Universitat Politècnica de Catalunya
Facultat d'Informàtica de Barcelona

Bachelor Thesis in Computer Engineering
Mention in Software Engineering

HPC applications for data-driven agent-based models of pedestrian movement

Quim Làzaro Gel
Advisor: Dr Josep Casanovas García
Statistics and Operations Research department
Tutor: Eng. Carla Diví Cuesta
High Performance Computing for Societal Challenges
Co-Tutor: Dr Iza Romanowska
Computational Social Sciences and Digital Humanities

Barcelona Supercomputing Center

June 23, 2020

Abstract

The management of large multinational facilities is a complex process involving finding the balance between customer satisfaction, safety concerns and commercial interests. This challenge is particularly pronounced in periods of transitions, such as stages construction work, facility upgrade and maintenance. However, with the growth of the Internet of Things (IoT) and unlocking of HPC for commercial endeavours in recent years offers to address this challenge through the use of technology. The increasing amount of data coming from these heavily monitored system, combined with AI and simulation techniques offers a new approach to the management of large facilities.

The Horizon2020 project: *IoTwins. Distributed and Edge-based Industrial Twins for SMEs: a Big Data Platform* aims to develop and enhance the use of so-called Digital Twins. This approach consists of building a simulation that uses data gathered by IoT devices in order to provide an in-silico model of the real system that can be experimented on. This simulated replica of a real systems give a wider knowledge on how the system will behave under different circumstances thus limiting the use of less efficient management techniques such as trial-and-error.

In this Bachelor Thesis' a Digital Twin of the Football Club Barcelona flagship sports venue: the Camp Nou has been developed. The most important aspect in the functioning of the venue are pedestrian flows and optimising them, ensuring robust emergency planning and managing change related to phased construction project involving a full renovation of the Camp Nou precinct are the main priorities. This prototype of Pedestrian Movement Model shows the feasibility of combining various data streams to represent multiple scenarios of crowd management. This model will serve as a baseline for integrating data coming from a number of sensors and preprocessed with Machine Learning techniques. This model will be integrated as a part of the whole IoTwins structure of test-beds 5 and 11 that focus on the FCB facilities.

The approach taken in the simulation part of the project is Agent Based Modelling. An increasingly popular simulation technique that is able to represent heterogeneous population consisting of individual Agents, for example, representing pedestrians. Their movement is defined with an specially developed algorithm that represent the space around the Agents as a mathematically derived cost function that combines multiple factors affecting the movement of pedestrians. The Agents enter the precinct, move around it and leave it following their independent paths. The model is validated and calibrated using currently available data. Several example scenarios have been run to show the feasibility of the approach for optimising emergency evacuation and construction-caused disruptions to normal operations.

Resum

La gestió de grans instal · lacions multiús és un procés complicat, que té a veure en trobar un equilibri entre la satisfacció del client, aspectes de seguretat i els interessos comercials. Aquest repte s'accentua en períodes de transició, com èpoques de construcció o la millora i manteniment de la instal · lació. Tot i això, amb el creixement de l'Internet of Things(IoT) i l'accés a HPC per a usos comercials als darrers anys ha proporcionat una manera d'adreçar aquest repte a través d'aquestes tecnologies. Les dades provinents d'aquests sistemes fortament monitorats, combinats amb IA i tècniques de simulació, permeten una nova manera d'abordar la gestió de grans instal · lacions.

El projecte de l'Horizon2020: *IoTwins. Distributed and Edge-based Industrial Twins for SMEs: a Big Data Platform* té com a objectius desenvolupar i potenciar l'ús dels anomenats Digital Twins. Aquesta tècnica consisteix en construir una simulació que usa les dades recol · lectades per dispositius IoT per generar un model informàtic del sistema real en el qual es poden realitzar experiments. Aquesta rèplica digital del sistema real proporciona una visió més àmplia de com es comportarà el sistema sota diferents circumstàncies, limitant d'aquesta manera l'ús de tècniques menys eficients de gestió com la prova i error.

En aquest Treball de Fi de Grau s'ha desenvolupat un Digital Twin del recinte insígnia del Futbol Club Barcelona: el Camp Nou. L'aspecte més important en el funcionament del recinte són els fluxos de vianants i la seva optimització, assegurar un pla robust enfront de les emergències i gestionar els canvis relacionats amb el projecte de construcció que consisteix en la renovació del recinte del Camp Nou també són prioritaris. Aquest prototip de Model de Moviment de Vianants mostra la viabilitat de combinar diverses fonts de dades amb l'objectiu de representar diversos escenaris relacionats amb la gestió de multituds. Aquest model servirà com a referència per integrar les dades provinents de sensors i preprocessades utilitzant tècniques de Machine Learning. Aquest model estarà integrat amb l'estructura de tot IoTwins dins els test-beds 5 i 11 que se centren en les instal · lacions del FCB.

El mètode escollit per la simulació del projecte és la Simulació Basada en Agents. Un paradigma de simulació cada vegada més popular que és capaç de representar poblacions heterogènies que estan formades per agents individuals que representen els vianants, per exemple. El seu moviment està definit per un algoritme desenvolupat específicament que representa l'espai al voltant dels Agents de manera matemàtica, derivada d'una funció de cost que combina diferents factors que afecten els vianants. Els Agents entren al recinte, es mouen seguint les seves prioritats i després en surten seguint el seu camí individual. El model es validarà i calibrarà amb les dades accessibles en aquest moment. Diversos escenaris d'exemple han demostrat la viabilitat del model per optimitzar l'evacuació en cas d'emergència i les afectacions que comporten les renovacions del recinte.

Resumen

La gestión de instalaciones multiuso es un proceso complicado, consiste en encontrar un equilibrio entre la satisfacción del cliente, la seguridad y los intereses comerciales. Este reto se incrementa en periodos de transición, como en situaciones de construcción o de mejora o mantenimiento de la instalación. Sin embargo, con el crecimiento de Intenet of Things(IoT) i el acceso a HPC para usos comerciales en los últimos años ha proporcionado una manera de abordar este reto usando estas tecnologías. Los datos provenientes de estos sistemas fuertemente monitorizados, combinados con IA y técnicas de simulación, dan lugar a una manera de afrontar la gestión de grandes instalaciones.

El proyecto del Horizon2020: *IoTwins. Distributed and Edge-based Industrial Twins for SMEs: a Big Data Platform* tiene como objetivo desarrollar y potenciar el uso de los Digital Twins. Esta técnica consiste en construir una simulación que usa los datos recolectados por los dispositivos IoT para generar un modelo informático del sistema real en el cual se pueden realizar experimentos. Esta réplica digital del sistema real proporciona una visión más amplia de como se comportará el sistema bajo diferentes circunstancias, limitando de esta forma el uso de técnicas menos eficientes de gestión cómo el ensayo y error.

En este Trabajo de Fin de Grado se ha desarrollado un Digital Twin del recinto insignia del Futbol Club Barcelona: el Camp Nou. La faceta más importante en el funcionamiento del recinto son los flujos de peatones y su optimización, asegurar un plan de emergencias robusto y gestionar los cambios relacionados con el proyecto de construcción que consiste en la renovación del recinto del Camp Nou son también prioritarios. Este prototipo de Modelo de Movimiento Peatonal muestra la viabilidad de combinar distintas fuentes de datos con el objetivo de representar múltiples escenarios relacionados con la gestión de multitudes. Este modelo servirá como referencia para integrar los datos provenientes de sensores y preprocesados utilizando técnicas de Machine Learning. Este modelo estará integrado con toda la estructura de todo IoTwins dentro de los test-beds 5 i 11 que se centran en las instalaciones del FCB.

El método escogido para la simulación del proyecto es la Simulación Basada en Agentes. Un paradigma de simulación cada vez más popular que es capaz de representar poblaciones heterogéneas que están formadas por Agentes individuales que representan a los peatones, por ejemplo. Su movimiento está definido por un algoritmo desarrollado específicamente que representa el espacio adyacente a los Agentes de manera matemática, derivada de una función de coste que combina distintos factores que afectan a los peatones. Los Agentes entran en el recinto, se mueve siguiendo sus prioridades y después salen siguiendo su camino individual. El modelo se validará y calibrará usando los datos accesibles en este momento. Múltiples escenarios de ejemplo han demostrado la viabilidad del modelo para optimizar la evacuación en caso de emergencia y las afectaciones que comportan las renovaciones del recinto.

Acknowledgements

Abans que res, volia donar les gràcies a la Carla per ser la millor directora de TFG que podria haver demanat i per tota la implicació que ha tingut en el projecte ajudant-me i guiant-me en tots els moments que ho he necessitat. En este sentido también le quiero agradecer Iza Romanowska por el hecho de pensar en mí como la persona indicada para asumir el reto que representa el proyecto IoTwins, gracias a su guía este trabajo no habría resultado así sin ninguna duda. He aprendido muchísimas cosas gracias a ella y estoy impaciente por seguir trabajando con ella por lo mucho que puedo aprender de ella. No podia faltar el Josep Casanovas per acceptar ser el ponent del meu treball i per tot el guiatge, tant acadèmic i laboral, com personal, que ha suposat per mi treballar amb ell durant aquests dos anys i mig. También quería nombrar a German por ayudarme con Python, ya que la configuración del lenguaje de scripting y yo no somos muy amigos y me ha ahorrado muchos dolores de cabeza.

També volia donar les gràcies a la Marta Cuatrecasas i a la MªPaz Linares per tombar-me a l'entrevista que vaig fer ja farà gairebé dos anys i mig per entrar a l'inLab, ja que gràcies a això i a la recomanació de la Marta vaig poder accedir al meu actual lloc de treball i com expresso en les conclusions descobrir una branca de la Informàtica a la qual em vull dedicar. També els voldria agrair que tot i no estar contractat a inLab m'incloguessin en el programa Talent, ja que m'ha permès conèixer a persones increïbles que s'han convertit en grans amics amb els quals he compartit molts moments genials. En Mario, Victors(Tomàs, Diví i Sanchy), Gina, Mathi, Gis, Marta, Josep, Albert, Juan, Alexs(Mas, Catalan i Català) Gonzalo, Sergio i ja paro que si no es farà infinit. Moltes gràcies a tots per fer-me part d'aquest projecte també i regalar-me tants moments.

Relacionat amb això moltes gràcies a tot el personal del bar de la FIB per la inestimable feina que fan, sense vosaltres crec que no haguéssim aguantat cap dels de dalt. Merci per tot Jose, Jaime, Gerdardo i Santi.

Aquí no podien faltar la meva família que sempre m'ha estat fent costat i donant suport a les meves decisions. En especial als meus pares, que després d'un primer quadrimestre desastrós em van convèncer perquè ho intentés un segon quadrimestre i gràcies a ells i a aquesta empenta he arribat fins aquí.

Finalment, ara toca "la família que es tria", encara recordo el tercer dia que vam tenir la fatal idea d'anar a fer una birra al bar i des de llavor hem gaudit i patit la carrera junts. Merci per ser com sou Xexu, Marcos, Jordi, Nico, Adri, Santi, Mario, Víctor(sortiu dos vegades vosaltres dos no us queixareu) i Ballu. Heu fet que això fos tan divertit com podria arribar a ser.

Contents

List of Tables

List of Figures

Glossary

ABM Agent Based Modelling.

AI Artificial intelligence.

BSC Barcelona Supercomputing Center.

CASE Computer Applications in Science and Engineering.

CSV Comma-Separated Values.

CUDA Compute Unified Device Architecture. Used to program accelerated applications.

FCB Fútbol Club Barcelona.

GPU Graphics Processing Unit.

HPC High Performance Computing.

HPC4SC Hihg Performance Computing for Societal Challenges.

IoT Internet of things.

KPI Key Performance Indicator.

ML Machine learning.

MN4 MareNostrum 4.

MPI Message Passing Interface. Framework used to define the message flow between more computer nodes executing the same application.

MSE Mean Squared Error.

OpenMP Framework used to program multi-thread applications.

PANDORA PANDORA is an Agent Based Modelling framework developed at BSC.

UML Unified Modeling Language.

XML eXtensible Markup Language.

Chapter 1

Introduction

This Bachelor Thesis for the degree on Computer Science with a mention on Software Engeneering, from *Facultat d'Informàtica de Barcelona* has been developed within the B modality with a *Conveni de Cooperació Educativa* at the *Barcelona Supercomputing Center* (`BSC`). The advisor of the project is Dr Josep Casanovas García of Statistics and Operations Research department and the project has been tutored by Eng. Carla Diví Cuesta and co-tutored by Dr Iza Romanowska, both researchers at the `BSC`.

The main objective of this Thesis is to design and implement a prototype of a general digital twin simulation model, in order to represent the pedestrian movement dynamics that take place at a large sporting venue. This digital twin will be used to optimize and detect flaws in the real system. This implies a conceptualization and formulation of the model, a research of how this type of scenarios have previously been modelled, the implementation of the model and its validation as well as the experiment design and output analysis.

Since the computational cost of the execution of this type of model is expected to be high, this model has to be designed to run in a High Performance Computing environment to give results within a time frame appropriate to the decision making process.

1.1 Context

This Bachelor Thesis is part of the Horizon2020 project: *Distributed and Edge-based Industrial Twins for SMEs: a Big Data Platform IoTwin*, also known as IoTwins. IoTwins' main objective is to build a reference architecture for the development and deployment of distributed and edge-enabled digital twins of production plants and processes.

The IoTwins project is structured as a set of test-beds. The test-beds consist of pilots formed by two partners: an enterprise and an academic institution. This Thesis contributes to test-beds 5 and 11 formed by the Barcelona Supercomputing Center (`BSC`) and the *Fútbol Club Barcelona* (`FCB`). The approach proposed by the partners in test beds 5 and 11 focuses on the use of Edge Computing, Internet of Tings (`IoT`), High Performance Computing (`HPC`), Agent Based Modeling (`ABM`), Artificial Intelligence (`AI`) and Machine Learning (`ML`) techniques. Their combination is used to represent how pedestrians move through the venue space and how the pedestrian flow will be impacted by the construction work and infrastructure changes within the *Espai Barça* and inside the

Camp Nou stadium.

The primary contribution of this Thesis to the project will be the design, specification and development of a general `ABM` model representing pedestrian flows. This benchmark model will serve as a baseline for a range of scenarios parametrized using data collected through the sensors on the system and other means. In each scenario the baseline model will be modified to better represent the tested circumstances, for example a match day versus a non-match day

1.1.1 IoTwins project

The IoTwins project aims to design a reference architectures for a series of distributed and edge-enabled digital twins for a range of industries. It consists of implementation, deployment, integration, and experimental in-the-field evaluation of several test-beds divided into: manufacturing test-beds, facility test-beds and replicability test-beds[1].

A digital twin is a digital replica of a living or non-living physical entity, see *Digital Twins: The Convergence of Multimedia Technologies*[2]. These models aim to be as analogous representation of the system as possible. This allows to understand the dynamics of the system itself, perform predictions, find and correct faults and optimize the operations. This type of representation is specially interesting because the data is collected directly from the represented system and in real time in addition to the data available from the facility management. This allows the twin to be characterized more appropriately than with other data gathering techniques that do not follow the system constantly.

Using the digital twins as the reference approach, the IoTwins project is divided into twelve different test-beds of three different classes: test-beds in the manufacturing sector with the goal of optimizing production quality and plant maintenance, test-beds for the optimization of facility management and test-beds for the in-the-field verification of the replicability, scalability, and standardization of the proposed approach, as well as the generation of new business models.

1.1.2 Test-beds 5 and 11

Test-bed 5 is concerned with a sport facility management and maintenance and test-bed 11 is a replicability-feasibility study for smaller scale facilities. This is a high risk part of the project since facility management digital twins are much less developed compared to the more common manufacturing solutions.

- **Test-bed 5**: focused on the management of facilities involving the flow of large crowds, both during normal operation and during construction phases related to maintenance and upgrade of the venue. The pilot aims to analyze how crowds move using `IoT` collected data in *Espai Barça* and inside the *Camp Nou* stadium. This will be achieved by designing and implementing a general Agent-Based Model (`ABM`) that represents the system described by the data and using various `ML` models to provide input and validation data for the model. The `ML` models will be trained continuously using the data collected and comparing it with the output of the simulation. However, in the early stages of development the `ABM` model will need to be calibrated before training the `ML` models. This model will be used to analyze the general characteristics of the flow of pedestrians in several scenarios that can take place within `FCB` facilities. These scenarios will be defined by the stake holders and parameterised with the data obtained from the current state of the real system. The need for several scenarios stems from the fact that pedestrian flow differ significantly on a workday and on a match day, thus each of them needs to be represented as a separate scenario.

- **Test-bed 11**: this test-bed takes place within more closed spaces or with predetermined paths. Buildings of *Espai Barça* give us very different scenarios, *Johan Cruyff* stadium or the *Barça museum* for instance are different systems that match the previous description. The goal of this test-bed is to demonstrate that the model defined and implemented in test-bed 5 can be transferred into smaller venues. Its important because a general crowd model must be able to define not only large open spaces, malls or stadiums, but also this closed spaces or systems with not much freedom of movement. In essence the test-bed wants to test the scalability and replicability of the previous model.

