and prey, with the peak of the predator's oscillation lagging slightly behind the peak of the prey's oscillation. However, only using Agent-Based Modelling it is possible to overcome some assumption that must hold in the original System Dynamics model, thus introducing variables to account for time and space. About the former, using Agent-Based Modelling, it is possible to give both hare and lynx life expectancy, so that they now die because of age, being eaten up (hare) or starvation (lynx). About the latter, instead, hare and lynx are now space-aware so that it is possible to limit density for births and movement for hare, and limit distance and hunt moves for lynx. In contrast to System Dynamics, subject to the model parameters, lynx or hares may become totally extinct, which never happens in the System Dynamics model due to its continuous nature. The oscillations are stochastic because of the random and spatial nature of the model. In this Agent-Based model is also possible to view the 2D picture and trace a single hare, lynx, their families or generations and advanced analytics.

As just demonstrated, in general using Agent-Based Modelling it is possible to capture more real-life phenomena and useful insights than with an approach based on System Dynamics. However, this does not automatically imply that Agent-Based Modelling is a definitive replacement for System Dynamics, nor that it is always a suitable solution, as in the case shown in the next paragraph.

4.2 Re-conceptualization

AnyLogic is fully able to deal with System Dynamics and Agent-Based model, and even seamless integrate the two into the same environment. It also allows to automatically import and convert a Vensim® model, in order to easily continue developing it. However, a tool to make a straightforward conversion of a model from System Dynamics to Agent-Based is not available inside AnyLogic nor as a third-party application. In fact, converting a model isn't always an easy process and implies two issues of a different kind: one technical and one conceptual.

Concerning the technical issue, since in Agent-Based Modelling the process happens with a bottom-up up approach, individual data must be provided to the agents, as to make the global dynamics emerge from individual interactions. So, this process can work in a way but not in the opposite direction: as System Dynamics deals only with aggregates, extracting the individual data is not possible, as stock contents are indistinguishable from one to another. Switching from a System Dynamics to the corresponding Agent-Based model means – generally – increasing the level of detail, thus requiring a higher granularity of data to be available. In absence of data, a fill-the-blank problem arises, and only averages or estimated probability distribution can be used, which would probably lead to obtain conclusions with a questionable added value.

Concerning the conceptual issue, it must be noted that the re-conceptualization procedure only makes sense in case of enhancements to the agents, as to capture their individuality. If the items stocked inside the System Dynamics model are naturally passive and indistinguishable, there would be probably no benefit in converting them to agents. A notable example would be stocks of money: normally there is no interest in individual dollar history and dollars do not show any active behavior. Hence, from a practical point of view, the re-conceptualization procedure takes place according to the vision and the experience of the modeler, that now has to reformulate the system through a bottom-up approach. In this way, using his own intuition, the modeler is also seeking which aspects to highlight, in order to pursue the desired objective with the system and the data available.

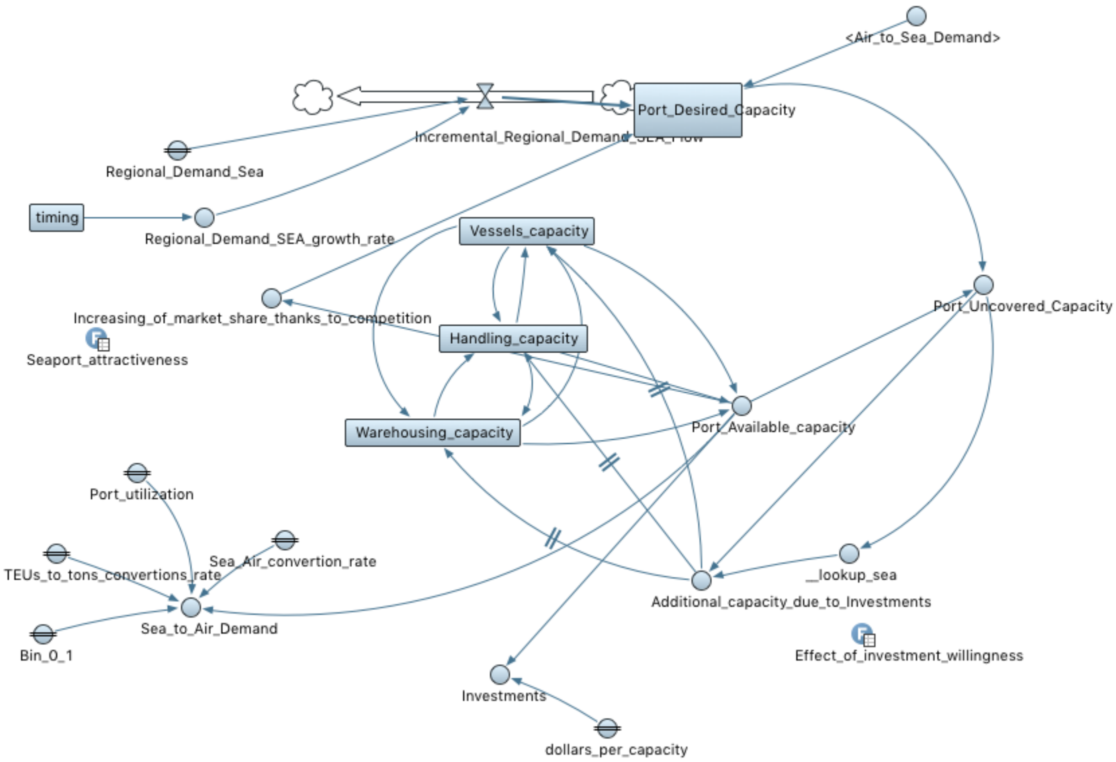


Figure 8 - Port SD Model

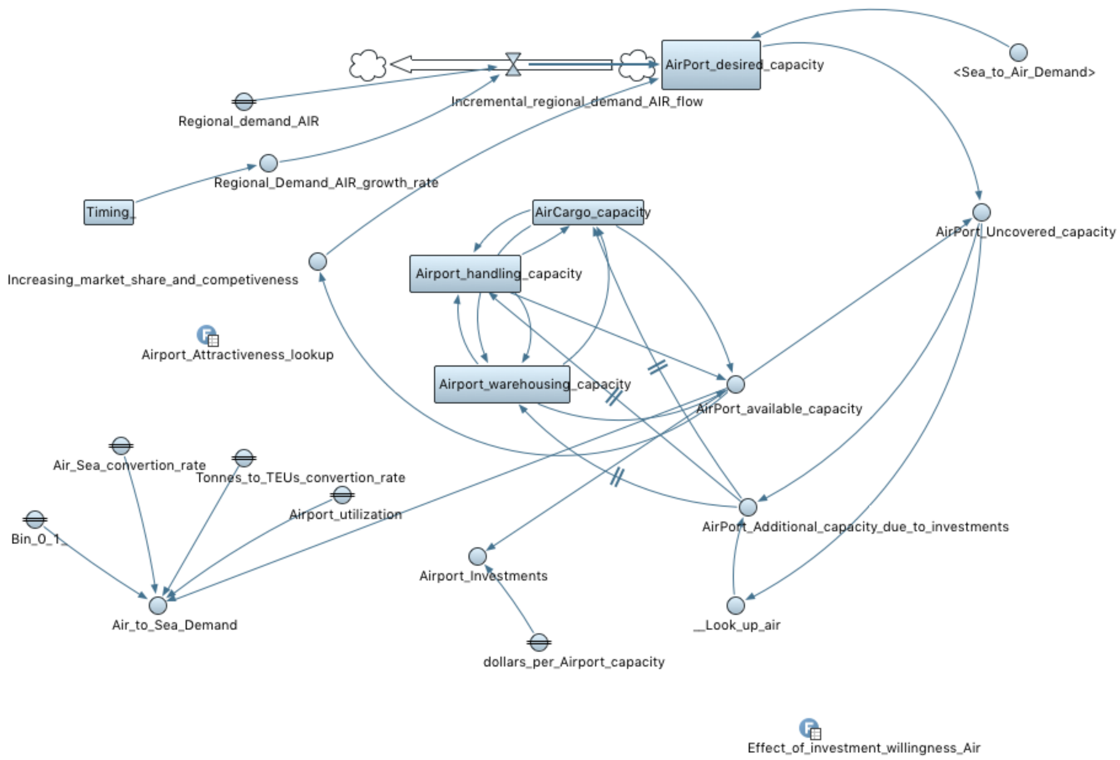


Figure 9 - Airport SD Model

Taking into consideration the earlier study conducted from Postorino, it is possible to note that making a conversion from System Dynamics to an Agent-Based model is nontrivial. The model in fact presents some criticalities that make it difficult to be converted in a straightforward way.
First of all, rather than being composed of a continuous flow of data from one stock to another, it is made up of interactions of a plethora of parameters, such as auxiliaries and constants. This of course implies that, in case of a switch to an Agent-Based model, the corresponding agents wouldn't have suitable states to be in or to make transitions to, but rather would just include some mathematical calculations.
The second important thing to notice is that the units of measurement considered in the original model, mainly dollars required for investments and capacity limits for bottlenecks, are not meaningful to be transformed into agents since they can be simply considered objects that do not show active behavior or individuality.