1.1.3 Architecture of test-beds 5 and 11

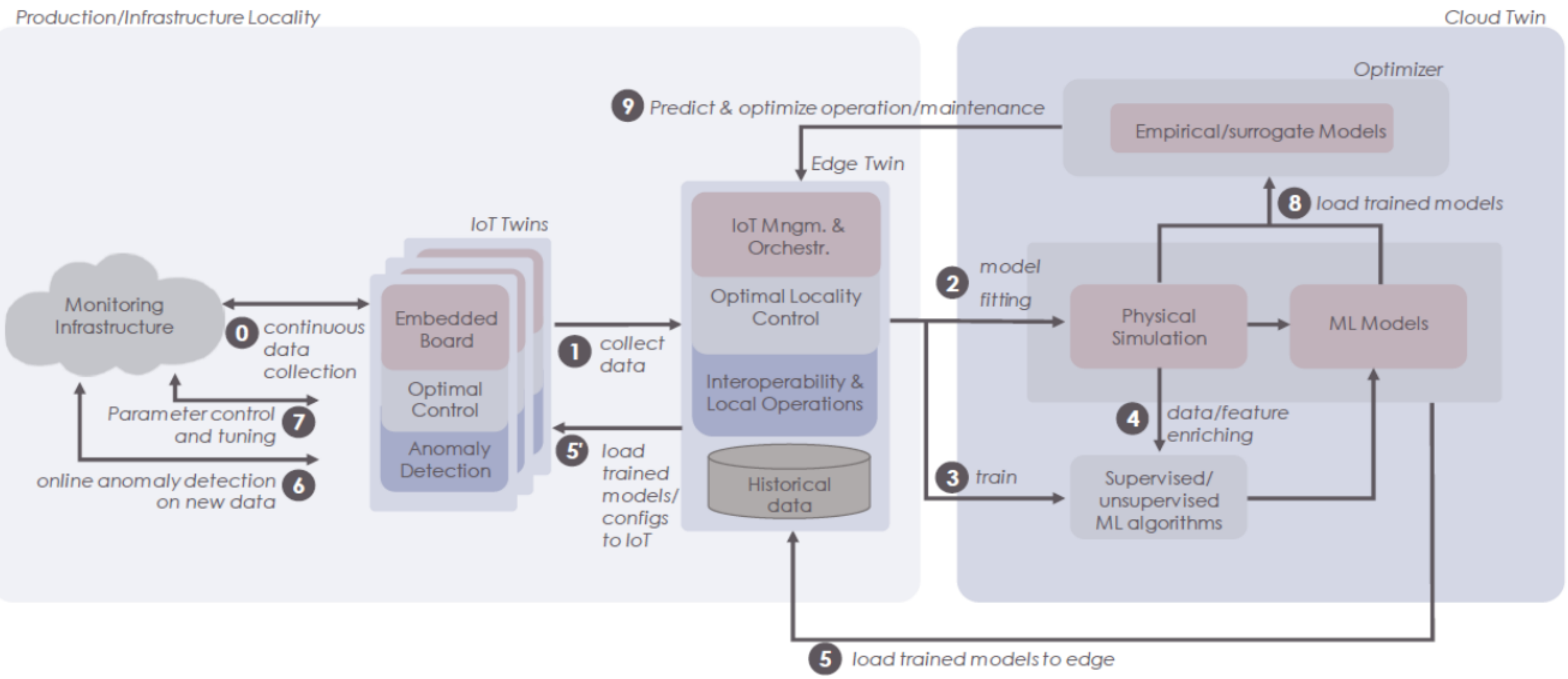
The architecture planned for the project is shown in Figure1.1. The architecture is divided on two different layers, Production/Infrastructure Locality and the Cloud Twin. The data gathering and characterization is done in the Production/Infrastructure Locality layer. This layer is divided into three different components: i) the Monitoring Infrastructure, ii) the IoTwins and iii) the Edge Twin.

The Monitoring Infrastructure is formed by all of the `IoT` devices, such as cameras or turnstiles. All of the data gathered and anonymized and sent to the Embeded Board for processing. Some of these devices are Edge Computing devices. The cameras, for instance, have a `GPU` that runs a trained Neural Network that analyzes in running time characteristics of the system: the velocity of the pedestrians, or the number of people in a group, etc. The raw data in the IoTwins component serves several purposes. Its main purpose is to monitor the system through the data flow, which can detect anomalies and alert the system manager on the Edge Twin to analyze them. The optimal parameters of the Neural Networks are set up on this component as well. Finally, the necessary anonymized data is sent to the Edge Twin.

The Edge Twin layer orchestrates all of the system deployed within the monitored and modelled facilities. This eases the management of the operations department of the client. The database of historical data is located here to quickly detect and classify anomalies. The interpreted data processed by the Optimal Locality control component is sent to the Cloud Twin to calibrate the `ABM` and fit the `ML` models.

The `ABM` model will simulate scenarios using data gathered at Production/Infrastructure Locality layer. On the Cloud IoTwin the `ABM` simulation will be run using the multiple scenarios defined by the team using the data collected on the first layer. This data and the output of the simulation runs will be used to train `ML` algorithms. The objective is to be able to predict the results of the simulation. This will allow a faster way to obtain the simulation results only using the `ML` algorithms. Finally using the output of the models, both `ABM` and `ML`, the stakeholder will be able to take decisions and optimize the management and disposition of the facility.

The high-level Platform architecture



IoTwins

Figure 1.1: Architecture diagram of test beds 5 and 11. Source: produced by the BSC Data Analytics and Visualization team for the IoTwins Project.

1.1.4 Terms and topics

In the following section I present a list of definition of key concepts and topics to facilitate a fuller understanding of the project.

- **Agent Based Modeling**

  Modelling technique that conceptualizes a system as a set of entities: Agents. These entities perform actions and change the state of other agents, themselves and the environment around them. The agents interactions drive the simulation dynamics.

- **Framework**

  Group of models and programming practices that offer a predetermined structure for designing software.

- **High Performance Computing**

  Execution environment of high computational power, typically a supercomputer or a cluster of graphics processing units. In this case, we use the BSC's supercomputer: MareNostrum 4.

- **PANDORA**

  PANDORA is an Agent Based Modelling framework developed at the BSC[3] and optimised for HPC applications.

- **Simulation**

  Technique that represents a real world system as a digital model and experiments with it and its operations [4].

- **Simulation**

  Conceptualization of a real live system as a computer program.

- **Test-bed**

  Group of companies or research centers that work together on a concrete use case of a project, like the modelling of the systems or the architectures of communication among different partners, etc.

1.2 Stakeholders

The five principal stakeholders involved in the project are `FCB` and the `BSC` who are direct research partners working on the project together, the other IoTwins partners who are less directly involved but still interested in the outcome and the end users of the model and its results: a building company who can use to model to optimise their operations and, finally, pedestrians at the Espai Barca who will benefit from an improved experience even if they may not be directly aware of the project.

Fútbol Club Barcelona

Test-beds 5 and 11 take place at the venue and facilities managed by `FCB`. The space concerned is composed of single level outdoor space known as *Espai Barça* and the multilevel building of the Camp Nou stadium. The *Espai Barça* is open for anyone during the week, making it a semi public space similar to a pedestrianised street. On match days the `FCB` closes the space from public access and uses it to manage the crowds and to optimise the check-in for people entering the stadium. The same limitation of access occurs at other facilities, depending on the event taking place.

In 2019, `FCB` has started a five-year project redesigning Espai Barca and upgrading the Camp Nou stadium. The current software and data analytic techniques used to manage pedestrian flows during normal operations are unable to give satisfactory information in how the changes caused by the construction work will affect pedestrian movement. Importantly, the currently used solutions cannot provide results within a short amount of time, since the computation time is more than a week. This is a major issue given the dynamic nature of any construction projects where plans and people, material and machine deployment may change from one day to another.

`FCB` wants to be able to get information of how the pedestrian flow will change if certain areas are renovated or restructured and therefore cut off from pedestrian access. This kind of information would enable the FCB managers to change the priority of some actions and better manage their facilities during the complex phase of construction.

Barcelona Supercomputing Center

`BSC` is the national research center for High Performance Computing in Spain. It is the principal partner of the test-bet in charge of installing and monitoring the edge devices in `FCB` facilities, developing the `ML` algorithms and data post processing of the information of the edge devices. It is also responsible for the development of a data-driven ABM simulation model capable of running various experiments with the data collected by the devices.

At the `BSC` there are two teams from the `CASE`: Computer Applications to Science and Engineering department and one team from the CS: Computer Science department who are involved in the project.

- **`CASE` Data Analytics and Visualization**: this team is responsible of the gathering and analyzing the data necessary to train the `ML` models that they will develop. This includes the installation of the edge devices and the infrastructure of management of the facilities. These models will be used as input, calibration and validation datasets for the execution of the model implemented by the `HPC4SC` group. The responsible for this team is Dr Fernando Cucchietti.
- **`CASE` Computational Social Sciences and Digital Humanities**: this team is in charge of developing and updating the simulation engine in order to satisfy the project's necessities. Moreover, they will be assisting with the `ABM` model design, validation and experimentation. The responsible for this team is Dr Iza Romanowska.

- **Computer Science High Performance Computing for Societal Challenges**: this team is in charge of designing, developing and documenting the new `ABM` model and the communication between the data layer and the simulation model. This team will also define the various scenarios representing the different facilities and situations identified. The author of this Bachelor Thesis is part of this team and will be responsible of the design, development and documentation of the model itself. The team is responsible for the validation of the model and the experiment design as well. The responsible for this team is Dr Josep Casanovas.

Other IoTwins partners

Since the IoTwins consortium has several partners involved in other test beds they are interested in sharing their progress and sharing their approach to solve common problems, as well as learning from each other to improve the performance of their test-bed. Once the platform is ready the partners will have access to the project of test-beds 5 and 11 and will be able to test the representation of their systems using the tool developed by `BSC` and `FCB`.

Pedestrains

The term 'pedestrians' includes the entirety of people that walk within the modelled `FCB` facilities. Ultimately, they are the 'end user' of the project outcomes since its main goal is to optimise the operations of the venue including increasing fluidity of pedestrian movement and limitation of unnecessary agglomerations. Pedestrians will benefit from the project since their mobility will be taken into account during the renovations and changes happening in the different spaces during all construction phases. Moreover, the emergency evacuation will be represented as one of the scenarios in this project, to enable adjustments, if necessary, to the evacuation plan.

Building Company

The company hired by `FCB` could modify, if necessary, the order and priority of their road-map in order to keep a satisfactory pedestrian flow during the renovation depending on the results of the experiments.

1.3 Problem definition

In the introduction we have defined that the main topic of this Thesis is the elaboration of a simulation model. This model is a general tool able to represent multiple pedestrian movement systems parameterised by data. Thus, the issue that will be solved here can be categorized as Pedestrian Movement Modelling.

We understand Pedestrian Movement Modelling as a simulation model that represents a large group of heterogeneous individuals and studies how they behave and interact among themselves and with their environment. In particular Pedestrian Movement Modelling studies how individuals move through a delimited space, how they move individually and as a group as well.

This project considers four tested scenarios, a non-match day in *Espai Barça*, a match day in the same space, the movement of visitors of the museum and a match day at the *Johan Cruyff* stadium. These distinctive situations will be studied in order to plan the renovations and reduce its negative impact on pedestrian flows, detect bottlenecks that can occur during the different situations and optimize the space distribution. This optimization has as its goal the fluidity of pedestrian traffic within the facilities.

1.4 Objectives

Following the goals of the IoTwins project and the test-beds 5 and 11 the main objective of this Bachelor Thesis is to **design, specify, implement and validate a prototype of a general simulation model that represents the pedestrian flow within a determined space using real data**.

Since this objective is quite complex a list of sub-objectives will be introduced. If all of them are accomplished, the main objective will be considered as achieved:

1. Design and specify a prototype of a general `ABM` model that represents the pedestrian behaviour within a given system to be implemented using the framework PANDORA.
2. Define and implement the behaviour of a single pedestrian and extrapolate it to a population.
3. Define and implement the group dynamics among the pedestrians.
4. Use data to run the experiments.
5. Define and implement the no match day scenario.
6. Experiment and validate the no match scenario.

1.5 Scope

The development of the project at `BSC` will involve the whole `IoT` and Edge devices infrastructure, as well as the development and training of the `ML` models for data profiling by one of the teams. Special attention will be devoted to the communication with the cloud computing services where the `ABM` model developed in this Bachelor Thesis will be executed to perform the experiments.

The scope of the Thesis will limit to the specification, design and implementation of a prototype of a general pedestrian movement model. This prototype must be able to represent various situations depending on the input. This input must drive and characterize the simulation and the scenario represented by the data. This Thesis will include the experimentation and initial calibration and validation of the prototype produced.

1.6 Methodology

To develop the project, the `HPC4SC` team has opted to use agile methodology with iterations of two weeks, based in Scrum [5]. Since the team is formed by three people working on the same office daily meetings are held to keep up with each other's advances. This methodology enables the team to have continuous feedback, as well as short achievable goals that speed up the development. To keep the other `BSC` teams involved in the test-beds 5 and 11 project monthly meetings are held, and each six months all partners meet to discuss the advances of their test-beds.

Validation process will be accomplished through the execution of several experiments, the first ones with mocked data will be performed during the development at each of the models stages. When each one of the features of the model have been partially validated successfully the validation process will be done with real data. The final validation will be supervised by the director and co-director of this Thesis and performed in collaboration with other members of the Data Analytics and Visualization team at `BSC` to limit errors.

1.6.1 Project monitoring tools

In this section the different monitoring tools used in the project management will be described.

- **Version control** The version control is done using Git [6], an open source solution that is widely used in the `BSC`. Git has a record of the changes made in a monitored project in a repository, the repository holds all versions of the code and multiple versions of the code can co-exist in different branches. Github [7] (Microsoft) will be used as a repository manager.

- **Trello** Trello [8] is a project management tool developed by Atlassian. It allows to create tasks, update their state, and manage the project backlog. Trello is used to plan sprints and manage the constant development of the project.

Chapter 2

Problem Study

To start the specification the system and the conceptualization of the general model a wide research on the topic has been done to incorporate all of the common features to the initial model design in order to make the model as complete and reliable as possible. Other simulation techniques will be explored to extract features from those methods as well as justifying why `ABM` is the chosen method for the development of the digital twin.

The different `HPC` techniques used to improve the performance of `ABM` models were reviewed focusing on the question how to face the high computational cost that this simulation approach implies. Finally I discuss and justify why ABM was chosen at the end of the section.

2.1 State of art: Pedestrian Movement Modeling

Many simulation techniques could have been used to represent systems of Pedestrian Movement Modeling. These techniques are aptly summarized in this editorial *Environment and Planning B: Planning and Design*[9]. In this section the most used techniques will be discussed and compared with the selected solution, Agent Based Modelling.

Cellular Automata

Cellular Automata used to be a popular simulation method in the past. Most of the techniques described in this section are redesigns or refinements of this generalised approach. These techniques, like Fluid Dynamics or `ABM`, provide a more realistic representation of the real system but require higher computational power and are more difficult to track and analyse. CA is one of the most simple and 'elegant' way of simulating a system. Thus even now features of various models are implemented as Cellular Automata inside another model.

This approach consists in defining a grid of cells representing the simulated system. The cells then update their value following defined rules. This grid contains the values of the behaviour that is chosen to be studied, e.g., for Pedestrian models the number of people. Each cell is the same size as all the other cells, e.g., if one cell represents a space of 0.3m all of the cells are equally large. Depending on the value of the neighbour cells and other rules of the model cells update their state at each time step of the simulation [10]. This update function in crowd models represent how the pedestrian flow evolves during the simulation, how pedestrians change their position and how do then interact with each other.

These models are really efficient and easy to implement, but it is difficult to define conceptualization of a complex systems into simple cell values and to define the update function to reflect realistic behaviour. Since the information is aggregated per cell, these type of models are not suitable for achieving the IoTwins project objectives. The definition of an IoTwin demands a much finer representation of the system, and this approach can only represent the general behaviour of the system.

Fluid Dynamics

This approach defines pedestrian movement as a system similar to a fluid. It represents the pedestrian flow using methods developed to study liquid that goes through a system.

In Fluid Dynamics (FD) the topology is represented as a system of pipes. The space where pedestrian transit is allowed is transformed into pipes and the entrances as sinks that let the particles of the fluid trough these pipes. The particles interact with each other, creating the flow and pushing one another in different directions. This interactions can be lead to turbulence - a phenomenon often observed in Pedestrian movement studies *Crowd Turbulence With ABM and Verlet Integration on GPU Cards*[10]. When a turbulence is generated it is said that a point of conflict has been found in the system. Thus we can detect the bottlenecks of the system by observing the flow of the particles in the pipes.

This kind of models are particularly useful when we want to understand how a system works on a general level and detect potential bottlenecks. However, FD suffers from the same limitation as the Cellular Automata models - human behaviour is represented in an aggregated form and full homogenity of the population is assumed. The IoTwins demands a finer detail of representation of human behaviour. In the real world, for example, it often happens that a person, or a little group, decides to go against the general flow which will produce a turbulence not predicted by a FD, as the individual preference is enough to create a conflict.

Event Based

Many systems, like a factory, a train station or a port terminal, can be represented as a succession of events occurring in the system. Pedestrian Movement Models can be implemented using this technique as a micro simulations with a lot of detail [11].

This type of simulations define the system as a set of resources, processors and entities, e.g., pedestrians. They can only update their position state by using processors or going through checkpoints previously defined on the system. Thus, the actions of pedestrians are the ones that drive the simulation *Spatial activity-based modeling for pedestrian crowd simulation.* [12]. The entities control the events generated as queues and execute them. They do not walk actually in the space, they move from processor to processor by updating their position by the processing of the events on their queues.

This representation of the pedestrian systems is conceptualized assuming that the entities will not generate conflicts as long as they not collision on the processors. The IoTwins project is specially interested on the interactions of the Agents while they are going through the represented system making Event Based models not appropriate.

Agent Based

This approach represents the system as a space, usually 2D, and the pedestrians that walk through the space are defined as Agents. Each one of the Agents occupies its own space and interacts with the environment and with other Agents executing Actions defined by the modeller. The definition of this interactions is crucial as shown in *Scalable HPC enhanced agent based system for simulating mixed mode evacuation of large urban areas* [13].

Each agent is executed individually and each one may have individual cognition. As a result agents can behave as people would do. Keeping this in mind we can model the Agents to react and interact with other Agents generating group dynamics that changes the behaviour of the agents through feedback. The main advantage of `ABM` is the heterogeneity of the Agents, see *A Model of Human Crowd Behavior : GroupInter-Relationship and Collision Detection Analysis* [14]. Having a set of heterogeneous Agents provides us a group of different individuals that will perform similar Actions, but taking into account their individual characteristics. Hence, we can extract really detailed information of how pedestrians behave, at an individual and at a collective level. The main drawback of this type of models is that they have a very high computational cost. This question has been extensively analyzed in *A high performance framework for agent based pedestrian dynamics on GPU hardwares*[15]. To go around this high cost the use of `HPC` environment to improve the performance of the tool and achieve a reasonable execution time is common.

2.2 Justification: Why Agent Based Modelling?

Agent Based Model has been the deemed the most appropriate approach to implement the general model to represent the multiple systems of test beds 5 and 11 due to its focus on individual human movement and its ability to represent in detail heterogeneous agents. Another thing to consider `ABM` is the wide experience of the team using this modelling technique which makes it easier to conceptualize, define and implement the model and solve the obstacles that arise during the development of the model.

`ABM` satisfies the requirements of the project, including the granularity that we are looking for to represent the system. Using the main strength of ABM, the heterogeneous characterization of the population, we will obtain a much more accurate results than if we were to use any of the other approaches discussed in this section. This approach will allow to understand the behaviour of the whole population through individual decisions taken by each Agent individually.

Nevertheless, not many general representations of this size and with this many Agents have been developed successfully, due to the complexity of the implementation and conceptualization of the tool. This places the presented project at the forefront in terms of state-of-the art in pedestrian simulation and will develop tools with high potential for wider use.

2.3 `ABM` in `HPC` environments

In this section we will discuss the `HPC` environments that have been used to execute `ABM` Pedestrian Movement Models previously.

Graphics Processing Unit (`GPU`) environment

The first computational environment we will discuss is one used widely in Blockchain and graphics applications: the `GPU`. This kind of computational environment provides a very efficient way of operating matrices and a high rate of instruction processing per second. Nevertheless, you have to have specific knowledge of a hardware acceleration languages such as Compute Unified Device Architecture (`CUDA`) in order to run your application in a `GPU` environment to execute your applications. This approach was implemented on the following studies: [15] *A HIGH PERFORMANCE FRAMEWORK FOR AGENT BASED PEDESTRIAN DYNAMICS ON GPU HARDWARE* [10] *Crowd Turbulence With ABM and Verlet Integration on GPU Cards.*

We can see that this approach limits itself to the execution of multiple instances the simulation on different racks of `GPU`s and multiple experiments are executed at the same time. Making more experiments at once is a way of obtaining results at a more efficient rate, but it requires very specific infrastructure and technological know-how, which is not always available.

Multiprocessing and Multi-core execution

One simpler solution is using `OpenMP` which deploys the multi-threading capabilities of most of the processors. However, this solution is much less scalable than the GPU environment presented before. Nevertheless, multi-threading can help to reduce the execution time of a program, especially when combined with a multi-core environment.

The multiprocessor environments are a net of interconnected processors that work together to solve a complex problem. Similarly, as what happens with a `GPU` environment Multiprocessing needs the user to have knowledge of Message Passing Interface (`MPI`) and `OpenMP`, or similar frameworks of parallel computing. This comes with the associate challenge that you have to develop a correct and coherent way to execute the model and represent the space [13] *Scalable HPC Enhanced Agent Based System for Simulating Mixed Mode Evacuation of Large Urban Areas.*

2.4 Justification: Why Multiprocessing and Parallel Computing?

We will use Multiprocessing and Parallel Computing in order to reduce the computation time of our model. This will be done through the `PANDORA` framework, which uses both technologies and is ready to be executed in *MareNostrum4* (`MN4`), the Spanish National Supercomputer, the team has access to.

Chapter 3

Specification

In this chapter the specification of the prototype will be addressed starting with the conceptual model representing the components of an `ABM` model. The execution of the model and the components will be specified using various flow diagrams. After that the non functional and functional requirements of the tool will be described. Completed with the use cases of the tool. Finally, the possible obstacles that can appear during the development will be discussed.

3.1 Conceptual Model

The conceptualization of the model is represented in the `UML` diagram 3.1 and in multiple flow diagrams showing the simulation and agent logic at each step of the simulation, Figure3.2 and Figure3.3 respectively.

3.1.1 `UML` representation

The conceptual model is partially conditioned by the framework used to develop the model. PANDORA needs the model classes to be the child classes that implement the stub classes of the engine. These classes represent the Agents, the Actions, the World and the Configuration of the simulation. These classes are already related at a framework level to avoid circularity and duplicate references.

The section 4.3 Model Design section provides detailed description of the representation of all Actors and their actions in the model. Here we focus on the general architecture. As shown in Figure 3.1 the framework is capable of handling multiple Actions and, if necessary, it can handle multiple Agent implementations. The classes grouped in the IoTwins Concrete model represent the model implementation while the Simulation core site classes are the stub classes provided by the framework. The stub classes are connected with the rest of the framework to assure a correct execution regardless of the particularities of the model being executed.

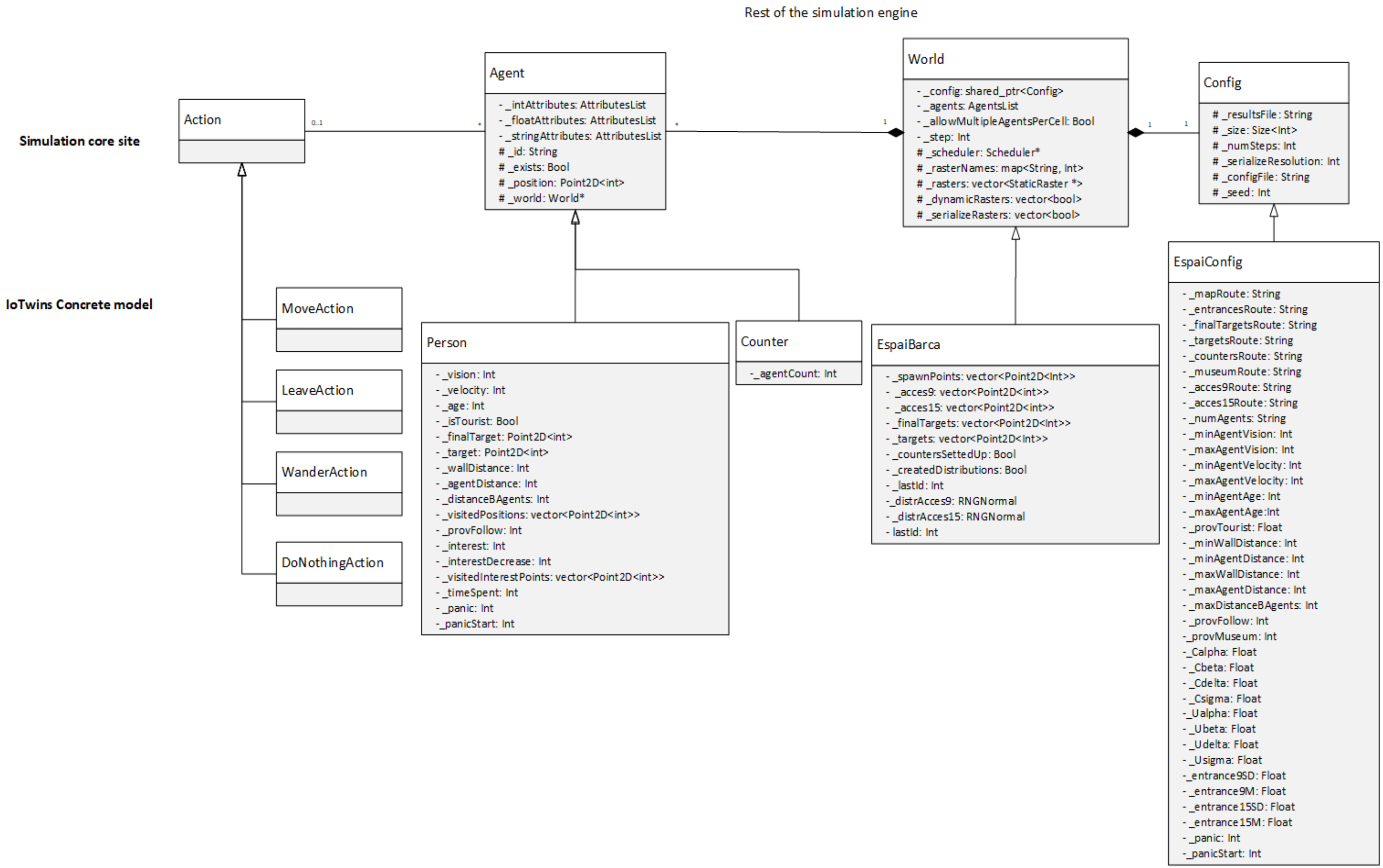


Figure 3.1: Conceptual diagram of the model(Generated with Microsoft Visio). Source: own compilation.

3.1.2 Execution flow

Figure 3.2 represents the execution of the entire model. The general structure is common for time-step simulations.

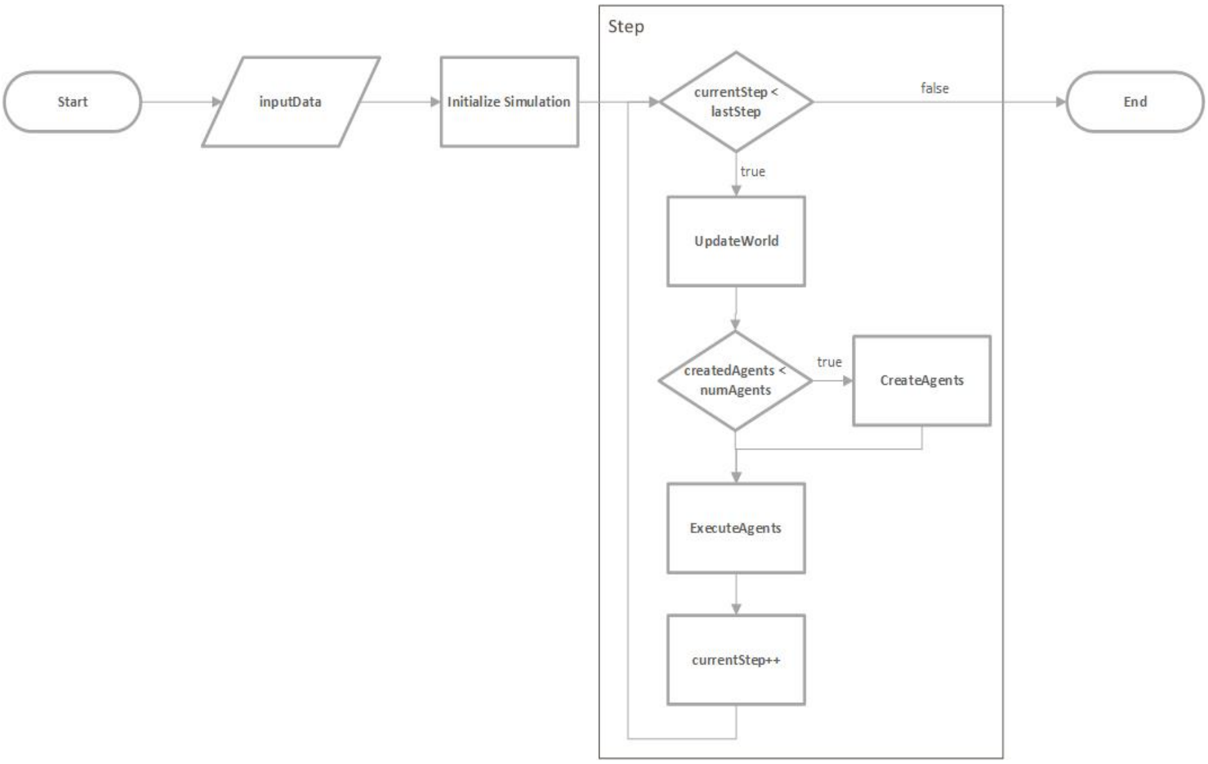


Figure 3.2: Model's execution flow (Generated with Microsoft Visio). Source: own compilation

First of all, the input parameters define the current scenario of the simulation. Once the instance of the simulation has been initialized, the simulation run starts. The execution will consist on the repetition of the step structure the number of times specified in the input.

The structure of each time step is the following: first, the world is updated from the previous state. If there are still Agents to enter the simulation, they are created at the beginning of the step. After that, the agents execute the actions depending on the current state of the simulation and the decision making strategy. Finally, the step is updated and the simulation checks whether it has to halt or not.

Figure 3.3 represents the decision making strategy designed for the Agents. At each step, the agents check if they have arrived at their destination, if that is the case they execute LeaveAction and are removed from the simulation. If not at destination, the Agent will select the best position to go to their current target and execute a MoveAction. During the movement agents can be attracted or pass through 'Interest Points' defined together with the stakeholders. If they have arrived at a predefined interest point and their interest in the location is not satisfied yet they stay close to the interest target to satisfy their requirements otherwise the continue their journey towards the destination. Finally, there is a possibility that the Agents move coherently with the whole group rather than moving to the available position that is closest to their target. Depending on each Agent this probability may change. If they have to move coherently with the group the Agent will execute the WanderAction.

Figure 3.3: Agent decision model(Generated with Microsoft Visio). Source: own compilation

To better explain each of the Actions that the Agents can execute each Action has been described by their own flow diagram.

Figure 3.4: MoveAction flow diagram(Generated with Microsoft Visio). Source: own compilation

In the MoveAction (Figure 3.4), each Agent analyzes the state of its surroundings. Then, s/he selects the best position among the available ones, while maintaining the cohesion with the other Agents near it, following the group dynamic[16]. This selection can be conditioned by various parameters, the distance to the target, the distance from other agents, obstacles or nearby interest points. Once the agent has calculated the cost of each reachable position and selected the best one, they update their position.

An alternative movement actions is the WanderAction (Figure 3.5). In this case, Agents follow the general pedestrian flow of other agents around them. After analyzing their surroundings, they select their next position according to the current position of the nearby Agents and the density of the crowd.

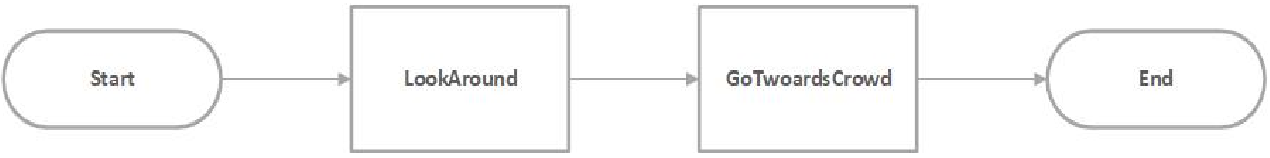


Figure 3.5: WanderAction flow diagram(Generated with Microsoft Visio). Source: own compilation

The other two possible Actions are DoNothingAction and LeaveAction. If an Agent executes a DoNothingAction(Figure 3.6) at this iteration the Agent remains static. This action helps us detect the amount of time spent still by each Agent and the locations where it happens most often (due to attraction to an Interest Point, or overcrowding, etc.). LeaveAction(Figure 3.7) is used to remove an Agent from the simulation when it detects that it has reached its final destination. The Agent updates their state to non-existent and is permanently removed from the simulation.

Figure 3.6: DoNothingAction flow diagram(Generated with Microsoft Visio). Source: own compilation

Figure 3.7: LeaveAction flow diagram(Generated with Microsoft Visio). Source: own compilation

3.2 Non-functional requirements

To assure the quality of the project the following non-functional requirements have been defined.