Despite the issues just described, it is incorrect to state that the model cannot be translated into Agent-Based at all. In fact, it is possible to re-conceptualize the model using a bottom-up approach and focus on the resource that make up the capacity limit. In this way, global dynamics will emerge from individual interaction, with the scope of replicating the functioning of the original model while simultaneously adding details on a lower level. Hence, the attention now will shift to investigate the ability of seaport resources (vessels, handling, warehouses) to satisfy the TEU demand and the ability of airport resources (aircargo, handling, warehouses) to satisfy the Tons demand. Of course, all other variables will be taken into consideration in order to replicate the same result with Agent-Based Modelling, allowing for a comparison between the two paradigms.

4.3 The New Model

Up to this point several topics have been introduced: the dynamics of the Dubai Logistics Corridor, the methodology of Agent-Based Modelling, the opportunities and issues arising from a comparison with System Dynamics and its possibility to be translated into an Agent-Based-model, albeit sometimes with non-negligible drawbacks. In this section the inner part of the work, carried out mainly in Dubai with research, modelling and testing, will be presented. The process behind the model and all the composing blocks will be explained hereafter, and some of the most relevant pieces of code will be also presented. Some simplifications and assumptions were obviously made during the process in order to obtain an effective model of a complex logistics system. In this way it is also possible to obtain a model that is on-purpose not tailored on the Dubai case only, but rather applicable even to other regional logistics hubs in the world, provided that a sufficient amount of data could be fed into the model.

As already brought to attention, the re-conceptualization of the previous model consists of a focus into the resources generating the bottleneck. These resources, that will be now modelled as agents, will be interacting between one and another, thus generating the whole system capacity limit. In addition, the incoming orders, expressed as TEUs for port and Tons for airport, will be also modelled as agents. As discussed in chapter 2, modelling in a practical way in a professional environment allows to consider also passive objects as agents, in order to take advantage of the powerful tools provided by the software to track individuality and interactions with other agents

The baseline of the model is constituted by the default *Main* environment, in which all the other agents live in. According to object-oriented programming, the logic followed in creating agents is as follows:

- Agent Type (class, in which characteristics are defined)
    - Agent Populations[..] (list, in which agents are grouped)
        - agent (the object itself)

Since the seaport and airport ecosystems are modelled in a specular way, just the following agent types are required to build the model: *Order, Transport, Handling, Warehouse, Link.* These types are then used to create Agent Populations for the seaport and the airport, as summarized in the next table.

- The Seaport contains the following populations: *teus, ships, handlings_sea_in, handling_sea_out, warehouses_sea.*
- The Airport contains the following populations: *tons, planes, handlings_air_in, handling_air_out, warehouses_air.*
- The Ground Link, which is used to connect seaport and airport, contains only *links.*

Agent type	**Agent population**		
	Seaport	**Airport**	**Ground Link**
Order	teus[..]	tons[..]	
Transport	ships[..]	planes[..]	
Handling	handlings_sea_in[..]	handlings_air_in[..]	
	handlings_sea_out[..]	handlings_air_out[..]	
Warehouse	warehouses_sea[..]	warehouses_air[..]	
Link			links[..]

Figure 10 - Agent Populations

4.3.1 Main

Figure 11 - Model: Main

Inside AnyLogic, the above-mentioned populations are placed in the *Main* environment and are represented with a red symbol. Sliders and buttons are used to perform parameter variations, in order to see in real time the change in system behavior. All other items, i.e. variables, parameters, collections, functions, events, are used to feed data into the system, for statistics analysis and for debugging purposes.

A fundamental part of the system is represented by the block on the left, which is in charge of generating the orders to be handled by all other resources in the system. It all starts from the events *create_demand_sea* and *create_demand_air*, that respectively generate *teus* and *tons* with a cyclic timer, according to the quantity specified via the variables *demand_sea* and *demand_air*. These go in couple with the events *step_increase_sea* and *step_increase_air*, that are used to periodically increase the demand using the annually compounded growth rate given in input.

4.3.2 Order

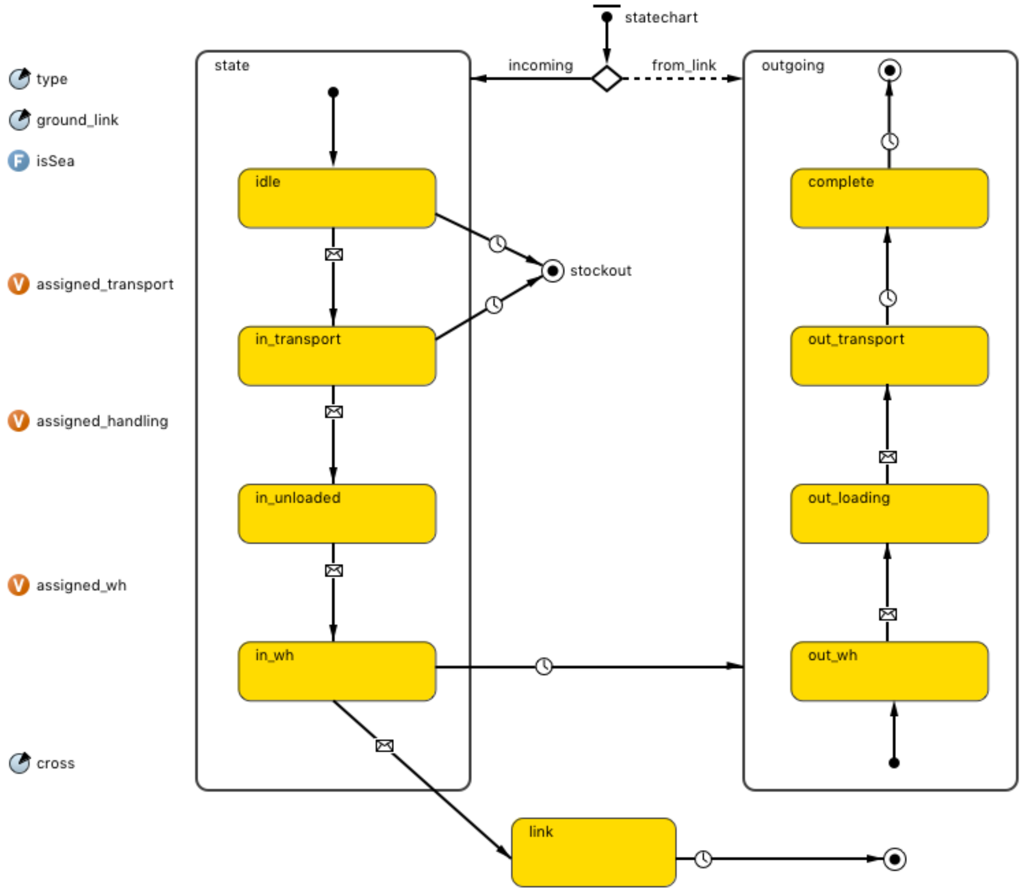


Figure 12 - Model: Order

Orders constitutes the main block on which the whole model is built on. Teus and tons travel throughout the system thanks to interaction of the other agents. In particular, a statechart is used to represents the state in which the order is actually located. An order, following the arrows that describe the process, can be physically located in the transport, in the handling, in the warehouse, either in the incoming or in the outgoing channel. In addition, two virtual locations, idle and complete, are being used. Given the fact that type *Order* is used for both *Teus* and *Tons*, *type* (parameter) and *isSea* (function) are used in conjunction to distinguish between the two.

The process starts with the creation of the order, that enters in the statechart from the above. As for the moment orders coming from the link are not being considered

so the dashed line (*from_link)* is not followed; also, the transition from *in_wh* to *link* is neglected for the moment.
This case will be covered later on.

Following the previous model, the idea is that the Dubai logistics corridor, as an ecosystem of many corporates, has a certain demand of orders (created from exogenous factors) to face and thus has to provide routes and transportation to be able to satisfy it.
As soon as an order is created it is placed in the *idle* state, it is added to a collection (see code) and waits for a transport agent available to “catch it”.

```
if (isSea())
  main.teu_in_idle++;
  main.collection_teu_in_idle.add(this);
} else {
  main.ton_in_idle++;
  main.collection_ton_in_idle.add(this);
}
```

Code Snippet 1 - Order, idle state, entry action

In case the order remains in *idle* state for too long the order is lost, it is eliminated from the system and is accounted as a stockout unit for statistics purposes. That clearly means that transportation capacity is inferior with respect to the demand it has to face and will therefore need an upgrade.
Otherwise, if one transportation agent is available, a message is sent from the transport agent to the order agent and the related message transition, from *idle* to *in_transport*, is triggered (envelope symbol).
After the transportation agent has arrived in the seaport or airport, it is the turn of handling resources to process the order, which takes places in two steps: first unload the transport, second move the order to the warehouse. Again, this happens thanks to an exchange of messages between agents, that in turn trigger message transitions.
After some time spent in the warehouse the order follows the opposite direction (right part of the statechart): starting from the warehouse it is moved to the handling resources and loaded into a transport, that is now leaving Dubai. Hence, the order is completed and after a timeout it is eliminated from the system.