- **Documentation** The project must have a clear and helpful documentation, with explanations in the code that make it clear how it works and with auxiliary documents, such as this one.
- **Efficiency** The software generated must be efficient. This is important to improve the time on the result extraction, allowing to perform as detailed experiments in as short time as possible.
- **Legibility** The code must be as self explanatory as possible to help its comprehension.
- **Replicability** The model must provide tools (e.g., seeds) that enables other users to replicate the results.
- **Scalability** The model must be robust enough to support larger spaces and high number of agents while maintaining a correct representation of the system.

3.3 Functional Requirements

The functional requirements of the project are the following ones.

- The simulation must significantly improve the simulation time compared to the software currently used by `FCB`.
- The simulation must reflect the real system to a high degree of detail.
- The simulation must be correctly verified and validated against data.
- The simulation must be replicable on different systems.

3.4 Use Case

This section describes the actions performed by the user of the model. First, the use case diagram show the different use cases of the model itself and then the ones related with the post process analysis.

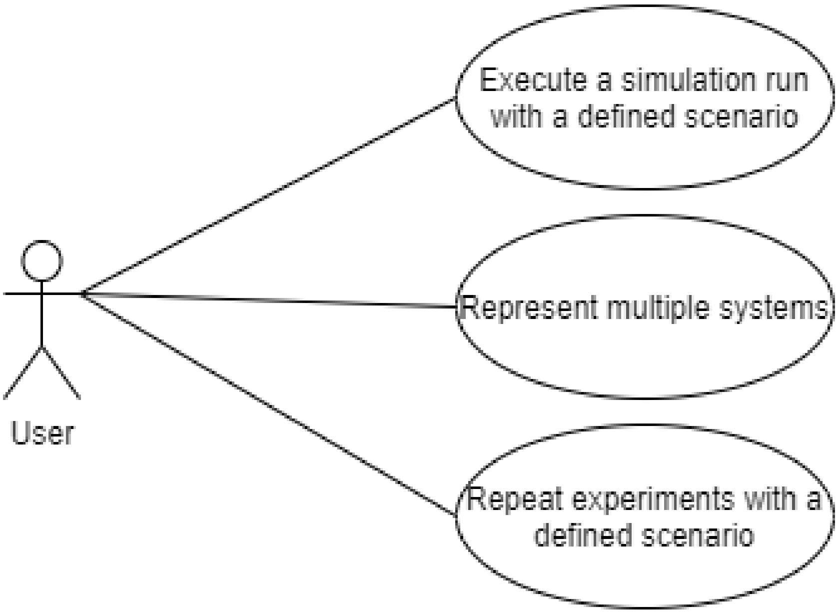


Figure 3.8: Model's use cases. Source: own compilation

Execute a simulation run with a defined scenario

- **Main actor**: User
- **Precondition**: The studied scenario is correctly defined as a model input.
- **Trigger**: The user wants to run a simulation of a described system.

Main scenario:

1. The user introduces the parameters characterizing the simulation model in a config.xml file.
2. The user runs the simulation with the config.xml file specified.
3. The model generates the output file with the simulation record at the location specified at the config.xml file.

Represent multiple real systems

- **Main actor**: User
- **Precondition**: The studied scenarios are correctly defined as a model inputs.
- **Trigger**: The user wants to execute various systems expressed as model inputs.

Main scenario:

1. The user introduces the parameters characterizing the simulation model in a config.xml file.

2. The user runs the simulation with the config.xml file specified.
3. The model generates the output file with the simulation record at the location specified at the config.xml file.
4. The user repeats the process for all scenarios.

Repeat experiments with a defined scenario

- **Main actor**: User
- **Precondition**: The studied scenario is correctly defined as a model input.
- **Trigger**: The user wants to repeat a given simulation run.

Main scenario:

1. The user introduces a specific seed in the config.xml file.
2. The user runs the simulation with the specific seed.

Once explained the use cases of the simulation model we will take a look at the output analysis scripts. Two scripts are used: one to generate a visualization animation showing the general movement of the Agents in the space; the second script gathers the data necessary to generate the `KPI`s of the simulation execution.

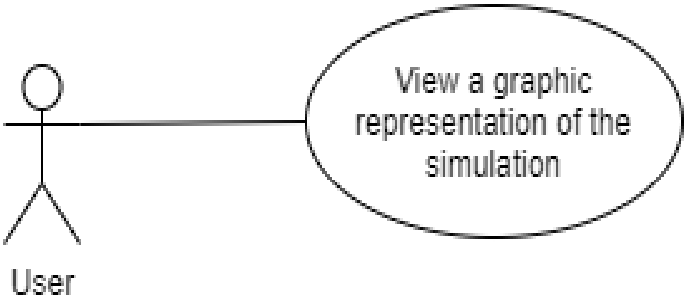


Figure 3.9: Visualization script use case.

View graphic representation of the simulation

- **Main actor**: User
- **Precondition**: A simulation run has been executed.
- **Trigger**: The user wants to view a graphical representation of the simulation run.

Main scenario:

1. The user executes the script with the path of the simulation output file.
2. The user executes the generated animation in any web browser.

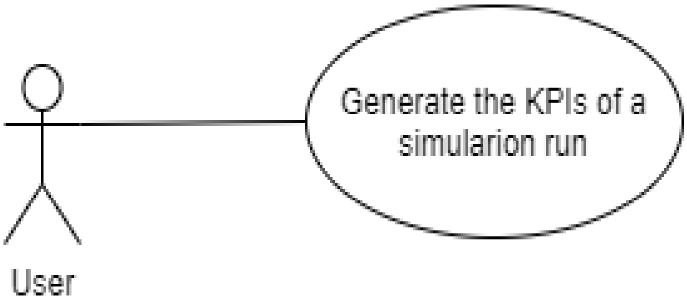


Figure 3.10: Output analysis script use case.

Generate the `KPIs` of a simulation run

- **Main actor**: User
- **Precondition**: A simulation run has been executed.
- **Trigger**: The user wants to generate the `KPIs` of the simulation run.

Main scenario:

1. The user executes the script with the path of the simulation output file.
2. The user obtains a `CSV` file with the information relevant to the generated `KPI`.
3. The user runs another script and the `MSE` value of each `KPI` is compared with the data avaliable.

3.5 Obstacles

In this section, I describe the possible obstacles that could appear during the development of the project and affect the accomplishment of the objectives.

Temporal Limitations

Like in many other project the estimation of the different tasks cost may not be accurate and lead to time discrepancies between the estimations and the real development rate.

Dependency of other partners

Some of the tasks of this Thesis require the synchronization with other teams having to accomplish their own tasks, for example the data-analysis for the data-profiling of the agents.

Implementation complexity

This project is a complex computational challenge and in the literature there are few examples of `ABM` models at this scale. This increases the risks as there are no template to follow. The initial design may turn incorrect or not optimal which would require to refactor the code in order to get the desired behaviour.

PANDORA's performance

The `PANDORA` framework currently can use multi-threading execution. However, the multi-processing scheduler is currently malfunctioning. One of the parts of the IoTwins project is to develop a new multi-processor scheduler that allows write actions. The author of this Thesis works closely with the `PANDORA` developer to assure the correct implementation of this new functionality, even though this task is not included in this Bachelor Thesis.

Chapter 4

Design

In this chapter, the design for the specification described in the previous chapter will be described. First, the environments where the model must be able to be executed will be explained and the chosen technologies justified and explained as well. After that, the design of the model will be explained using several flow diagrams to help with the description. Finally, the initial baseline scenarios will be described.

4.1 Architecture

In this section, I will describe how the model can be run in the development environment and in the `HPC` cluster.

4.1.1 Development environment

The solution implemented in PANDORA is the containerization of the framework using Docker[17]. This way the native Ubuntu16.04 can run with little effort on many systems. The container hosts the PANDORA's libraries and uses a shared volume[18] between the container and the system. The shared volume essentially allows the container to read the files from one folder and allows the data generated by the container to persist in the volume. This way the PANDORA library runs on its native system while capable of being run on any system with virtualization capacity. The files contain the implementation of the model and are related with the engine classes by inheritance, described previously (Figure 3.1).

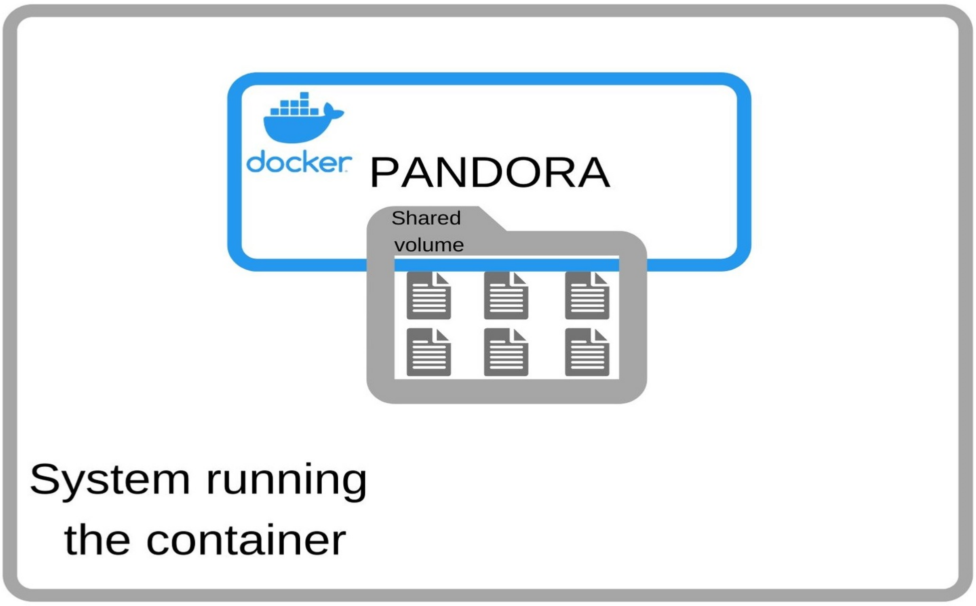


Figure 4.1: Development environment architecture diagram.

4.1.2 `HPC` Clusters

This ties with the `HPC` environment. The `BSC` clusters often run contained applications. Therefore, the Singularity app has been configured on all clusters as a way to transform the Docker containers into Singularity ones. The system allows the the contained app to run in parallel and uses all of the clusters resources.

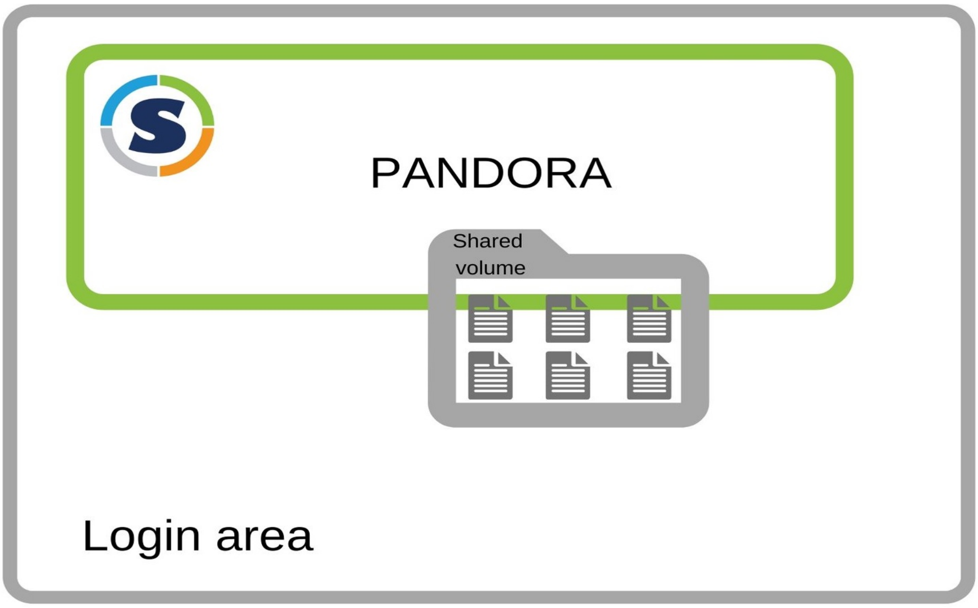


Figure 4.2: Development environment architecture diagram.

4.2 Technologies

In this section, I describe the technologies used for the development of the model introduced in the architecture section 4.1.

4.2.1 Docker

Docker is a tool to create and manage virtual containers. The containers are spaces separated from the main system that can host any type of system, from a database to an application such as Pandora. Docker is mainly used to hold the contained software in the same state as that in which the container was created so that the installation of the software is not necessary. The only thing the user needs to do is to install docker, get the desired container and run it.

4.2.2 Singularity

Singularity is another container manager. In this case it is used to convert the Docker container into a Singularity container accepted by the MareNostrum supercomputer. This layer preserves all of the Docker features. The key difference is that Singularity is a safer tool and has none of the know daemon Docker issues of running as the same time as the host. It swaps the operating system for the one in the container, and then binds the path so the file system of the host is accessible form the container.

4.2.3 Pandora

Pandora is an open source framework developed at `BSC` by Dr Xavier Rubio-Campillo. It is used to develop time-step Agent Based Models using C++ and Python3. The framework provides a time-step simulation engine and a scheduling system that runs the simulations. To implement a model using PANDORA the modeller has to develop an implementation for the virtual stub classes of the simulation engine Agent, Action, World and Configuration (Figure 3.1). The framework takes care of all of the scheduling during the execution of the simulation. There are two distinct schedulers, one `OpenMP` multi-threading approach and another multi-core one using `MPI` aimed for larger simulation experiments on an `HPC` cluster. PANDORA's scheduling is based on time-step simulations, where each time steps represents an amount of time in the real system. PANDORA is also prepared to work with spatial data inputs using the GDAL library [19] to input GIS files[20]. This spatial data is represented in PANDORA as raster maps, with a variable height index allowing superposition. An HDF5[21] file is provided as output. This output can be analyzed with any data analysis software, such as the Pandas Python library.

4.3 Model Design

In this section, I describe the design of the actors and actions previously specified. All of the components of the `ABM` model will be explained individually. This includes all

Agent types, the definition of space and the Actions that the Agents execute during the simulation.

4.3.1 Agents

Two different kinds of Agents where identified for the initial implementation in the simulation. The pedestrians and the devices that count them. The pedestrians can be divided into two categories, tourists and locals. They have very similar attributes but their rules of behaviour are significantly different. Some attribute values are randomly generated while others are models using the data input of the simulation.

- **_vision**: Distance in meters defining the vision radius at which the agent pays attention to the environment and the other Agents. Humans can obviously see as far as a kilometer or more thus the 'vision' attribute does not describe the maximum visibility radius but rather the 'attention' radius.
- **_velocity**: Maximum distance that an Agent can travel in one time step.
- **_age**: Age in years of an agent.
- **_isTourist**: True if the Agent is a Tourist, false otherwise.
- **_finalTarget**: Position to which the Agent is heading.
- **_target**: Location the Agent aim to reach. _target may be a Point of interest or a partial target if the route to the _finalTarget is divided into stages. It is always different from **_finalTarget**.
- **_wallDistance**: Minimum separation in meters between the Agent and vertical obstacles such as walls.
- **_agentDistance**: Minimum separation in meters between the Agent and another Agent if they walk in a group.
- **_distanceBAgents**: Maximum distance in meters between two Agents. This is used on the cohesive movement to avoid isolated agents when they walk in a group.
- **_provFollow**: Probability of an Agent to follow the crowd.
- **_interest**: Interest of an Agent in their current **_target**.
- **_interestDecrease**: Maximum interest decrease per step of the Agent.
- **_visitedPositions**: Vector of visited positions by the Agent.
- **_visitedInterestPoints**: Vector of visited interest points by the Agent.
- **_timeSpent**: Time spent in the simulation by the Agent.
- **_panic**: 1 if the simulation will have an emergency situation. 0 otherwise.
- **_panicStart**: Step when the evacuation of the system starts.

Local

Agents classified as 'Local' represent the pedestrians who are repeated visitors to the Camp Nou Stadium and its precinct. These Agents have a better level of spatial orientation and are not easily distracted by the interest points. This type of Agents has a higher average speed and they usually go alone.

Tourist

Agents classified as 'Tourist' present opposite characteristics to 'Local' Agents. They usually go in groups, which makes them slower on average. They stop at almost all of the interest points. Since they are not familiar with the spatial distribution of the stadium and the Espai Barca precinct thus they have higher probability of following the crowd.

Counter

The Counter Agents represent the devices in the system that count the Pedestrians at certain position. These Agents will be used to generate some of the `KPIs` that will be used to validate the model.

4.3.2 Space

The space where the model is located will be defined by the topology we use as input. The minimum information required to define the model's space include: the building boundaries, location of the access points, location of final targets and the distribution of interest points. The spatial later is fed into the model as a series of monochromatic bitmaps, which ensures a fast loading time. The generation of accurate bitmaps can take significant time, but it is done only once and it facilitates the handling of spatial information in the model and lightens the reading of the input. More bitmaps, if needed, can be added easily to provide more information as another layer of the topology. Each one of the bitmaps cells represents a space of 1m^2 in the real world.