During the whole process, it is possible to track which is the agent that the order is assigned to in each phase, thing that is realized with the variables *assigned_transport*, *assigned_handlings* and *assigned_wh.* This is a feature that differentiates Agent-Based Modelling from System Dynamics, because the latter only deals with aggregates and makes impossible to distinguish single units and obtain detailed analytics.

4.3.3 Transport

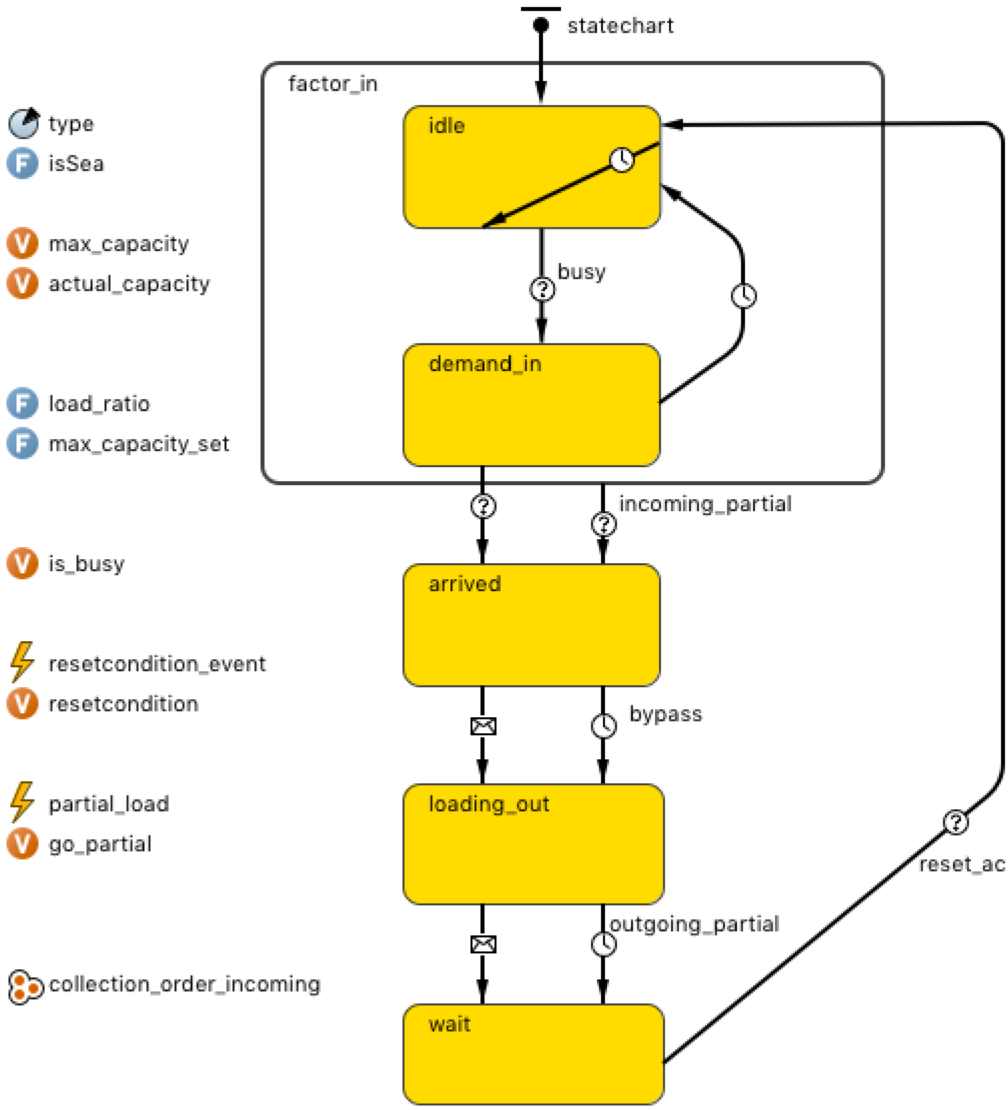


Figure 13 - Model: Transport

Transport represent cargo ships and planes in the system. As made for Order, *type* (parameter) and *isSea* (function) are used in conjunction to distinguish between the two when executing commands. The idea that holds behind this type of agent is the following, according to the statechart in figure:

- *idle*: transport waits for an order to catch, searching for it with the internal transition
- *demand_in*: repeats process until *max_capacity* is reached, then travels to the port or airport
- *arrived*: once arrived, the transport waits to be unloaded by *handlings_in* agents
- *loading_out:* after being fully unloaded, the transport waits to be loaded from *handlings_out* agents
- *wait:* lastly, transport leaves the port or airport and waits for a reset through a cyclic timer event.

The last state is crucial, since it is used to reset the capacity of the transport.
In fact, this is done in order to allow a transport agent to load only a certain amount of orders in the user-defined timeframe, thus creating the general capacity at an aggregate level. The event timer is started as soon as the first order is caught, allowing not only precise time tracking but also the detection of underutilization of the resource. The quantity is defined with the variable *max_capacity,* that is used in connection with *actual_capacity* for a couple of different purposes. Those include assessing whether the transport is fully loaded/unloaded (for triggering transitions) and for tracking statistics.

While being in *idle* state, the transport periodically triggers the internal transition to catch demand. The process is composed of the following lines of code and is briefly described after.

```
if (
  actual_capacity < max_capacity &&
  (isSea()
    ? !main.collection_teu_in_idle.isEmpty()
```

```
5         : !main.collection_ton_in_idle.isEmpty())
6   ) {
7       Order t = isSea()
8         ? main.collection_teu_in_idle.pop()
9         : main.collection_ton_in_idle.pop();
10
11      t.assigned_transport = this;
12      collection_order_incoming.add(t);
13      deliver("demand", t);
14
15      actual_capacity++;
16      is_busy = true;
17      partial_load.restart();
18      event.restart();
19  }
```

Code Snippet 2 - Transport, idle state, internal transition

The if statement prevents the execution of the code in case the maximum capacity is reached (to avoid exceeding the specified maximum capacity) or there is no order to catch (check required to avoid exception thrown). *IsSea* is used in conjunction with the ternary operator? and returns line 4 if true, line 5 otherwise. Hence, with a simple expression, it is possible to perform operations on a different set of collections, teus and tons, with just a single agent type. In case conditions are met, the order to catch is taken from the collection (line 8 or 9), the current transport agent is stored inside the order (line 11), the order is recorded on the transport (line 12) and a message is sent to trigger a transition from *idle* state to *in_transport* state inside the selected order (line 13). Subsequently, actual capacity is increased and a variable is set to true to trigger a conditional transition. In addition, two timeout events are activated: the former is used to send transportation to destination even if not fully loaded to not delay the orders (*incoming_partial*), the latter is used to reset the cycle (*reset_ac*) not before the specified period, thus preventing the transport to exceed the maximum capacity defined.

Being a straightforward process, at first it may look strange that there are a couple of transition per every state change, instead of just one. The reason behind is because the first set of transitions (on the left side) is related to the standard process while the second (on the right side) is used as a workaround solution in case of need.

- *incoming_partial* has already been described and is used to avoid unnecessary delays in case the capacity of ship is much greater than the demand to face.
- *outgoing_partial* follows the same logic of the above, but with the opposite direction. Moreover, it allows for a punctual execution of *reset_ac,* in order to not bias the maximum capacity defined
- *bypass,* as the word recalls, represents a temporary workaround for a very special situation that would cause the block of the overall process:
  It could happen rarely, depending on the stochastic nature of the system, when the warehouse capacity is the one causing the bottleneck and the demand exceeds the capacity for a prolonged time. Transports could end up being stuck in *arrived* state, waiting to be unloaded by the handling agents, that unfortunately are waiting for a free space in a warehouse. Hence, the transport is stuck and cannot be filled with outgoing orders that could in turn free some space in the warehouse. *Bypass* allows to recognize this situation and unlock it, by freeing a transport from orders. The workaround is left as an open point for future development but has been correctly tested and doesn't distort the normal operations of the system, since orders unloaded in this way are eliminated from the system if not picked up by handling agents quickly.