4.3.3 Actions

The Actions is where the majority of the simulation logic regarding the Agents is placed. The following Agents' Actions constitute the rules of behaviour: *the Move Action* and *the Wander Action.* Leave Action and doNothingAction have been discussed previously in section 3.1.2.

Move Action

The Move Action, the most important of the Actions of any Pedestrian Modelling study. Essentially this Action is solving a path-finding problem for a general topology, with a few restrictions attached to it. It has to be as efficient as possible, it must be a realistic way of movement, coherent with the general movement of all the agents, meaning that the obstacles are not static (one Agent cannot be at the same position as any other one).

A*
The most common path-finding algorithm is the A*[22]. A* is the most efficient algorithm to generate the path to the target. It starts with a width-search and each step of the algorithm updates the cost of the path and looks for a path with the least cost. Once reached the target reports the optimal path. Nevertheless, A* breaches one of the restrictions, it does not consider the possibility that the path can be blocked by the movement of the other Agents. Another thing to consider is that A* completes the search without moving from the origin, breaking the vision restriction. All of the other width-search algorithms such as Dijkstra[23] or the Greedy[24] algorithms face the same issues.

Local search
This type of algorithms can utilize the vision restriction and locally select the best position to move to in the current time step. We have to view the topology as a cost function and the Agent must navigate it while minimising cost. However, this kind of search can lead to infinite loops if the Agents reach a local minimum. This situation can be fixed with a recording of the positions visited, making it less space and time efficient, due to the necessity of backtracking if a local minimum is reached.

Coherent walking
Widely used by social studies using `ABM` and related to the Netlogo community. A heading direction is assigned to the Agent facing the target and the agent moves in the direction it is facing. The obstacles in this case are avoided by slightly changing the heading and moving towards a valid position. The position selection algorithm can be conditioned by the elements of the restriction. *A priori* this could seem the best, and most realistic option. However, it must be kept in mind that most of the models used in the social models are significantly smaller than the one developed in this Thesis.

Wander Action

The Wander Action makes the agent move with the general flow of the whole system. It is common to use the approach known as Flocking (or Boids) model[25]. It consists of agents who are using their local state and that of nearby agents to evaluate where the flow is heading to and what is its density. Since every Agent moves according to the result of their own local information all of the Agents move coherently with each other.

4.3.4 Input configuration

Finally, the input parameters of the model must be defined. PANDORA's input is an `XML` file with the parameter values and the relative path to the maps used by the model. There are also some parameters that are specified by default. The model uses the following input file. At the moment, a user interface to generate this input files and help with PANDORA's experimentation is being developed as another bachelor thesis:

```
<?xml version="1.0" ?>
<config>
    <output resultsFile="./scripts/experimentation/results/exp1.h5"
    logsDir="./logs"/>

    <numSteps value="3600" serializeResolution="1"/>

    <seed value="4"/>

    <size width="970" height="970"/>

    <inputData numAgents="1000" numCounters="3"/>

    <topology map="resources/bitmaps/buildings2.bmp"
    entrances="resources/bitmaps/noPartit/entrances.bmp"
    finalTargets="resources/bitmaps/noPartit/entrances.bmp"
    targets="resources/bitmaps/noPartit/targets5.bmp"
    counters="resources/bitmaps/noPartit/counters.bmp"
    museum="resources/bitmaps/noPartit/museum.bmp"
    acces9="resources/bitmaps/noPartit/acces9.bmp"
    acces15="resources/bitmaps/noPartit/acces15.bmp"/>

    <agentData minVision="5" maxVision="10" minVelocity="1"
    maxVelocity="1" minAge="0" maxAge="80" porvTourist="0.8"
    minWallDistance="0" maxWallDistance="1" minAgentDistance="1"
    maxAgentDistance="2" maxDistanceBAgents="10" provFollow="50"
    provMuseum="31"/>

    <coefficients Calpha="1" Cbeta="1" Cdelta="0.8" Csigma="0.01"
    Ualpha="1" Ubeta="1" Udelta="0.01" Usigma="0.01"/>

    <statistics entrance9SD="88.17" entrance9M="175"
    entrance15SD="18.406" entrance15M="50.222"/>

    <panic panicOn="0" panicStart="1000"/>
</config>
```

The default parameters needed by the framework are the following ones:

- **output resultsFile**: File where the recording of the simulation will be stored.
- **output logsDir**: File where the logging system will report if used.
- **numSteps value**: Number of steps of the simulation. One step represents one second on the real system.
- **numSteps serializeResolution**: How many iterations are executed per time-step.

- **seed value**: Seed that is used for the simulation run. If the value is -1 the seed is randomly picked.
- **size width**: Width of the word used.
- **size height**: Height of the word used.
- **inputData numAgents**: Total number of Person Agents.

The specific parameters for the model developed in the Thesis are the following ones:

- **inputData numAgents**: Number of Counters.
- **topology map**: Bitmap of obstacles.
- **topology entrances**: Bitmap of entrances to the modelled space. These points generate new agents.
- **topology finalTargets**: Bitmap of valid final targets of the agents.
- **topology targets**: Bitmap of interest points of the system.
- **topology counters**: Bitmap of the position of agent counters.
- **topology museum**: Bitmap of the museum entrances.
- **topology acces9**: Bitmap with the location of gate 9.
- **topology acces15**: Bitmap with the location of gate 15.
- **agentData minVision**: Minimum Agent vision.
- **agentData maxVision**: Maximum Agent vision.
- **agentData minVelocity**: Minimum Agent velocity.
- **agentData maxVelocity**: Maximum Agent velocity.
- **agentData minAge**: Minimum Agent age.
- **agentData maxAge**: Maximum Agent age.
- **agentData porvTourist**: Odds of the Agent being a tourist.
- **agentData minWallDistance**: Minimum value of the minimum distance that an Agent can be form an obstacle.
- **agentData minAgentDistance**: Minimum value of the minimum distance that an Agent form another Agent.
- **agentData maxAgentDistance**: Minimum value of the minimum distance that an Agent form another Agent.
- **agentData maxDistanceBAgents**: Maximum distance from other Agents.

- **agentData provFollow**: Base odds of an Agent to execute a Wander Action in a step.
- **agentData provMuseum**: Odds of an Agent to select the museum as its finalTarget.
- **coefficients Calpha**: Alpha coefficient of the cost function.
- **coefficients Cbeta**: Beta coefficient of the cost function.
- **coefficients Cdelta**: Delta coefficient of the cost function.
- **coefficients Csigma**: Sigma coefficient of the cost function.
- **coefficients Ualpha**: Alpha coefficient of the utility function.
- **coefficients Ubeta**: Beta coefficient of the utility function.
- **coefficients Udelta**: Delta coefficient of the utility function.
- **coefficients Usigma**: Sigma coefficient of the utility function.
- **statistics entrance9SD**: Standard deviation of the function representing the entrances of Agents through access gate 9.
- **statistics entrance9M**: Mean of the function representing the entrances of Agents through access gate 9.
- **statistics entrance15SD**: Standard deviation of the function representing the entrances of Agents through access gate 15.
- **statistics entrance15M**: Mean of the function representing the entrances of Agents through access gate 15.
- **panic panicOn**: 1 if the simulation will have an evacuation of the system. 0 otherwise.
- **panic panicStart**: Step on which the Agents will start the evacuation of the system.

4.4 Scenarios

Since the model must be able to represent different scenarios in multiple situations and in this case different typologies. The scenarios must be defined to be able to transform the necessary raw data for the model. Through the inclusion, implementation and experimentation of new scenarios the model will be calibrated and, at a later stage, extended. All of the knowledge gathered during the execution of all of the scenarios will help the team to make the model robust enough to handle a vast number of pedestrian movement scenarios. In this section, I describe the scenarios developed in the IoTwins project that will be implemented throughout the duration of the project.

4.4.1 Scenario 1: Match day

This scenario represents a day during which there is a football match taking place at the *Camp Nou* Stadium. This happens on average once every two weeks and attracts up to 100 000 visitors to the venue. As such, it is a uniquely challenging situation requiring high level deployment of staff and resources to ensure smooth flows of pedestrians and to guarantee their safety.

This scenario can be divided into two distinct phases, the access and the leaving of the stadium. Each one of the phases works in a different way. To access the stadium an Agent has to go through several controls, at the access door and at the gates of the stadium. We need to take into account that the Agents do not have to access the precinct through the door that is closest to the stadium gate related to their seat as all visitors with a valid ticket can enter through any of the control points. Also the Agents with objects that can not be introduced to the stadium (motorbike helmets, backpacks, etc.) have to first go to the storage to leave the forbidden objects before heading over to the stadium. This structure leads to long queues that form at the gates inside the prescient. Most of the visitors have to cross these dynamic obstacles to reach their seat.

The after-match phase is less complex, since the Agents only have to leave the stadium and the precinct sometimes going back to the storage to pick-up what they have left before accessing the stadium. However, compared to the access phase, these actions are more condensed in time since all visitors depart at the same time.

4.4.2 Scenario 2: No match day

This scenario represents a typical day at the *Espai Barça*. It is open to visitors and pedestrians can walk inside the precinct to cross it from different access doors that remain open during the day.

The precinct is open to the visitors of the museum, the shop and the various establishments (bars, restaurants, etc), as well as the workers. The museum in particular has been defined as one on the scenarios to study given that it attracts up to 15 000 visitors a day. The access to this museum is on the commercial alley and will be represented with this scenario.

This scenario has been defined because it was evaluated that the construction renovation works at *Espai Barça* will have a bigger impact on pedestrian movement during non-match days. Changing the way pedestrians access and cross the precinct will therefore be necessary.

4.4.3 Scenario 3: Museum

The most distinctive scenario is the Museum scenario. The `FCB` museum is the most visited museum on Barcelona(1,785,903 on 2015[26]). The museum is part of the *Camp Nou Experience*, which includes a tour, guided or free, in the museum and around the stadium.

Parts normally non-accessible of the stadium are visited on the tour, such as the press room or the locker room for example. The tour is guided, but in some areas there is some freedom of movement.

4.4.4 Scenario 4: Johan Cruiff Stadium

Finally, the last scenario will be the Johan Cruiff stadium on located on the street of *Sant Joan d'Espí*. It has been chosen as a venue for simulation as the results will be more applicable to a wider range of case studies. Most sport venues are of similar capacity to this one, with only a few being as large as *Camp Nou*. In this case the scenario is similar to scenario 1, but with less capacity and a different topology and access controls. In this case the main interest of this scenario will be the emergency plan and how to evacuate the precinct. With all other scenarios the emergency plan can be modeled as well, but the project specifies that emergency planning will first be developed and tested on this venue before replicating it in the much larger *Camp Nou*.

Chapter 5

Implementation, Experimentation and Validation

In this section, the implementation of the model's prototype will be described. The exhaustive description will cover the Agents, the World and all of the Actions implementations. The model is developed as a general tool but the particular implementation uses the scenario 2: No match day, as a guideline. This was the selected scenario to start the experimentation with the model following the road-map of the whole project. The other scenarios are planned to be implemented and tested between November 2020 and September 2021. Afterwards, the experimentation frame will be developed taking into account the lack of data due to being on the early stages of the project. At the current moment the team only has access to the counting of people entering the facility and the data of one counter inside of the facility. Finally, the validation strategies that will be used to validate the model at later stages will be discussed. A exhaustive validation will be done when the data gathering structure is running.

5.1 Agent: Person

Since the Agents on this simulation share the same attributes and only differ in their values, they are represented as instances of the Person class. On their creation its taken into account if the Agent is a Tourist or not and the attributes are set accordingly. Before the selection of the Actions that will be executed each Agent analyzes their surrounding area and depending on their state decide which actions to take. This is done using the **updateKnowledge** method with runs in parallel.

In this method the Agents update their state. Then detect the interest points within their vision radius and decide if they want to go there or not depending if they have already seen the interest point or not. It is in this function that the Agents check if they have satisfied the interest on their current target. If true, they no longer are interested in it and proceed to go towards their final target. Along the way the Agent can visit multiple interest points. Once the interest is satisfied in one of them this point is stored in **_visitedInterestPoints**. The Agent will ignore it during the rest of the simulation to avoid Agents getting stuck or looping around one interest point.

The action selection is done by implementing the **selectActions** method provided by the framework. In this method the Actions to execute in the current step must be pushed to the **_actions** attribute of the stub class Action. The scheduler will handle in which order the Agents execute actions, but the sequence of Actions of each Agent is the one specified by the **_actions** list.

In this case, following the flow in Figure 5.1, each Agent will only perform one of the possible four Actions following this decision model.

Figure 5.1: Agent decision model.

Finally, the attributes that need to be registered on the output file must be registered and serialized with the **registerAttributes** and the **serialize** methods from the virtual functions of the stub class. The default information stored: the id of the Agent and their position are recorded per Agent and per step.

5.2 Agents Counter

The Counter has a much simpler implementation. Using the **updateKnowledge** method the Agents update their **_agentCount** attribute that represents the number of pedestrians that have gone trough the counter. After that, this attribute is serialized and registered.

5.3 World

PANDORA uses the World stub class to define the typologies and the type of environment the simulation take place in. In the World class the simulation structure is defined. The World is a composition of all of the Agents in the simulation and the input configuration used. When the World instance is created it uses the **createRasters** method to define which raster maps the simulation will use. These rasters are generated with the same size as the World, specified in the configuration file. The rasters are organized into layers, so the modeller can decide the order, although there can only be one raster per layer. Then the data is loaded into the rasters form the files specified on the input using PADORA's GDAL module.

Finally, the **step** method is used to define the structure of a time-step (Figure 3.2). The step is going to be executed as many times as specified on the input, one step representing one second on the real system. At every step new Agents will be created, if necessary, using the **createAgents** method according to the data input retrieved from data files passed by the `FCB` team.

Every attribute of the Agents is susceptible to change in the experimentation stage. The distributions means and limit values are stored on the Configuration class, this class is friends class with the World class. A friend class can access the private attributes of the class it is friends with. PANDORA recommends the use of this structure to avoid replicating the input parameters on other classes.

Once Agents have been created they execute their actions. The implementation of the World class calls the **executeAgents** of the simulation scheduler. Once all of the Agents have performed all their Actions with the scheduler, the method **removeAgents** eliminates these Agents which have arrived at their destination.

5.4 Actions

The implementation of the actions is done by implementing a subclass of the Action stub class and implementing the **execute(Engine::Agent&)** method. This method works as a main function of the Action. This function receives a pointer of the Agent performing the Action. Usually information gathered by the agent is used during the Action execution.

5.4.1 Move Action

The most important and the most complex one, is the MoveAction. This Action models the way pedestrians move. The Agents not only have to reach their targets, they have to do it in a realistic way and within a reasonable time window. Taking that into account the selection of the algorithm is crucial for the performance of the model.

In this case, taking into account that **coherent walking** and the **local search**, seen in section 4.3.3, seemed as the two viable implementation strategies the team decided to implement both and compare their performance.

Coherent Walking

This algorithm sets up a heading direction of the agents, using degrees or cardinal points. When created, the Agents set up their heading direction in the **updateKnowledge** phase towards the interest point, if they are interested on one, and if not they set it to their final Target. Once the Agents have set up the heading they are ready to execute the MoveAction.

The first thing they try to do is move to the next position aligned with the heading, taking into account the **_velocity** of the Agent. If possible, the Agents try to move the maximum distance at each step. If this next position is not a valid one, e.g., it is an obstacle or it is occupied by another Agent, then the collision avoidance algorithm activates.

This algorithm determines where to turn the heading to, right or left with a maximum turn, the selected turn will be the one closer to the final target direction. If both are the same the turn is selected randomly. Once the heading has been corrected the Agent tries to move to the new position in front of it. If it is not a valid position still the Agent stays put for this step.

Although this felt like a promising start, this algorithm does not behave well in systems this big and with topologies with round obstacles. There is a point when trying to go around one of such obstacles that the algorithm detects that it is better to go around the obstacle through the other direction that the Agent has first taken. It leads to agents moving back and forth from one local minimum to another. Even with the memory feature included to avoid the Agents revisiting positions the algorithm lead to underwhelming results. This movement algorithm works well with square topologies and delimited spaces.

Local search

Moving on to the **local search** algorithm, the idea behind it is to perform an **A*** search but at a local scope. Only the positions that can be reached by the Agent during the current step are taken into account as valid, saving calculating all of the path until the final target like the original **A*** would do. This way we save computation time of calculating a path that will likely not be correct and represent real-world human decision making more accurately.

After selecting these positions a cost value is assigned to each one of them. The cost function takes into account several variables and the position with the least cost will be selected as the next position. In this case the distance to the target, the distance to an obstacle, the distance to other Agents and if the position is too far form other Agents. The cost function has been defined this way to maintain the group coherency when the MoveAction is executed. The cost function is defined as:

$$C(i,j) = \alpha_c D_t + \beta_c D_a + \delta_c D_w + \sigma_C F_a$$

The coefficients (α_c, β_c, δ_c and σ_c) represent the relevance of each one of this factors. These coefficients must be calibrated through sensitivity analysis and parameter sweep for each scenario to obtain their optimal values for a general system.

Using the cost function causes two interesting features to emerge in the model:

1. The space is converted into a cost map.
2. The heading feature of the **coherent walking** approach is maintained on this approach.