4.3.4 Handling

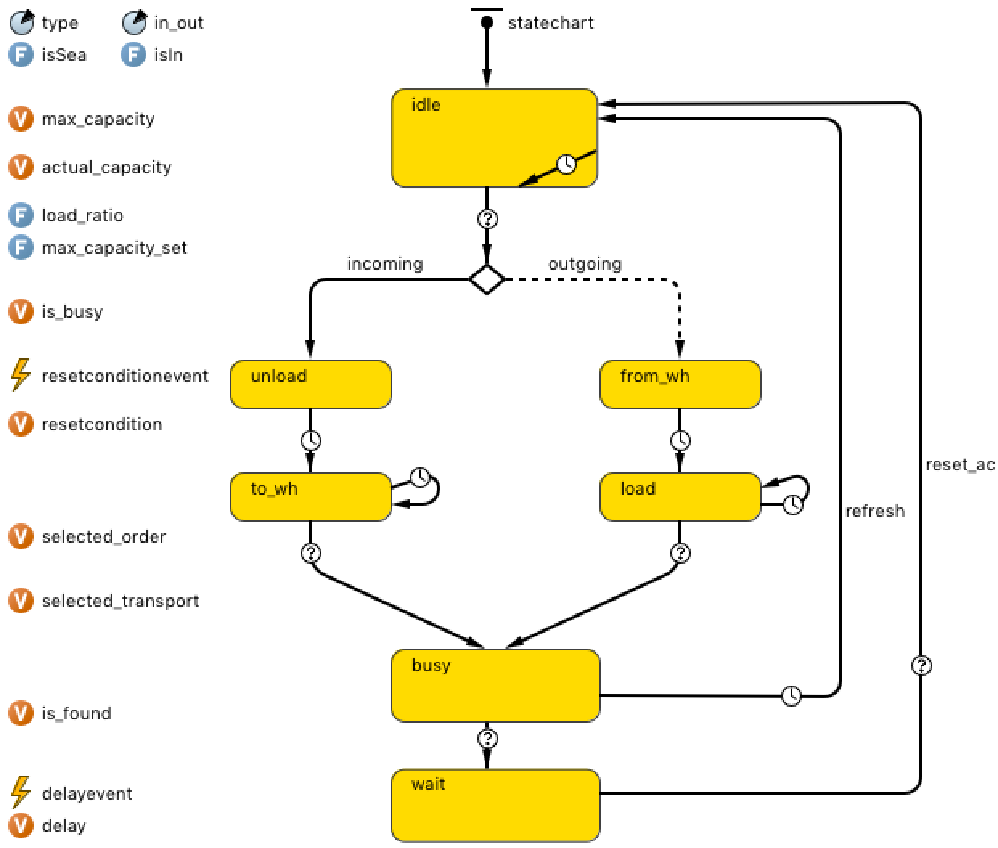


Figure 14 - Model: Handling

Handling resources serve as middlemen between the transport and the warehouses, both for incoming and outgoing orders. As for *Transport* and *Order*, *type* (parameter) and *isSea* (function) are used in conjunction to distinguish between the resources belonging to the port and to the airport. *in_out* (parameter) and *isIn* (function), with the same logic, are used to distinguish resources handling incoming and outgoing orders. In this way it is possible to obtain 4 populations of agents using just a single Agent Type: *handlings_sea_in, handling_sea_out, warehouses_sea, handlings_air_in, handling_air_out.*

As it happened for *bypass*, the distinction between in and out is a practical workaround as it was the simplest way to avoid possible interruptions of the process in case of excess demand and handling resources representing the bottleneck. Again, this solution is left as an open point for future improvements but has been successfully tested and does not change the overall performance of the system, allowing for a precise compliance with the limit given through *max_capacity*.

The iterative process followed by each handling agent is described in the statechart and involves:

- *idle*: wait for an order to process, using internal transition to search it
- (only for incoming) *unload*: unloads order from a transport, by sending a message to it, thus triggering its message transition.
- (only for incoming) *to_wh*: searches for the first available warehouse to accommodate the order processed and sends it there
- (only for outgoing) *from_wh*: moves the order out of the warehouse
- (only for outgoing) *load*: loads order on the transport with same operations as in *unload*
- *busy*: after processing one order the *actual_capacity* counter is increased, and the process is restarted. In case *actual_capacity* has reached *max_capacity,* the conditional transition to the next state is triggered
- *wait*: waits for the reset of the capacity and the start of a new cycle. The event timer is started as soon as the first order is processed, following the same logic that was previously adopted for transport.

While being in *idle* state, the handling agent periodically triggers the internal transition to process orders according to the following lines of code

```
if (!is_busy) {
  if (
    isIn() && isSea()
      ? !main.collection_teu_in_wh.isEmpty()
      : !main.collection_ton_in_wh.isEmpty()
  )
    selected_order = isSea()
      ? main.collection_teu_in_wh.pop()
      : main.collection_ton_in_wh.pop();

  if (
    !inIn() && isSea()
      ? !main.collection_teu_out_wh.isEmpty()
      : !main.collection_ton_out_wh.isEmpty()
  )
    selected_order = isSea()
      ? main.collection_teu_out_wh.pop()
```

```
18          : main.collection_ton_out_wh.pop();
19
20     is_busy = true;
21     resetconditionevent.restart();
22   }
```

Code Snippet 3 - Handling, idle state, internl transition

The process is similar to the one already documented for transportation *idle state,* as for example the preliminary check of collection to avoid a thrown exception. Here, in particular, if statements are used in conjunction with the ternary operator to select the correct order collections. Depending on the handling agent parameters, the agent selects one of the four alternatives: incoming teu (line 8), incoming ton (line 9), outgoing teu (line 17), outgoing ton (line 18). After that, it changes a variable to true to trigger a conditional transition to next state. Lastly, the event timer to trigger the reset transition is started, allowing for a precise compliance with the limit given by max_capacity in the specified timeframe.

When in *unload* (or *load*) state, after a message is sent to move the order and the transport load counter is decremented, another action has to be performed in addition.

```
1  if (isSea()) {
2    send("unload", selected_order);
3    selected_transport.actual_capacity--;
4    if (selected_transport.actual_capacity == 0)
5      deliver("isempty", selected_transport);
6    // selected_transport.onChange()
7  }
```

Code Snippet 4 - Handling, unload state, entry action

The handling agent checks whether the transport is completely unloaded (or loaded) and if this condition is met sends a message to it, in order to trigger its message transition. This last passage is mandatory since in AnyLogic conditional transitions for an agent are monitored only if the variable change is performed inside the same agent. If the change is performed externally, the condition is not monitored i.e. handling agent making a change to *actual_capacity* of a transport agent. Two solutions come up: either sending a message to the transport *(line 5)* or calling the method *selected_transport.onChange()* to make transportation agent aware of the change (line 6).

In state *to_wh* the following internal transition is executed:

```
1  for (Warehouse w : isSea()
2     ? main.warehouses_sea
3     : main.warehouses_air
4  ) {
5    if (w.actual_capacity < w.max_capacity && !is_found) {
6      send("movetowh", selected_order);
7      selected_order.assigned_wh = w;
8      w.actual_capacity++;
9
10     is_found = true;
11   }
12 }
```

Code Snippet 5 - Handling, to_wh state, internal transition

The handling agents searches for an available warehouse using a for cycle; in particular, the list through which it iterates is given using – as in the previous cases – isSea and the ternary operator, which returns line 2 if true, line 3 otherwise. When the first positive occurrence is found, a message is sent to the order to trigger the transition to warehouse state (line 6), the warehouse is stored in order's history (line 7) and warehouse's capacity is incremented (line 8). Finally, the boolean condition is changed to true in order to stop the iterative search and trigger the conditional transition of the handling agent to the next state.

In state *load* a similar process occurs, again through the following internal transition:

```
1  for (Transport t : isSea()
2    ? filter(main.ships, t -> t.inState(t.loading_out))
3    : filter(main.planes, t -> t.inState(t.loading_out))
4  ) {
5    if (t.actual_capacity < t.max_capacity && !is_found) {
6      send("load", selected_order);
7      t.actual_capacity++;
8      is_found = true;
9    }
10   if (t.actual_capacity >= t.max_capacity) send("go", t);
11 }
```

Code Snippet 6 - Handling, load state, internal transition

The handling agents searches for an available transport using a for cycle; in this case the particularity is that the list through which it iterates comes filtered, in order to select only the transport agents that are in state *loading_out*. Depending on isSea result the list selected is in line 2 or line 3. When the first available transport is found, a message is sent to the order to trigger the transition to *out_transport* state (line 6), the actual capacity of the transport is incremented (line 7) and the boolean condition is changed to true in order to stop the iterative search and trigger the conditional transition of the handling agent to the next state (line 8). Finally, for the same reason that holds for *load,* a message must be sent to the transport in case it is fully loaded to trigger its transition; alternatively, a conditional transition could be used for the transportation agent, in combination with *t.onChange()* .

4.3.5 Warehouse

Figure 15 - Model: Warehouse

The warehouse type is very simple and wouldn't need to be modeled as an agent, but it is treated like it just for convenience. Despite the fact that AnyLogic allows to create custom java objects, those do not present a graphical interface, and since the software doesn't make distinction between object and agents in practical terms, the latter are preferred.

No statechart needs to be used, since the only scope of a warehouse is to passively accommodate orders, thing that is possible by checking whether *actual_capacity* is lower than the *max_capacity.* As always *type* is used to distinguish between sea and air, while functions are used to collect individual statistics.

The further implementation of the Warehouse type, for example with internal processing, resource availability and physical space allocation depending on order's characteristics, is left as an open point for future development.