Representing the system as a cost function helps with the general model. Many more factors can be taken into account in the cost function depending on the system and more information layers can be easily worked into the model. For instance, if the system we are working with has slopes or stairs this can be introduced into the model as a heat-map with increasing or decreasing cost on different sections of the terrain. This added cost can be added to the cost function easily.

This representation also allow for a dynamic obstacle avoidance, since the cost value is calculated from each Agent's point of view. At each step the Agents detect if other Agents are stopped in a queue or taking a picture and thus creating a new obstacle.

As for number two, if an Agent is walking towards, for instance (-x,y). The cost of keep walking to -x will be less than walking to (+x,y), and if the case arrives that has to move to (-x,-y) since the cost map will be updated each step the change of "heading" will change slowly. This implies that the heading feature emerges from the cost map approach. Emergence is one of the key features of Agent Based Modelling and it is interesting to see it occurring in the model. In this specific case the heading feature has not been implemented in the model, but it emerges from Agents behaviour.

There are issues to deal with using this approach as well. Since the representation of a cost map is a mathematical way of viewing the system exists the possibility that the Agents can get stuck at a local minimum. This is one of the main weaknesses of the local search approach. To deal with this the memory feature has been used, recording the last visited positions of the Agent, allowing to backtrack if necessary and there is no better option.

The risk exists that the definition of the cost function cannot represent all of the different systems and scenarios. Even though this approach is can be easily modified it has to be validated with all of the scenarios taken into account on the project. As said before this will be done in spring 2021.

5.4.2 Wander Action

Most of the time when walking thought the street we do not constantly mentally update the current position or think in what direction and how fast we are going. In many cases one just simply follow the crowd and its pace. In both cases the people just walk in the same direction to where most people go if it is roughly consistent with their target destination. These two situations are simplified and modeled as the Wander Action.

Each one of the Agents updates their state and checks at which of the accessible positions there are more people nearby. Not-valid positions are not taken into account. This behaviour causes the pedestrian flow to remain uniform. Since the heading feature emerges from the MoveActions the WanderAction maintains this feature as well. If not lost the Agents wander towards the crowd, but maintain the minimum distance. These features are calculated within a utility function and the position with a higher score will be the one selected. The utility of each position is obtained as:

$$U(i,j) = \sum_{i \in N} \sum_{j \in N} \alpha_u A_{i_j} + \beta_u D_a + \delta_u D_w + \sigma_u F_a$$

As well as with the MoveAction the coefficients must be calibrated trough sensitivity analysis with the implementation of future scenarios.

Finally the Agent gatherings emerge from this Action, comparable to situations in which people go to see where other people are, causing spontaneous aggregations. A dynamic obstacle emerges from this implementation that the MoveAction needs to resolve.

5.4.3 Leave Action

To implement the LeaveAction and to maintain the coherence with the framework agent control the **remove** method. This method will set the **_exist** attribute of the calling Agent to false. When the scheduler of the simulation executes the **removeAgents** method the Agent will be removed form the simulation.

5.4.4 Do Nothing Action

Executing this Action means that the Agents maintain their position during the current time-step. In future scenarios this will be used to report how much time the Agents remain static. This will be done to record how much time the Agents are waiting in a queue, being stuck in an aggregation or looking at an interest point.

5.5 Scenario 2: No match day

To define this scenario without the data provided by the data gathering team, various papers and open data studies have been consulted by the BSC team. The general characteristics are described on the *Walk this way*[27]. Chandra and Bharti give a broader summary of walking speeds[28], and in *Agent-based simulation of pedestrian behaviour in closed spaces: a museum case study*[29] the velocity changes between situations are described. This data is going to be used to start calibrating the model while case study specific data is being collected by the Camp Nou team. AIn this iterration of the model the only available real data used in the project are the parameters of the arrival, length of stay and exit distributions of Agents in the system.

Using this data the input file characterizing the Agent attributes is described in the following table:

Parametery	**Value**
minVision	5
maxVision	7
minVelodity	1
maxVelocity	2
minAge	8
maxAge	80
provTourist	0.5
minWallDistance	0
minAgentDistance	2
maxAgentDistance	2
maxDistanceBAgents	10
provFollow	50
provMuseum	31
entrance9SD	88.17
entrance9M	175
entrance15SD	18.406
entrance15M	50.222

Table 5.1: Input values defining scenario 2. Source: own compilation

The definition of the scenarios for the general model is heavily determined by the spatial layer input. Here, the topology has been obtained through the transformation and modification of the Open Street Maps representation of the *Espai Barça* precinct. This representation gave the details of the buildings (obstacles) and the location of entrances to the precinct, where the Agents are generated. This transformation has been realized using QGIS[20], isolating the layers where this elements are represented and using Gimp to transform the 256 color bitmaps into mono color bitmaps to decrease the file size.

Once this two bitmaps have been defined, the interest points(on blue) have been discussed with the members of the team and manually incorporated as another bitmap. In this case the elements identified as interest points where maps of the precinct, orientation signals, pictures of the players, ticket selling windows and points of view of the stadium.

Finally, the final targets in this scenario are both entrances(on green), the museum and the store gates. Superposing the four bitmaps we obtain the following topology:

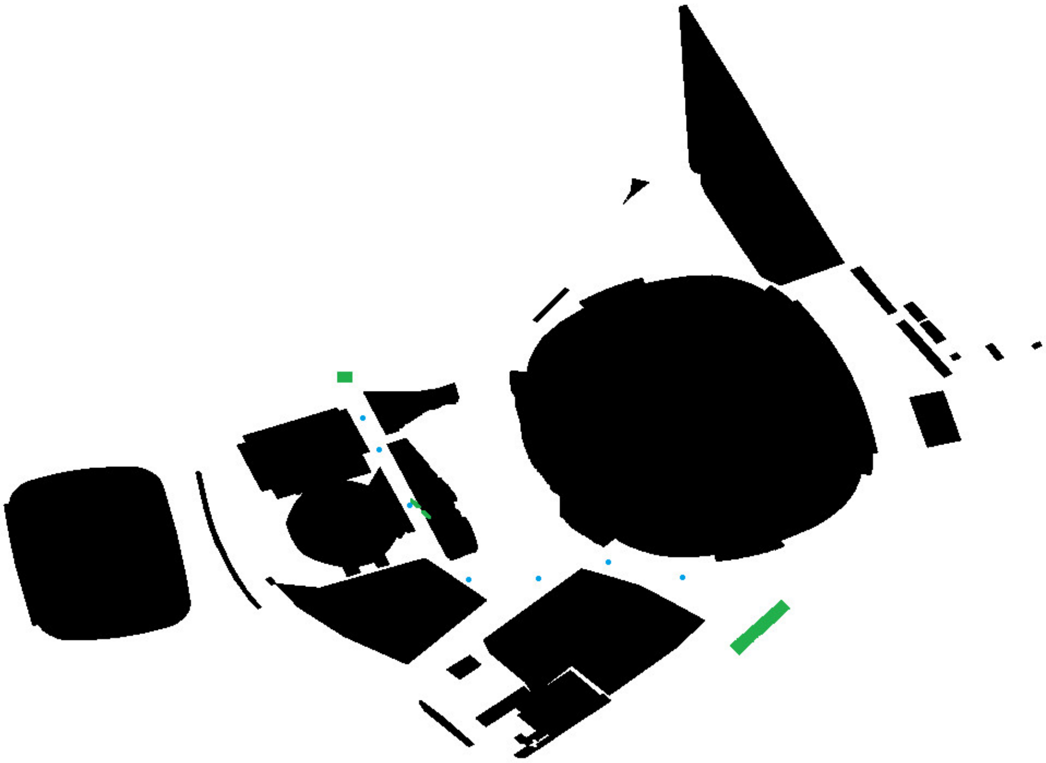


Figure 5.2: Bitmaps defining scenario 2: No match day. Source: Open Street Maps

5.6 Experimentation

The experimentation using the prototype developed is going to be based on the scenario and with the data described above, to start the calibration of the model as soon as possible. The experiments are going to be used to verify the model, to assure that the model represents the real system in a correct way on multiple situations. To conduct the verification of each of the experiments Key Performance Indicators(**KPI**)s have been defined and will be compared with the available data. When the data infrastructure or more raw data is available to the team the experiments are going to be performed with all of the input being real data. The access to more data than will allow to calibrate the model better and thus characterize better the system.

The experiments have been designed taking into account the limited data available to the team at the moment. Real data is key to `ABM` to establish a baseline to compare with on each simulation run. The data available are the number of entrances through access 9 and 15, the number of accesses to the museum and the average time spent on the facility. The experiments are going to be run with `MN4`.

5.6.1 Experiment design

The experimentation has been defined in four sets of experiments with different a objective each of them.

Set1: Calibration

The first set of experiments aims to start the calibration of the model. Due to the amount of parameters that the model has and the limited time to perform experiments on this Bachelor Thesis the team decided to calibrate the coefficients for the Agents' movement algorithm using sensitivity analysis.

The calibration is performed by executing an experiment for each one of the combinations of the different coefficients. Taking into account that we have 4 different coefficients with values compressed between 0 and 1 this leaves infinite combinations. The team decided to narrow down the possibilities to four possibilities, 0.1, 0.3, 0.6, and 0.9. Also, during the development of the model it has been detected that the α parameter must have a value close to 1 for the model to execute correctly.

In the end, set1 is composed of 64 short experiments covering all of the combinations of β, δ and σ. This sensitivity analysis will serve as an initial approximation to the optimal values of the coefficients to better represent the pedestrian movement of the model and thus achieve better results on experiment sets 2, 3 and 4.

Set2: Verification

The verification set has as its objective the verification of the model and to establish a baseline for future experiment sets. The aim of model verification is to establish whether the model as a whole behaves as intended. Using the calibrated coefficients, resulting from experiments set1, we run seven simulations with identical input configuration, but with different seeds.

Each one of the runs produces all of the `KPIs` associated with the model and compared with the available data. This way we can evaluate the degree to which stochasticity of the simulation affects the development of a simulation run and its variance.

Once the baseline has been established sets 3 and 4 were executed. These experiment sets represent scenarios of interest to the stakeholders: an emergency evacuation and topography modification due to renovation construction work. For these situation there is no available data because they are yet to occur. Thus these simulations give us some insight into how pedestrians within the system may behave under extraordinary circumstances.

Set3: Emergency

Experiment set 3 focuses on how the pedestrians may evacuate from the facility. It uses the same seeds as in set 2 experiments but is interrupted with an emergency situation. The simulation starts the usual way, but at step 7300 the evacuation starts and all Agents in the system head to the exit closest to their current location. The evacuation takes 400 steps(6,6 minutes) - time more than sufficient to evacuate the venue.

This set of experiments aims to detect bottlenecks across the facility during the evacuation and to give an approximation of the necessary time to evacuate specific areas of the venue. Scatter representations of the Agents on the system are used to visually detect said bottlenecks and the runs will also serve to estimate the minimum time of evacuation of the facility.

Set4: Obstacles

Finally, on set4 we want to determine how the changes to the facility's topology may affect pedestrian flows. Using the same configurations as in set2, but with different building maps (that include obstacles that the Agents need to get past). These simulations analyze the changes to the behaviour of the Agents for different spatial constrains. The same configurations and the seed are the same as in set1 experiments to isolate the impact of the new topology on pedestrian flows.

This set of experiments will also help us detect if the movement algorithm is able to find a way around the new obstacles presented by the new topologies.

5.6.2 Key Performance Indicators (`KPIs`)

The Key Performance Indicators are generated from the output data of the simulation. This data needs to be processed using Python scripts that generate the values of the `KPIs` then these are compared with the real data to see how the changes on the simulation input have affected the system. The `KPIs` defined at the moment, taking into account the limited data available are the following ones:

- Number of Agents leaving through access 9.
- Number of Agents leaving through access 15.
- Average time within the venue.
- Average time spent by an Agent to go from the Counter position to access 15.

5.6.3 Result analysis scripts

PANDORA does not provide a way to analyze ths simulation output, and the tool for this intended purpose, Cassandra, has not received maintenance since 2012 so that most of its components are deprecated. Another Bachelor Thesis is developing a prototype of user interface with to replace Cassandra.

In the meantime the team decided that the results would be analyzed using a series of Python scripts to be able to easily batch analyze the execution traces of the model, and this way start to calibrate the model. A script has been developed to generate an animation that represents the movement of the Agents during the simulation. Another one extracts the information relevant to the `KPIs`. This data is stored on `CSV` files for further statistical analysis.

5.6.4 Results

Here the results obtained and the knowledge gathered from each one of the experiment sets will be discussed.

Set1

The first thing noticed on the set of experiments was that the distribution of people entrances to the system did not represent accurately the real data. The team decided to use the raw data available, the number of entrances every 15 minutes, to create the exact number of Agents every 900 steps(15 minutes). This will increase the error of the data generated by the simulation, but the representation will be better than the one gotten using the erroneous distributions

To perform the sensitivity analysis necessary for the initial calibration of the model the team decided to calculate mean squared error(`MSE`) of each one of the `KPI`s comparing the values generated by the simulation runs. The selected combination of coefficients was the one with the combination of lower `MSE` for the four `KPI`s.

These combination of coefficients is $\beta = 0.9$ $\delta = 0.3$ $\sigma = 0.6$. The surface plots show the evolution of the error, fixing one of the variables to the found value and modifying the value of the other two (Figure5.3, Figure5.4 and Figure5.5):

Error with $\beta = 0.9$:

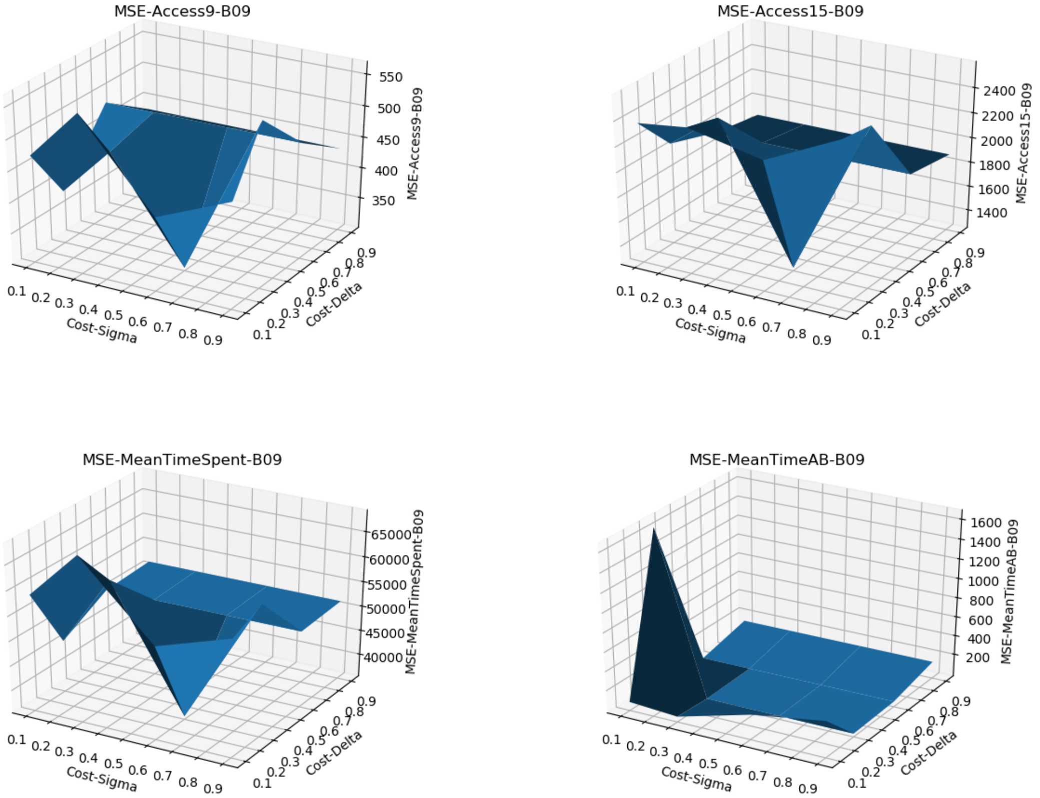


Figure 5.3: `MSE` evolution for $\beta = 0.9$(Generated with matplotlib). Source: own compilation

On the graphics above, on Figure 5.3, we can see that the local minimums on the functions are on the (0.6,0.3) or really close by. On the functions that the (0.6,0.3) is not the minimum, the value on the other functions is a lot higher than the value of (0.6,0.3). These tendency repeats on the below graphs, changing the point for the variable coefficients on each((0.9,0.6) in the case of $\delta = 0.3$ and (0.9,0.3) in the case of $\sigma = 0.6$, see Figure 5.4 and Figure 5.5). In the future further sensitivity analysis will be required, but these results will be used as a baseline and worked in the next sets configuration.

Error with $\delta = 0.3$:

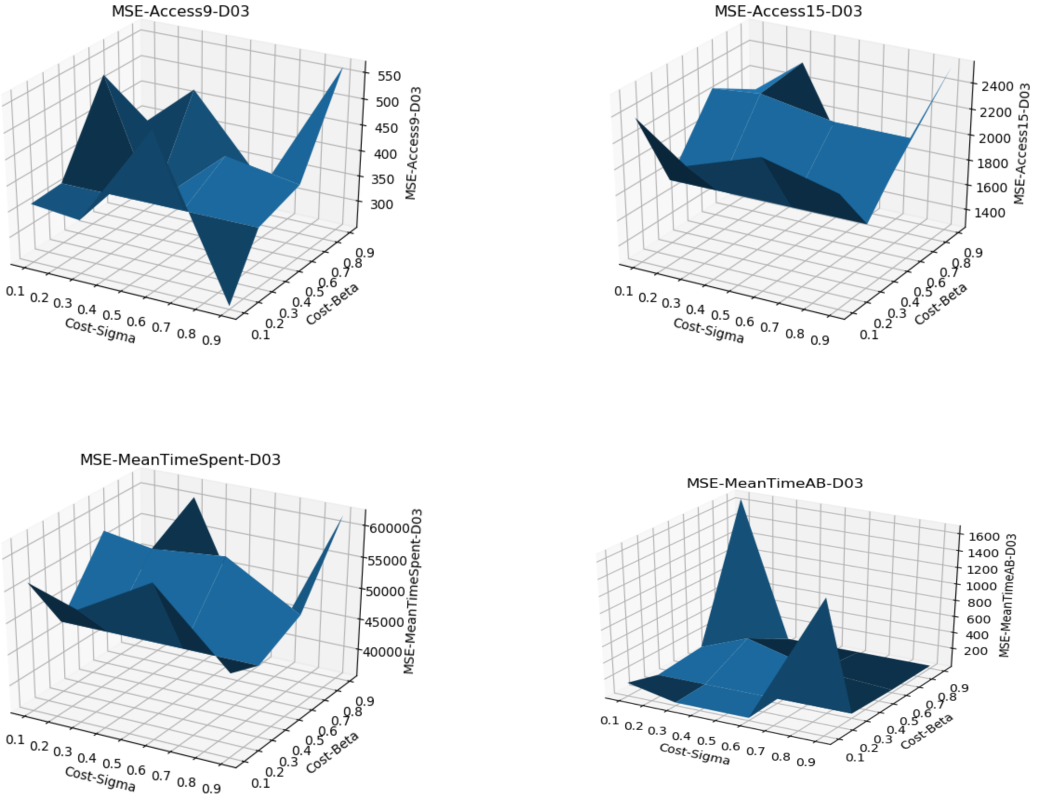


Figure 5.4: `MSE` evolution for $\delta = 0.3$(Generated with matplotlib). Source: own compilation

Error with $\sigma = 0.6$:

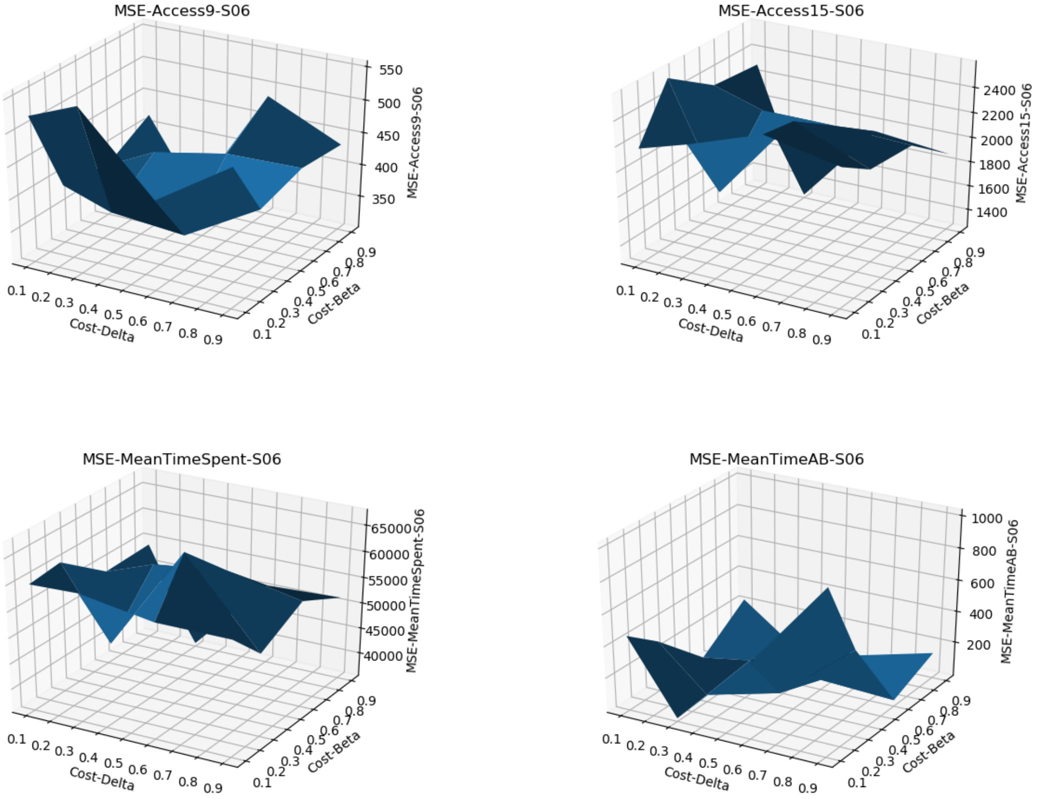


Figure 5.5: `MSE` *evolution for* $\sigma = 0.6$*(Generated with matplotlib). Source: own compilation*

Set2

Set 2 experiments aimed to demonstrate that the model can represent the real system and that a complete day execution can produce results within less than a week. One full day experiment, with 10835 Agents was successfully executed in about 30 hours, seeing that the experimentation time was limited, the team decided to execute sets2, 3 and 4 with less Agents and until peak hours of the day to explore a wider range of experiments.

As said before the output values obtained from this set will serve as a baseline for see the impact on the pedestrian flow in experiment sets 3 and 4.

The `MSE` error in general has augmented comparing it with the set1 experiments, which was expected, since the experiments are bigger and more Agents and more time lead to bigger discrepancies between the data and the simulation's output.

The obtained baseline is show in the Table 5.2 below:

	mseAccess9	**mseAccess15**	**mseTimeSpent**	**mseTimeAB**
experiment1	4125.73	6802	139906	0.04
experiment2	4444.86	6871.26	141060	187.46
experiment3	4453.86	7264.2	157898.01	217.92
experiment4	4451.53	7149.53	153838.71	27.44
experiment5	4290.93	7091.66	158905.11	543.14
experiment6	4290.93	6976.8	154870.48	434.04
experiment7	4223.06	7173.4	153094.8	0.59

Table 5.2: `MSE` obtained on experimentation set2. Source: own compilation

The `MSE` is really high on average, on all of the `KPIs`. This was expected since the prototype is not yet complete and need some more features and data to be calibrated correctly. A lot of the error comes as well form the way the Agents move themselves. We can see that most of the error is on the time spent at the venue by the Agents, and in the simulation they leave the venue a lot faster than what is shown by the data, increasing the error on the exit `KPIs` as well. This is not surprising given that we have not simulated the time agents spend inside the venue - in the museum and the stadium.

From this set we can extract that the movement algorithm of the Agents must be refactored, as well as that further data analysis will be necessary to complete the model and be able to better represent the system. This executions have shown us that even though the Agents move with a correct average speed between two checkpoints they spend and incorrect amount of time. The Agents in the simulation spend less time, in general, than the shown on the data. The model must be refactored to match this more accurately.

Set3

The emergency evacuation experiments demonstrated that in every one of the experiments all of the Agents managed to evacuate the facility within the limit of 400s. The time of evacuation in each one of the runs is shown on Table 5.3, as well as the mean time necessary to evacuate the venue.

	Exit time (s)
experiment1	131
experiment2	188
experiment3	199
experiment4	178
experiment5	184
experiment6	159
experiment7	152
Mean exit time	170.14

Table 5.3: Exit time of set 3 experiments. Source: own compilation

On this experiment set bottlenecks have not been detected with the current topology. The only point of conflict found would be near access 9, represented in Figure 5.6. Most of the Agents find it to be the closest exit. Nevertheless, on every one of the situations the Agents managed to exit all through access 9 without much delay.

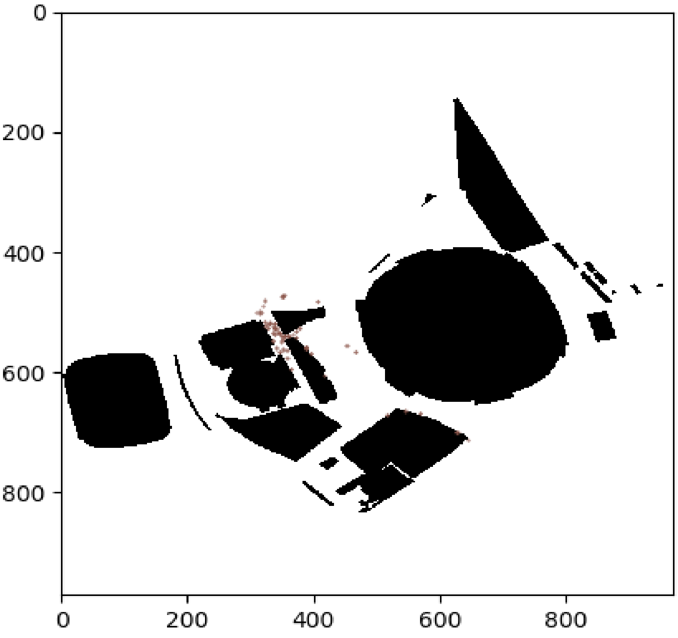


Figure 5.6: Agglomeration near Access9 on evacuation. Source: Own Compilation

Set4

Finally, set 4 has confirmed that the movement algorithm need to be refactored to be able to better sort the various obstacles that conform the topology. The experiments have show that the algorithm has trouble of the agents reach a dead-end and have to go around a large obstacles going away form their target.

Table 5.4 shows the changes to the `KPIs` values compared with those obtained in experiment set 2:

	mseAccess9	**mseAccess15**	**mseTimeSpent**	**mseTimeAB**
experiment1: Set1	4125.73	6802	139906	0.04
experiment1: Set4	5477.53	425.3	6499.32	10985.13
experiment2: Set1	4444.86	6871.26	141060	187.46
experiment2: Set4	4424.13	7114.2	158089	11.05
experiment3: Set1	4453.86	7264.2	157898.01	217.92
experiment3: Set4	5008.06	5217.33	7157517.05	438.88
experiment4: Set1	4451.53	7149.53	153838.71	27.44
experiment4: Set4	10568.46	944.4	42242909.4	51.69
experiment5: Set1	4290.93	7091.66	158905.11	543.14
experiment5: Set4	5408.4	415.53	6275.05	33054.87
experiment6: Set1	4290.93	6976.8	154870.48	434.04
experiment6: Set4	5311.26	357.8	4347.54	106.45
experiment7: Set1	4223.06	7173.4	153094.8	0.59
experiment7: Set4	5451.66	746.26	28268.57	316.21

Table 5.4: `MSE` comparison of set 2 and set 4. Source: own compilation

From these executions we can see that in general terms the Time spent error is increased or decreases. This is due to some Agents not being able to find their way around the new obstacles and being in the simulation during a lot of steps, this way if the majority of the Agents get stuck the time spent is lower since they never leave the simulation. Below, in Figure 5.7 and Figure 5.8, show two examples of where the Agents get stuck and cannot find the way around indicating that the current movement algorithm requires improvement.

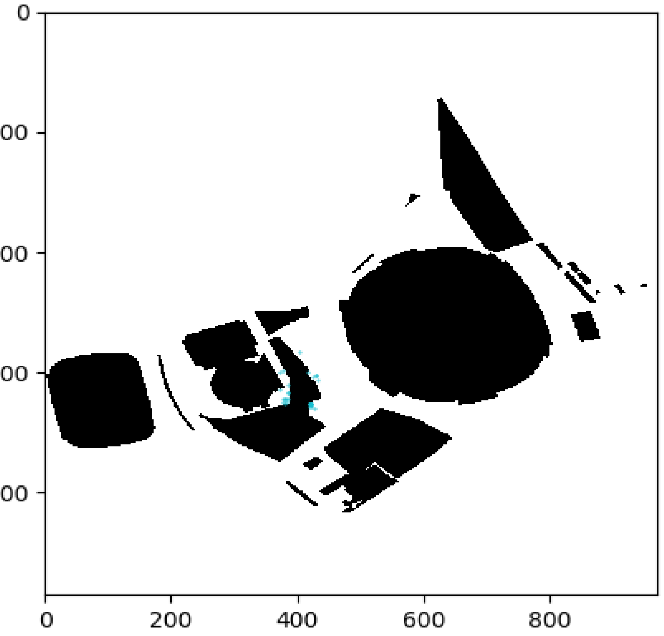


Figure 5.7: Agglomeration during experiment 5. Source: Own Compilation

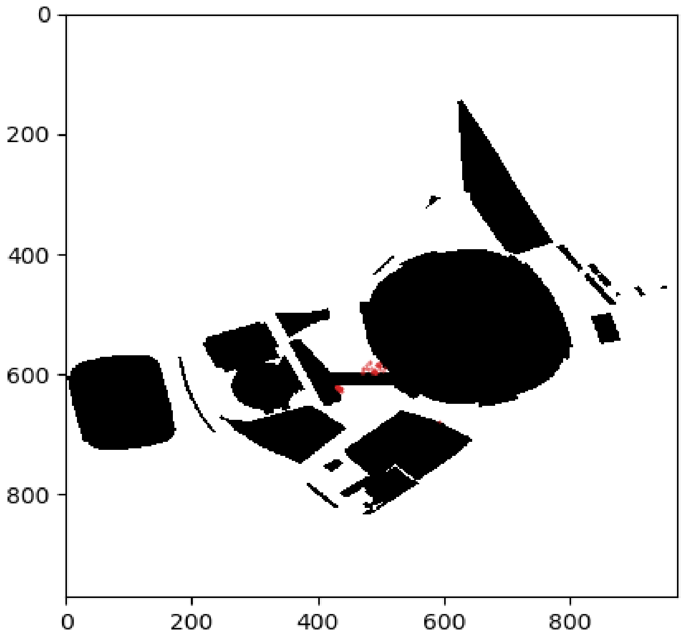


Figure 5.8: Agglomeration during experiment 6. Source: Own Compilation

5.7 Validation

The validation strategy proposed can be divided into three levels : model validation, data validation and result validation.

Regarding the model validation the strategy used is the white-box strategy proposed on *Agent-based simulation validation: A case study in demographic simulation* [30]. This strategy consists on evaluate the model based on its execution. Each one of the model features and modules is analyzed individually to assure the correctness of the programmed model, thus verifying it, and its behaviour during the execution, validating it. The model is validated using expert validation, in this case in charge of Dr Iza Romanowska co-director of this Bachelor Thesis and expert in `ABM` simulation.

The data validation is needed to assure that the data used to build the scenarios are sufficiently representative and robust and that it reflects the behaviour real system. The Data Analytics and Visualization team is in charge of this supervised by the PI of the IoTwins at the `BSC` Dr.Fernando Cucchietti.

Finally, the results validation strategy proposed is the black-box strategy presented on the same paper. This strategy consist on treating the model as a black box. A certain validation dataset is defined and the results of the model must be compared with this dataset. The objective is to assure that the model represents the real system accurately enough using different sets of inputs and under different situations. This is shown on the previous section, section 5.6, with the comparison of the data with the output values of the simulation.

5.7.1 Issues of Agent Based Models Validation

It is known that Agent based models are not a perfect representation for the time being. One of their key properties, emergence, can make these type of models difficult to validate. They have a high variance and they demand a finer programming of the model and accurate input data. This issues increment with a model aimed to be able to represent multiple systems. Making the scenario and data gathering phases critical for the correct calibration of the model [30].

Chapter 6

Project Management

On this chapter the project temporal planning will be bracken into the different tasks needed to accomplish the correct development of the project. The tasks will be described and time-estimated individually. And the dependencies between tasks will be acknowledged. The human resources needed to the project and the risks of developing it are going to be described as well. The economic planning will be exposed afterwards. The sustainability of the project and the laws and regulations affecting the project will be described. Finally the affectations on the daily work and the overall project will be described.

6.1 Temporal Planning

6.1.1 Task description

The IoTwins project started September 2020. It's planed to finish in September 2023. This Bachelor Thesis has been developed between September 2019 and July 2020 with a dedication of 800 hours, divided in 20 hours per week. In this Chapter all the Thesis tasks are going to be described and estimated.

The management tasks (`MT`) are the following ones:

`MT.1` Daily meetings

Every day the `HPC4SC` team invests ten minutes into daily meetings. These meetings help to keep updated the state of the project for the whole team. The meetings are held in the team's office.

`MT.2` Meeting with other `BSC` teams

Every two weeks the three teams of `BSC` that are involved into IoTwins, Data Analytics and Visualization, Computational Social Sciences and Digital Humanities and `HPC4SC`, meet during two hours to update the advances made in the last two weeks.

`MT.3` Meeting with other partners

Every six months all of the different partners of the IoTwins project meet during two days to comment the advances of each test-bed.