4.3.6 Link

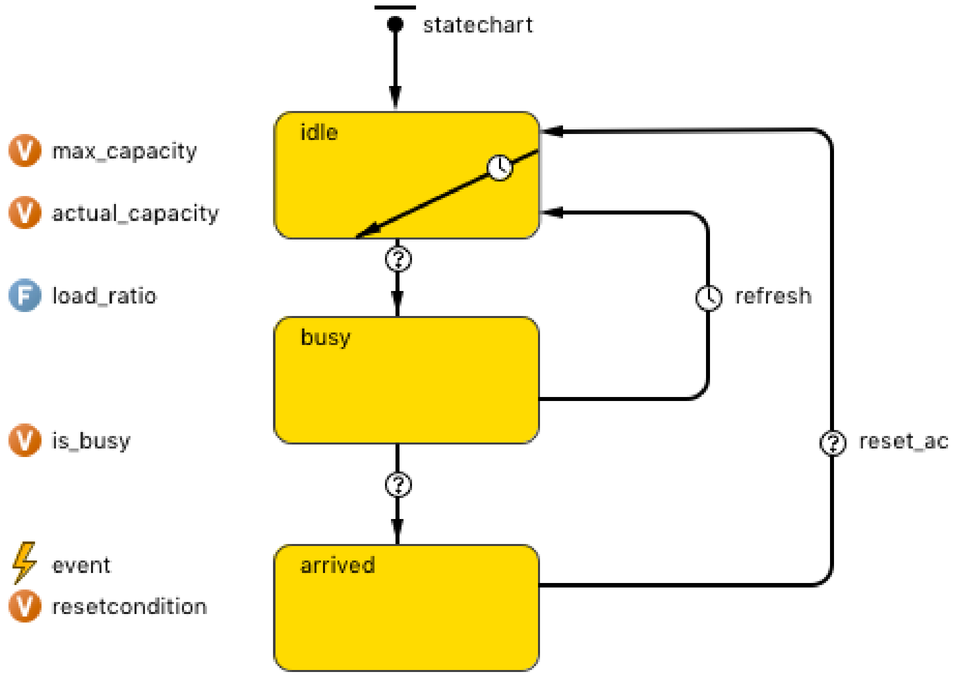


Figure 16 - Model: Link

Considering that the port and the airport are interconnected, the link serves as a connection point between the two ecosystems, taking care of sea-to-air and air-to-sea orders. This is obtained in conjunction with the boolean parameters *cross* and *ground_link* that are inserted in each order.

The former – if true – makes it possible that once an order has reached the warehouse, it waits for a message from a link resource to be moved to the state *link.* The timeout transition, that would move it to the outgoing sector, is never executed for crossing orders thanks to a guard-after-trigger boolean expression inserted in the transition. Hence, a teu (order agent) that arrives from sea incoming warehouse to the link is converted to ton and is promptly sent to the air outgoing warehouse; this also holds vice versa with tons converted into teus.

When a new order is generated, and it is the result of a conversion, the latter parameter makes it possible to automatically send it to the outgoing sector (dashed line) and insert it in *out_wh* state (initial pointer in composite state).

With the same logic that holds for types transport and handling, the internal transition in *idle* state is used to catch orders waiting to be converted

```
1 if (actual_capacity < max_capacity && !main.collection_link.isEmpty()) {
2   t = main.collection_link.pop();
3   send("link", t);
4   main.convert(t);
5
6   actual_capacity += t.isSea() ? 1 : 1.0 / main.conversion_ton_teu;
7   is_busy = true;
8   event.restart();
9 }
```

Code Snippet 7 - Link, idle state, internal transition

This is obtained, as usual, by sending a message to order (line 3), thus triggering its related message transition to the *link* state. Then, the actual capacity of the link is incremented of one teu unit, or its equivalent tons units transformed in teus, given by a user-defined conversion factor (line 6).

The real conversion takes place in line 4 by calling the main-level function *main.convert(t)*.

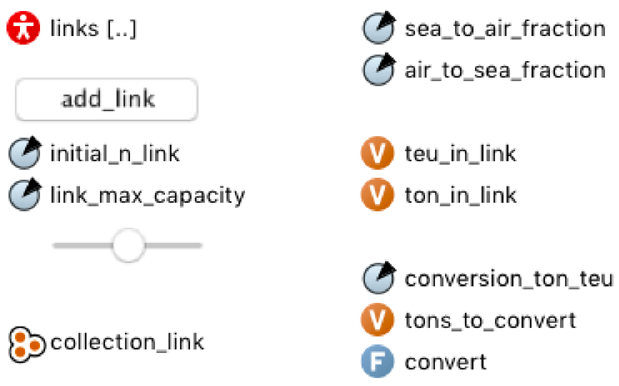


Figure 17 - Model: Main (2)

The function *main.convert(t)* receives the order to convert as argument and is briefly explained hereafter.

```
1  if (t.isSea())
2    for (i = 0; i < conversion_ton_teu; i++) {
3      new_order = add_tons("air", true, true);
4    }
5  else {
6    tons_to_convert++;
7    if (tons_to_convert >= conversion_ton_teu) {
8      New_order = add_teus("sea", true, true);
9      tons_to_convert -= conversion_ton_teu;
10   }
11 }
```

Code Snippet 8 - Main, convert function

In case the order to convert is a coming from the sea, the teu is broken apart and an equivalent number of tons are created; in line 3, the arguments in parenthesis refer to the parameters passed when creating the new order, respectively *type, cross, from_link.*

In case the order is coming from the air, many tons are mixed together in one teu, that is created upon reaching the conversion factor.

Chapter 5

Simulation and Validation

After having explained the idea behind the model and having documented the implementation inside AnyLogic, in this chapter simulations are carried out. The chapter is composed of two paragraphs of equal importance. The first shows and describes the functioning of the model through the use of a dashboard created ad-hoc. The second aims to assess the results in comparison with the System Dynamics model in order to validate it.
On the contrary, conclusion, implications and open points deriving from the discussion will be presented in the final chapter.

5.1 Simulation

The documentation of the previous chapter describes single agent interacting with each other in order to carry on a relatively simple process. However, it could become difficult to understand, from a practical point of view, the behavior of the system at an aggregate level. Agent-Based Modelling heavily relies on the emergence property, so the description of its single components sometimes is not sufficient to forecast what will be the results on the whole system.
For this reason, a practical tool is offered in order to facilitate this task: a dashboard, which is presented hereafter. In a single screen it is possible to have full access to all relevant KPIs that are used to monitor the system.
It features several graphs in order to monitor the situation at each time step and provide live insights. In fact, it evolves little by little following the simulation time, offering the possibility to make changes to system parameters with sliders and buttons and immediately see the consequences in real-time.

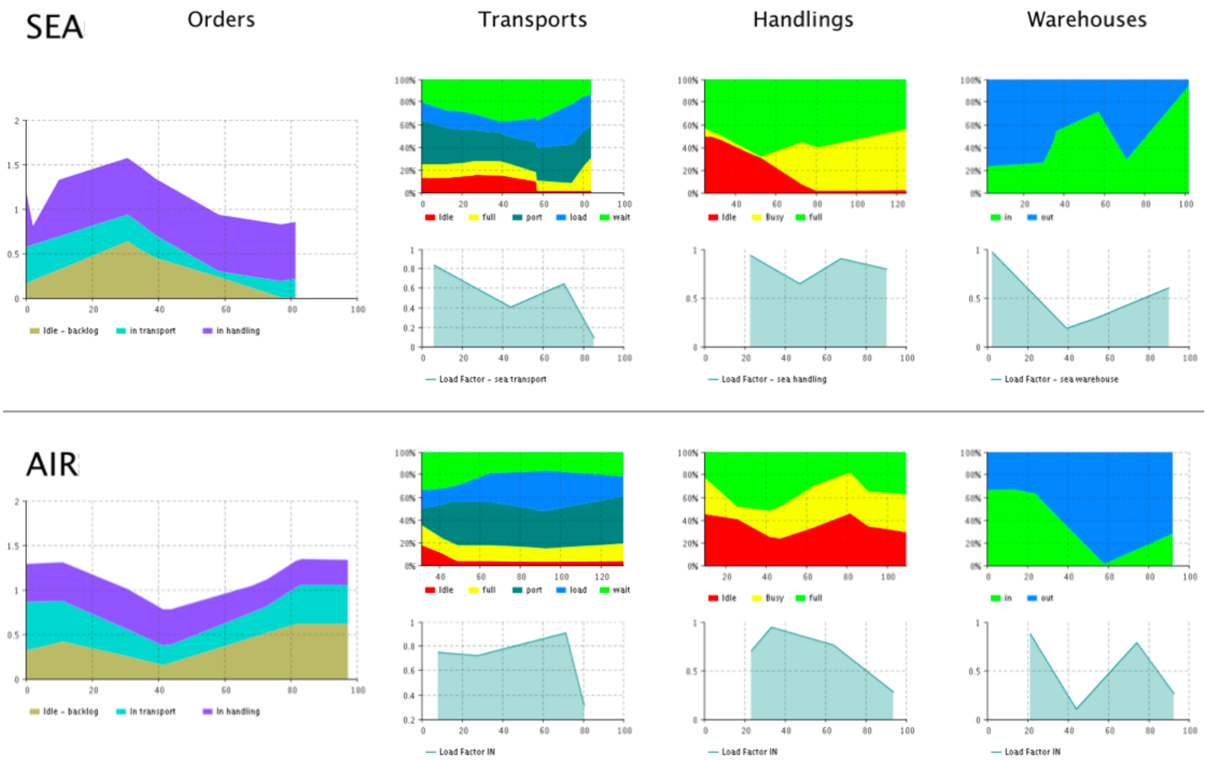


Figure 18 - Dynamic Dashboard

As the labels recall, the layout is organized as a grid: the top part refers to port's resources, while the bottom part refers to those of the airport system. Columns are used respectively to identify aggregate orders, transports, handlings and warehouses. Graphs can be grouped into three categories:

- **WIP**: The graph describing orders refers to the actual quantity in circulation in the system at each timestep, differentiating between orders in idle (or in backlog), loaded in transport or managed by handling agents. Once an order is successfully arrived at the warehouse, it is not counted anymore. Thus, representing the current work-in-progress (WIP) orders, graph follows the logic "lower is better".
- **STATE:** Colorful graphs in first and third row are used to quickly represent transports and handlings actual state in the statechart. In particular red is used to highlight resources in idle state i.e. not working because of excess capacity. The logic followed is "green is better", because in that way it

means that resources have quickly completed operations assigned and are just waiting for the reset to start again.

On the contrary, since warehouse has no states, the graph represents the composition of orders in it, distinguishing between incoming and outgoing orders. It should mainly show "green", but a possible problem can arise if the blue fraction increases considerably, meaning that *handling_out* agents are struggling to process outgoing orders, thus making it difficult to free up space in the warehouse.

- **LOAD FACTOR:** Light blue graphs in second and forth row represent the load factor of each resource, i.e. the ratio of the actual amount of order demand and the maximum capacity. As a consequence, a value close to 100% should be maintained. In fact, if the ratio is much lower the resource is currently experiencing excess capacity, while on the other hand if the ratio is greater the resource needs an upgrade, otherwise orders will start accumulating in backlog

With no doubt, the best way to understand its functioning is showing how it behaves with a real-case example and describe it alongside each of the 7 phases.

The example portrays how the increase in demand affects the whole system, by looking at the resources involved in the process, with a particular emphasis on the utilization rate. While the demand grows, additional agents need to be added to the system in order to compensate for the lost orders due to stockout. In this way, system capacity is affected and in turn load factor of other populations changes.

For simplicity, only teu orders are considered, but the following description applies also in case of tons orders, since the two ecosystems – port and airport – are modelled in a specular way and follow the same process logic.

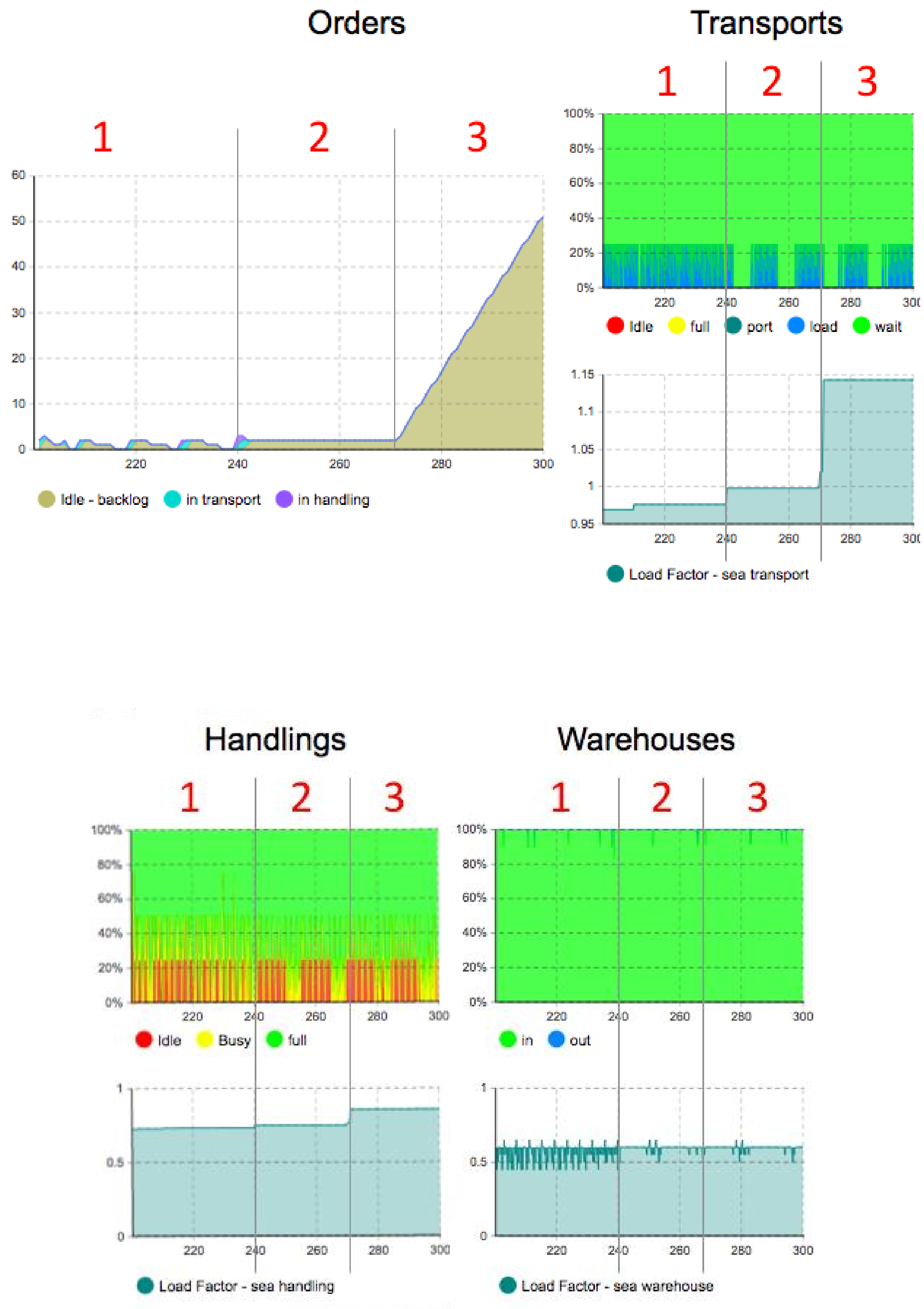


Figure 19 - Dashboard Example

1. Demand is lower than the actual system capacity, given by the minimum between the capacity of transports, handlings and warehouses. In this case the bottleneck is defined by transport capacity, which is the one that has the highest load factor. As for now, its value is still below 100% so that the demand can be completely satisfied.
   In fact, WIP graph value is very close to zero, indicating that no backlog is present, and orders are quickly delivered to destination.
   As for handlings, the load factor graph shows a quite low value, suggesting that the resource is underutilized with respect to its capacity. In fact, the corresponding state graph above shows that, on average, some resources are not working (idle, in red).
   Warehouse does not show any interesting behavior, apart from normal state graph oscillations given by the discrete nature of the paradigm being used.

2. Demand increases and transport load factor graph shows a value around 100%, meaning that the demand now is equal to the capacity of the whole system. In fact, WIP graph assumes the form of a rectangular, meaning that the average number of orders in idle state remains constant.
   Nothing relevant takes place for handling and warehouse resources, since the capacity limit is caused by transports.

3. Demand is greater than the capacity and transport load factor shows a value greater than one, meaning that an upgrade needs to be performed to avoid stockouts. For this reason, orders are starting to accumulate in idle state, and if no corrective measure is performed, they are soon going to be eliminated and counted as stockout.
   Again, nothing relevant takes place for other resources.

Figure 20 - Dashboard Example (2)

4. Keeping the same demand as in phase 3, no investment has been performed yet to increase the capacity of transport resources. Backlog orders have accumulated for a prolonged time in the backlog up to the point that their individual timer was triggered. So, the WIP does not grow anymore and stays around a steady level. Hence, inability to process backlog order for a long time so results in orders lost and accounted as stockout after their individual timer was triggered. An upgrade of capacity is considered necessary and cannot be delayed.

5. After new transport agents have been added to the system, the transport load factor decreases below 100%. So, the order backlog starts diminishing, and as a consequence stockout now stops. In particular, a non-negligible part of WIP is now composed of orders that are in state "in transport", meaning that they are now waiting to be processed by handling agents. In fact, since handling load factor is now the closest to 100%, this resource is generating the bottleneck for the whole system. As a consequence, the corresponding state graph shows that resources switched from idle to fully operational. This is in connection with transport state graph that shows that a considerable number of agents is waiting to be unloaded (dark green) or loaded (blue) by the handling agents. Warehouse faces an increase in the average number of orders to accommodate as a consequence of the rapid recovery of the backlog orders, because now there are more orders in the system.

6. New transport agents are added to the system in order to absorb the backlog orders more quickly. In fact, the slope of the WIP graph rotates inward, meaning that the backlog demand now needs less time to be processed, thus restoring the normal operating level of orders.
In principle it could seem a clever maneuver but in practice it requires an investment that is not essential for the actual moment now and will imply negative consequences.

7. As a natural consequence, in this phase the WIP graph is again under control, but the overinvestment of the previous phase now results in excess capacity for transport agents. The load factor remains the same as the previous phase since the demand is the same, but state graph shows a fraction of resources in idle state, meaning underutilization. At the end of phase 7 another handling agent was added, just to prove that an investment in a non-bottleneck resource only results in underutilization, as shown in state graph and load factor.

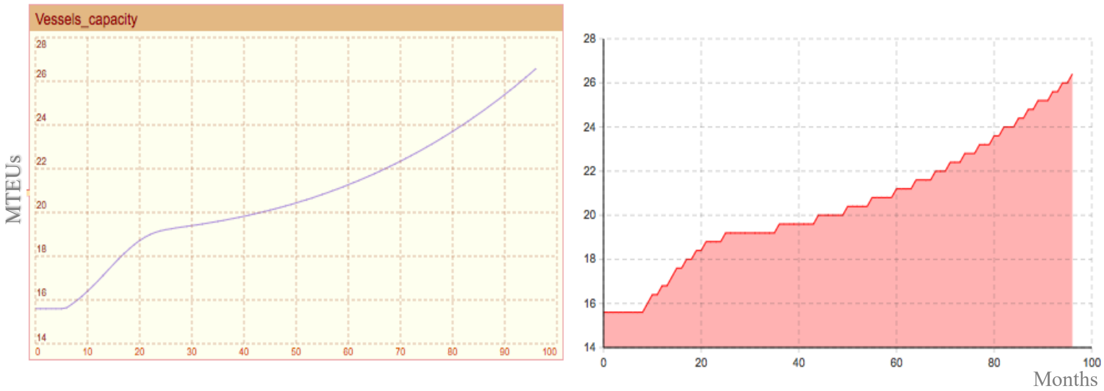
5.2 Validation

In simulation, the first goal of the system designer is to construct a system which can be representative of the real world that is being modelled. So, a fundamental step of the simulation process involves testing the results obtained with data from the past or similar models available. The scope is assessing that the model is as close to reality as possible, so that it could be safely used as a practical tool to support decision making. The importance of experimenting with a simulation model were already covered in chapter 2, while more advanced possibilities and tools will be briefly discussed in the future steps.

With the aim of validating the Agent-Based model, a comparison with the original System Dynamics model is performed. In particular, in order to have comparable results, the same initial data was fed into the system. This refers to not only to the initial variables and lookup tables, but also to the structure of mathematical operations carried out in the original System Dynamics model. To be precise, it is more correct to state that the System Dynamics and the Agent-Based model were merged together into a hybrid model.
As initially introduced in the re-conceptualization procedure, agents take the place of the stocks describing the resources generating the bottleneck, i.e. *Vessels capacity, Handling capacity, Warehousing capacity, AirCargo capacity, Airport handling capacity, airport warehousing capacity.* Hence, each of those is now expressed as the sum of the capacity of each individual agent, that is given by the following function: *sum(population, p -> p.max_capacity)*. It can be noticed that looks inside each agent of the population (being formally a for cycle for the population); this is just an example of what done on purpose to preserve agent's individuality, which allows to differentiate characteristics in them, as discussed in future steps.
The following results show a complete correspondence to the previous System Dynamics model, thus validating the model object of research.

Port resources:

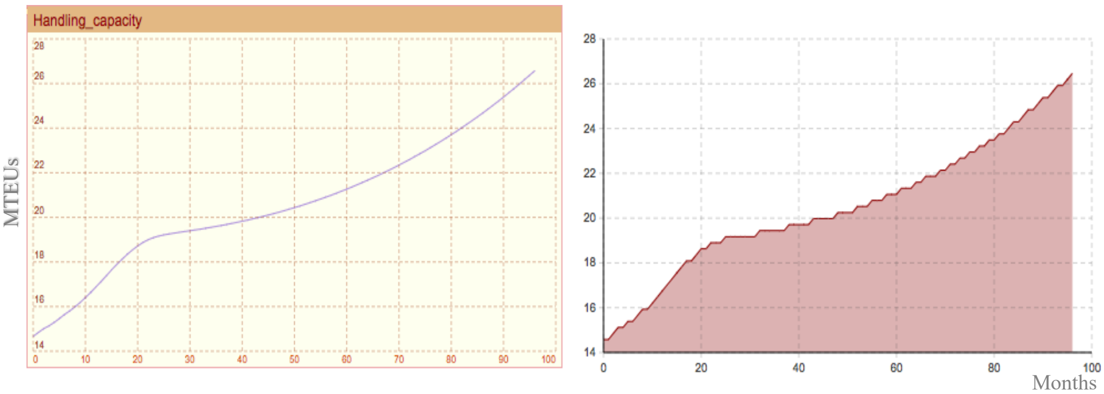
- Transport



System Dynamics *Agent-Based*

Figure 21 - Validation: Sea Transport

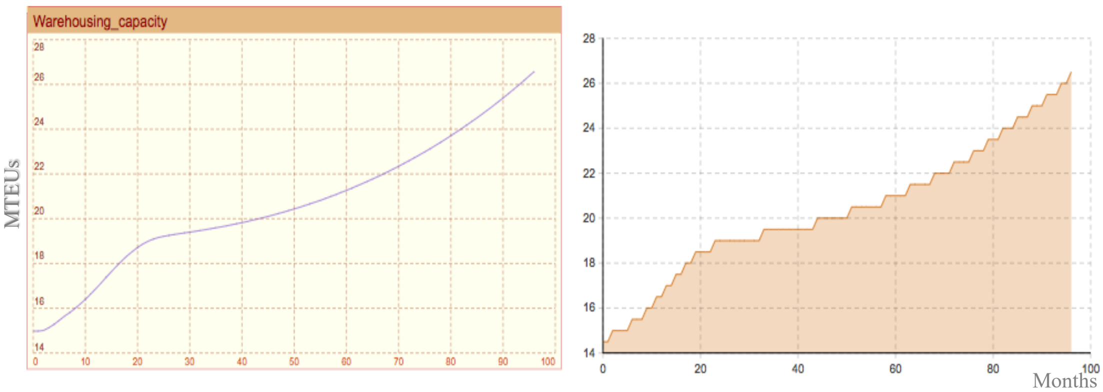
- Handling



System Dynamics *Agent-Based*

Figure 22 - Validation: Sea Handling

- Warehouse

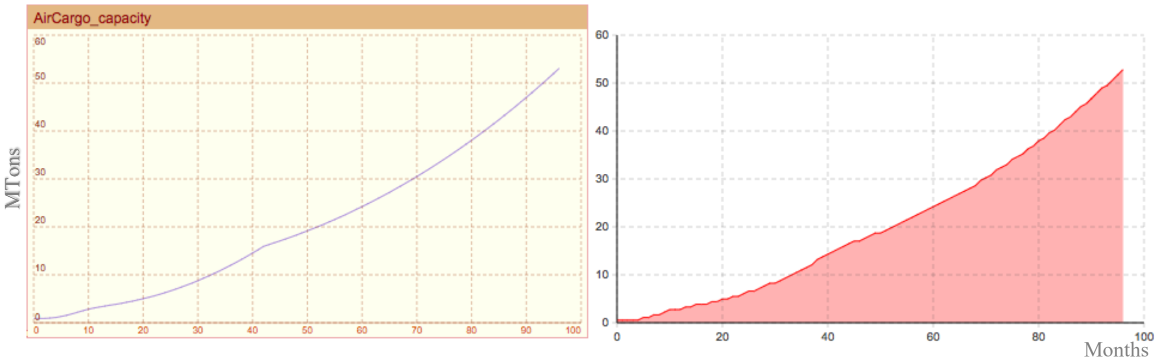


System Dynamics *Agent-Based*

Figure 23 - Validation: Sea Warehouse

Airport resources:

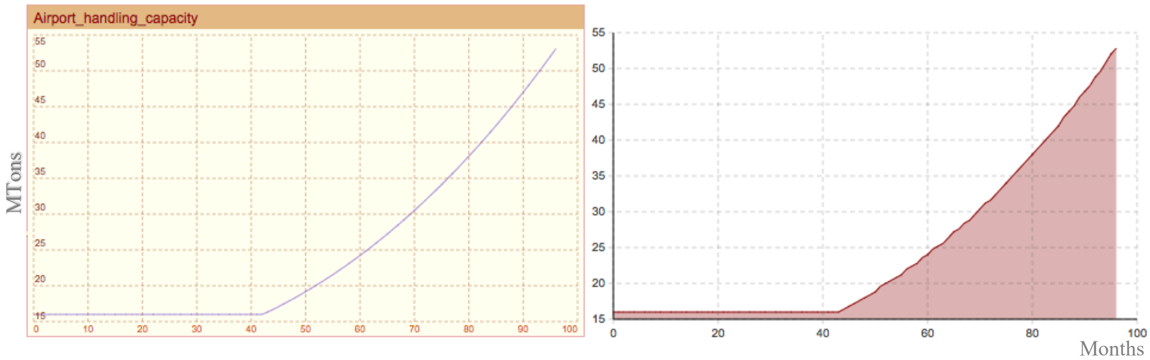
- Transport



System Dynamics *Agent-Based*

Figure 24 - Validation: Air Transport

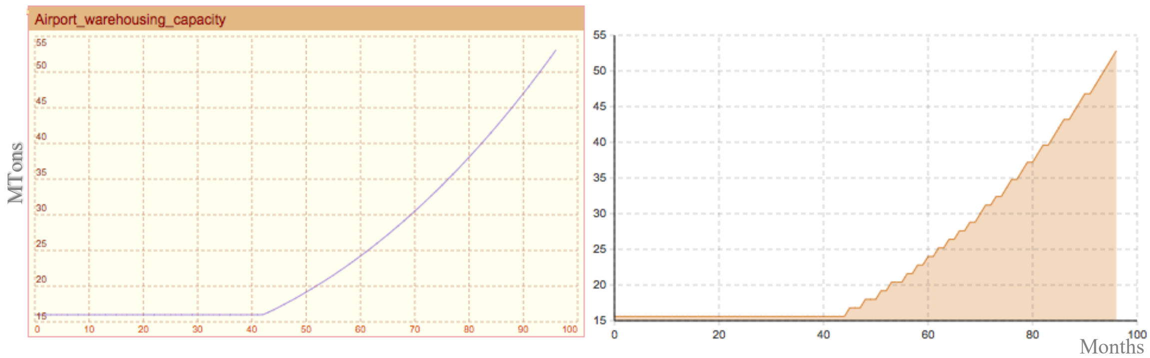
- Handling



System Dynamics *Agent-Based*

Figure 25 - Validation: Air Handling

- Warehouse



System Dynamics *Agent-Based*

Figure 26 - Validation: Air Warehouse

For the sake of brevity, only the graphs of one scenario were presented above for validation purposes. They refer to the "best case scenario", which features the following characteristics:

- Cross demand: Active
- Sea-Air Conversion: 15%
- Air-Sea Conversion: 15%
- Stakeholder Behavior: Risk Adverse

Test were performed in each scenario and the validity of the model is always verified, except for a small difference given by discretization. In fact, a clarification is necessary: the step behavior can be considered a characteristic of Agent-Based Modelling, given by its the discrete nature; on the contrary, System Dynamics yields continuous solutions.

The graphs presented in the previous study from Postorino may seem a bit misleading since they show a little stepwise behavior. They were obtained using Vensim®, whose internal engine is – probably – not so precise in comparison with more recent software.

After having imported the original model inside AnyLogic, simulations were performed and graphs were analyzed. For this reason, the System Dynamics graphs presented above show a perfect continuous behavior, as it is naturally expected from a System Dynamics model.

Conclusions

In the field of study of complex systems, Agent-Based Modelling represents a significant innovative step, given its ability to process more data and highlight individual behavior and interactions. This has been proven through research on history and development of the paradigm, the differences emerged between the academic and the practitioners' communities, the methodology and best practices, additional opportunities in comparison with traditional paradigms, drawbacks and pitfalls. In the recent years, Agent-Based is being used for a growing number of applications in a variety of fields and disciplines - as in the logistics sector - to solve problems that require deep data analysis and complex decision making.
Hence, a preliminary analysis of the evolution of logistics sector in Dubai has been carried out firstly to understand the available opportunities and the challenges to face. Starting from a previous model of the Dubai Logistics Corridor, developed using System Dynamics, the construction and validation of an Agent-Based model to deepen the study has been proven successful. The model follows a re-conceptualization process, allowing to exploit the additional functionalities of the new paradigm, which gives the possibility of deepening the analysis level.

However, it must also be clarified that Agent-Based Modelling does not always provide valuable insight for some applications, being less efficient, harder to develop or not matching the nature of the problem, that could be solved instead by using other less-powerful paradigms.
This is also connected to the fact that, in order to work properly, a sufficient amount of granular data should be provided as an input to the model, such to benefit its full potential. In fact, Agent-Based Modelling is for those who wish to get their hands dirty and go beyond the limits of traditional approaches; this is especially true in case the system being modeled contains active objects (people, business units, animals, vehicles, projects, stocks, products, etc.) with timing, event ordering or other kind of individual behavior. For that kind of systems, AnyLogic will not only enable to develop agents in an efficient way with minimum coding required, but

also will support migration from other paradigms and integration into a hybrid (or multimethod) model.

In the specific case of the Dubai Logistics Corridor, data availability from port and airport authorities was limited due to technical difficulties and business competition safeguard. For this reason, as for now it is pointless to add more complexity to the model by including other agent types (e.g. cranes, agvs, berths, aprons, etc.) or other variables (e.g. profits and losses). Hence, the model is kept quite general and this makes it possible that also other logistics hubs fit into the model. Even though System Dynamics alone has revealed to be a successful paradigm to investigate the situation at Dubai logistics corridor, Agent-Based model enables to scratch the surface and dive into data, proving to be an accurate proof-of-concept to model regional logistics hub in general. In addition, whenever low-level data becomes available, it becomes possible to - for example - distinguish origin and destination of routes and orders, discern individual capacity of transports, workforce and other resources, account for physical location movement and time availability, analyze the cost associated to each stockout or system slowdown, investigate the ground link development and the construction of new links to the region. Furthermore, AnyLogic offers the further opportunity to extract data directly from online databases, thus automating the data filling process up to the point that the model could represent the real system. Although possible, the main scope of simulation software is to provide a support tool for decisions that could also extend for a long time into future, rather than performing optimization in real-time or operations management, for which ERP software is more suitable.

The complexity of the world nowadays has risen the number and difficulty of decision that businesses have to face, but fortunately has also provided powerful instruments to deal with aforementioned complexity and uncertainty. In this context, Digital Transformation and Big Data represent a fundamental enabler of Agent-Based Modelling, giving the possibility to create a “Digital Twin” of the real system and perform accurate simulations to support data-driven decision making.

Acknowledgments

I would like to express my most sincere gratitude to Professor Alberto De Marco and Professor Hussein Fakhry, without whom I wouldn't have been able to undertake this inspirational experience. I kindly thank them not only for the precious teaching and help in developing the final thesis, but also for their support and trust.

I would also like to thank Professor Zakaria Maamar and Dr. Giovanni Zenezini for their insights and help, as well as Zayed University for its hospitality during the time spent in Dubai.

I am grateful for the support of my family, my friends and a very special person besides me.

References

Bin Kalli, Fernandez Nova, Mohammadi, Sanie-Hay, Al Yaarubi. (2013). The Dubai Logistics Cluster.

Axelrod. (1997). The complexity of cooperation: Agent-based models of competition and collaboration.

Bandini, Manzoni, Vizzari. (2009). Agent Based Modeling and Simulation: An Informatics Perspective.

Bonabeau. (2002). Agent-based Modeling: Methods and techniques for simulating human systems

Borshchev, Filippov. (2004). From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools.

Borshchev. (2013). The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6

Brito. (2016). Agent-Based Simulation for Yard Management in Container Terminal Operations

Brooks, Robinson. (2000). Simulation.

Démare, Bertelle, Dutot, Leveque. (2017). Modeling logistic systems with an Agent-Based model and dynamic graphs.

Eldabi, Brailsford, Djanatliev, Kunc, Mustafee, Osorio Muriel, Andres. (2018). Hybrid simulation challenges and opportunities: a life-cycle approach.

Franklin, Graesser. (1997). Is It an agent, or just a program? A taxonomy for autonomous agents.

Gorod, Hallo, Ireland, Gunawan. (2019). Evolving Toolbox for Complex Project Management.

Henesey, Persson, Davidsson. (2006). Simulation of operational policies for transshipment in a container terminal.

Jennings, Sycara, Wooldridge. (1998). A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems.

Sterman. (2001). System Dynamics Modeling, Tools for learning in a Complex World

Keenan, Paich. (2004). Modeling General Motors and the North American Automobile Market.

Macal, North. (2010). Tutorial on Agent-Based Modelling and simulation

Pal. (2015). Agent-Based Simulation for Supply Chain Transport Corridors.

Peng, Wei, Junqing, Bin. (2014). Evaluation of Airport Capacity through Agent Based Simulation.

Ribino, Cossentino, Lodato, Lopes. (2016). Agent-Based Simulation Study for Improving Logistic Warehouse Performance.

Saidi, Prasad, Scacciavillani, Roi. (2010). Dubai World Central and the Evolution of Dubai Logistic Cluster

Schieritz, Milling. (2003). Modeling the Forest or Modeling the Trees A Comparison of System Dynamics and Agent-Based Simulation

Stavash, Chadha, Wedgwood, Welsh, Parker, Teitelbaum. (2003). Agent Based Models for Logistics in Wargaming.

Sundarakani. (2017). Transforming Dubai Logistics Corridor into a Global Logistics Hub.

Vidal, Huynh. (2010). Building Agent-Based Models of Seaport Container Terminals.

Vuori. (2011). Methodological Issues of Agent-Based Modeling in Social Sciences.

Winikoff, Wagner, Young, Cranefield, Jarquin, Li, Martin, Unland. (2011). Agent-Based Container Terminal Optimisation.

Wooldridge, Jennings. (1995). Intelligent agents: Theory and practice.