`MT.4` Documentation

The documentation of this Thesis will be develop in parallel with the whole design and implementation.

The technical tasks are divided in four different task groups, matching the development phases: design (`DT`), implementation (`IT`), validation (`VT`) and experimentation (`ET`).

`DT.1` Problem analysis

To propose a correct solution on the early stages of the project an extensive research of the different techniques must be done.

`DT.2` Definition of the model´s ontology

We define the entities (agents) and their variables as well as actions determining how the Agents behave, who performs given action and how they do it.

`DT.3` Describe activities flow - scheduling of the model

The activities in the system follow a logic order of occurrence. This order is conceptualized as a flow diagram facilitating the implementation of the model and documentation.

`DT.4` Describe the different agent types

The model will allow for multiple type of agents with different behaviours. Based on sex, age, tourist-local status, this distinction is necessary for assigning attributes and behaviour patterns to each type.

`DT.5` Define utility function

The utility function that some Agents will use to move must be mathematically correct and specified as an equation.

`DT.6` Conceptualize the system

The system must be conceptualized as a UML to analyze the key activities and concepts of the system. This task is necessary for facilitating implementation and to complete the documentation.

`DT.7` Shape input data

To be able to coordinate with the Data Analytics and Visualization team at `BSC` the input files needed to run the model must be properly defined and specified. Also the data must be adapted into PANDORA's input format.

`IT.1` Implement a general space definition

The model will support multiple map representations. This necessitates implementation of general space, that will allow for different map inputs.

`IT.2` Implement Agent characterization

The Agents must be implemented according to the description of their agent types, made in `DT.5`.

`IT.3` Implement Actions

The activities defined in `DT.3` are going to translate into Actions in the `ABM` model. These Actions must be implemented respecting the flow defined in `DT.4`. Special attention to MoveAction, since it will be the most complex of them all.

`IT.4` Implement Unit Tests

To assure the correctness of the code the implementation of Unit tests is mandatory.

`VT.1` Define the validation strategy

The validation strategy must be defined to satisfy the requirements and correctly test the model's execution.

`VT.2` Implement the validation script

A script will be implemented to execute automatically the validation of the different versions of the model.

`ET.1` Experiment definition

Several experiments will be designed for each scenario, including a sensitivity analysis, a wide parameter sweep and scenario testing.

`ET.2` Experimentation

Since the experiments will be ran in an `HPC` environment a script will be developed to input the different data and analyze the experiment's results. Several experiments will be run for each scenario we want to study.

`ET.3` Result analysis

The results of the experiments must be analysed and visualised. If the results fall within the validation standards then the results will be analyzed and commented with the project stakeholders - in particular, other teams involved in the project.

Task Table

The table below, Table 6.1, show all of the summarized tasks, the dependencies between them and the estimation of hours of each task.

Task	**Task name**	**Estimated time (hours)**	**Dependencies**	**Specific resources**
MT.1	Daily meetings	30		
MT.2	Meeting with other BSC teams	40		
MT.3	Meeting with other partners	20		
MT.4	Documentation	170		
DT.1	Problem analysis	40		
DT.2	Definition of the model´s ontology	20		
DT.3	Describe activities flow - scheduling of the model	10	**DT.2**	
DT.4	Describe the different agent types	20	**DT.2**	
DT.5	Define utility function	10	**DT.4**	
DT.6	Conceptualize the system	20	**DT.2** **DT.3** **DT.4**	
DT.7	Shape input data	10		
IT.1	Implement a general space definition	40	**DT.2**	Data to define the space
IT.2	Implement Agent characterization	30	**DT.4**	Data for agent characterization
IT.3	Implement Actions	120	**DT.3** **IT.2**	
IT.4	Implement Unit Tests	30	**IT.1** **IT.2** **IT.3**	
VT.1	Define the validation strategy	10		
VT.2	Implement the validation script	30	**VT.1**	Data for comparing with simulation results
ET.1	Experiment definition	30		
ET.2	Experimentation	60	**IT.1** **IT.2** **IT.3**	HPC environment
ET.3	Result analysis	60	**ET.3**	

Table 6.1: Summarized project tasks. Source: own compilation

6.1.2 Resources

The human resources used in the test-beds 5 and 11 of the IoTwins project are the Research Student, the author of this Bachelor Thesis. It will be in charge of the definition, implementation, validation and experimentation tasks. The Researcher will coordinate the Research Student and overview the design the data input and output of the model and experiments. The Principal Investigator will supervise the design, validation and experimentation tasks of the Research Student.

The necessary hardware resources are the following ones: one computer for each development team member for developing all of the tasks and the `HPC` cluster where the experiments will be done. This resources are needed in task `ET.2`.

Data resources will be needed to define the system and characterize the model, as seen above more specifically in `IT.2`, `IT.3` and `VT.2`.

Finally the software resources needed for the project are Windows 10[31] licences, all of the tasks. Ubuntu[32], the operating system of the cluster service, Docker[17] for running the simulation of the cluster. And text editors for the documentation, Overleaf[33] and Microsoft Visio[34], and as a programming Integrated Development Environment (`IDE`) Visual Studio Code[35].

6.1.3 Time estimation

Figure 6.1 represents the Gantt Diagram of this Bachelor Thesis. This diagram shows the development of the project in time and the dependencies between the tasks.

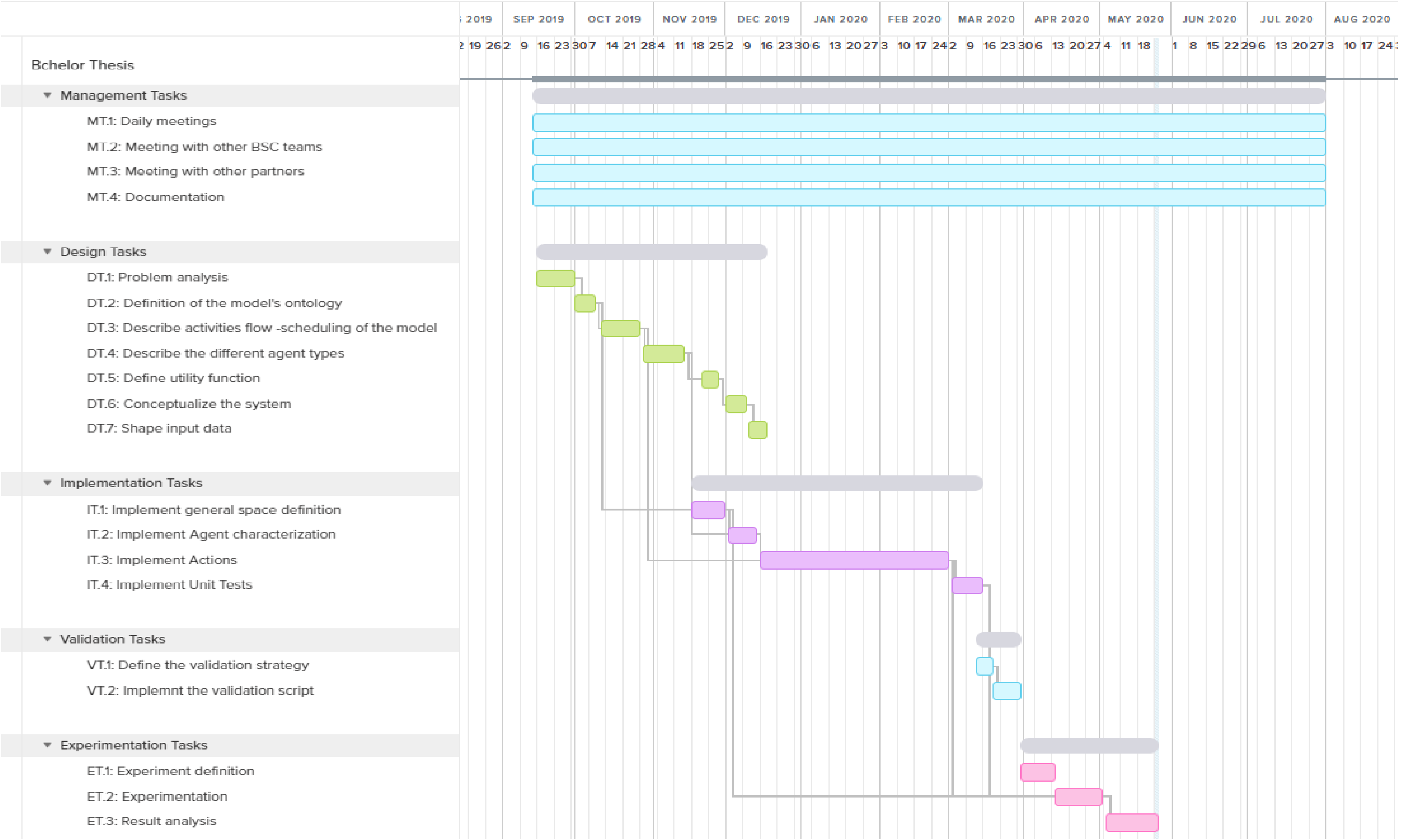


Figure 6.1: Thesis Gantt diagram (generated with TeamGantt). Source: own compilation

6.1.4 Risk management

In this section the strategies to solve or mitigate the impact of the obstacles previously identified will be exposed.

Temporal limitations

The few weeks before the ending of the project can be destined to solve the delay instead of investing them into the documentation. This way we have some a margin of 40 hours. This time will be invested into finish delayed tasks by the Research Student.

Dependency of other partners

On the case that the other teams do not accomplish their tasks in time substitute data will have to be found. If there isn't any mocked data will be used instead. This data gathering will take 30 hours and will be done by the Researcher and Research Student.

Implementation complexity

More time will be invested on the implementation tasks. If the problem proves to be of a design kind, then the design tasks should be refactored. The implementation ones affected will be refactored as well. This refactor will be done by the Research Student. This will have a heavy impact in the project hard to estimate in hours precisely.

PANDORA's performance

Since the execution of the framework will take longer, less experiments will be run. Affecting the experimentation tasks, but not affecting the whole road-map of the project.

6.2 Economic Planning

In this section the budget of the project will be described. The different elements taken into account will have their cost estimated as well.

6.2.1 Identification of costs

The main cost of the project are staff costs. We have to take into account the equipment amortization of the hardware used, as well as the licenses of privative software. Moreover, contingencies and incidental costs will be taken into account as well.

Table 6.2 shows the amount of time dedicated, in hours, by each profile to the tasks defined previously. This tasks can be seen in detail in the Task description chapter.

Task	Research Student	Researcher	Principal Investigator
MT.1	30	30	30
MT.2	40	30	30
MT.3	20	20	20
MT.4	170	0	0
DT.1	40	0	0
DT.2	20	10	10
DT.3	10	0	5
DT.4	20	0	5
DT.5	10	0	0
DT.6	20	0	0
DT.7	10	10	0
IT.1	40	0	0
IT.2	30	0	0
IT.3	120	0	0
IT.4	30	0	0
VT.1	10	0	10
VT.2	30	0	0
ET.1	30	30	30
ET.2	60	20	0
ET.3	60	0	60
TOTAL	**800 hours**	**150 hours**	**200 hours**

Table 6.2: Staff dedication per task. Source: own compilation

6.2.2 Cost estimates

Table 6.3 indicates the staff, described before in this document, cost of this Bachelor Thesis. The price hour shown is gross and social security ha been taken into account as well.

Profile	Quantity	Hours	€/hour	Cost
Principal investigator	1	200	40€	8.000€
Researcher	1	150	22€	3.300€
Research Student	1	800	9€	7.200€
TOTAL				18.500€

Table 6.3: Staff cost estimation. Source: own compilation

Table 6.4 shows the amortization of the hardware (`HW`) and the licences of the privative software (`SW`) cost. These amortizations have been provided by the `PI` of the project.

Resource	Quantity	Cost
Windows 10	1	55€
Micrsoft Visio	1	335€
Laptop	1	100€
HPC Cluster	1	5.000€
TOTAL		5.490€

Table 6.4: Hardware and software cost estimation. Source: own compilation

Finally, Table 6.5 we have to take into account the indirect costs of the project. These costs are related with the office, rental, electricty, etc. This information has been provided by the sales department at `BSC`.

Resource	Cost
Office	5.000€
Furniture	300€
Office exprenses	3.000€
TOTAL	8.300€

Table 6.5: Indirect cost estimation. Source: own compilation

6.2.3 Management control

During the development of the project the staff and direct costs will be monitored in monthly meetings with all of the `BSC` teams involved in the IoTwins project. Taking into account that with an increase of the 10% of the Student Researcher's the alternate actions described in the Risk Management section will be accomplished and that the total cost of this developer is 7.200€. The temporal deviation is valued in 720€.

A 5% off the budget will be dedicated to contingency, to face unplanned expenses.

At the end of the project the final expenses will be compared with the initial budget and then it will be determined if the temporal deviation line item can assume the costs. If not the contingency funds will be used to cover the difference.

Type	Cost
Staff	18.500€
HW and SW	5.490€
Indirect	8.300€
Temporal deviation	720€
Contingency	1.650€
TOTAL	34.660€

Table 6.6: General Budget. Source: own compilation

6.3 Sustainability

Even though UPC is trying to make their students more aware of the sustainability issue, I think there is much more to learn. On many subjects this is treated only as a class project that has no interest. I feel there should be more tools available to learn about the issue that is going to be the center of the new Horizon Europe(2030). Sustainability is going to be the center of the technological and scientific development in the next years, and we do not know about it enough.

Nevertheless, since this Thesis develops an HPC application let's focus on that. HPC clusters consume vasts amount of electricity, Mare Nostrum 4 consumes 1.3MW per year. The supercomputers are fed with electricity that comes from not renewable energy. supercomputing centers are only interested in having more powerful machines and in maintaining them, but I feel more attention should be put into the environmental impact of these machines.

After doing the survey I've realized that in any of the projects of the Bachelor I haven't done any proper sustainability analysis. I think this kind of approach should be more in mind for all of us developing computer science projects.

6.3.1 Economic dimension

These project's objective is to simulate how pedestrians behave in certain spaces. In this case FCB facilities. For now the budget presented before in this document is adequate for a first prototype. Once all of the results of the simulations have been analyzed and the necessary changes made on the field, FCB will decide if this project is viable economically.

6.3.2 Environmental dimension

As said before, this Thesis involves the experimentation in HPC clusters. This issue should not be overlooked since this machines consume a lot of energy. The experiments must be well planned beforehand to avoid errors that could make repeat the experiments.

6.3.3 Environmental dimension

As said before, this Thesis involves the experimentation in HPC clusters. This issue should not be overlooked since this machines consume a lot of energy. The experiments must be well planned beforehand to avoid errors that could make repeat the experiments.

6.3.4 Sustainability matrix

To end with the sustainability analysis it is summarized in the Table6.7 representing the sustainability matrix of the project. The matrix summarizes the exposition made in the previous sections.

	PPP	**Useful live**
Economic	8	7
Environmental	4	3
Social	7	7

Table 6.7: Sustainability matrix. Source: own compilation

6.4 Laws and regulations

In this section, the laws and regulations affecting this project are going to be acknowledge here.

The tool developed on this Bachelor Thesis will not store any of the input data of the model. We need to remind, that the data using to characterize the Agents is extracted from public studies and has been correctly anonymized. Even though the tool will be open source it's okay to remind that all the future data used on the project must respect the rules and guidelines of the General Data Protection Regulation (GDPR)[36].

6.5 Modifications due to Covid-19's confinement

Due to the confinement declared with the *estado de alerta* ordered by the *Gobierno de España* the development of this Thesis and the whole IoTwins project at `BSC` has been affected. Since March 13 all of `BSC` teams decided to start teleworking from home. The daily meetings by the `HPC4SC` where suspended and the team arranged weekly meetings with the Computational Social Sciences and Digital Humanities team, while maintaining the two week iterations. The communication carries on this meetings and using e-mail.

The confinement delayed the data gathering infrastructure installation further on the future and to this day the start date of data gathering remains unclear. Monthly meetings with the whole `BSC` team where also suspended and the communication of teams advances was under the responsibility of the group leaders Dr.Iza Romanowska and Dr.Fernando Cucchietti.

Chapter 7

Knowledge integration

This chapter will describe how the knowledge gained on the degree has been applied on this Bachelor Theses. The useful subjects will be described and the specific knowledge of each subject. Then it will be justified why this Bachelor Thesis is within the Software speciality of the *Facultat d'Informàtica de Barcelona*. Finally, the technical competences of the Thesis will be described and at which degree this competences have been achieved.

7.1 Knowledge learned in subjects

Next the subjects of the Software speciality and the knowledge applied to the Thesis of each one of the subjects.

Enginyeria de Requisits (ER)

This subject gives a wide capacity to conceptualize systems and to describe the requisites of the planned application. The specification is the main topic of this subject. It gives notions of how to describe the application functionalities through different mechanisms. The most used are the use cases and user histories.

Gestió de Projectes Software (GPS)

GPS gives a wide knowledge on how to plan the development of a software project and how to organize the development team during the duration of the project. This subject also gives knowledge of how to detect the dependencies between the defined tasks.

Simulació (SIM)

Most of the knowledge used of this Thesis comes from the simulation subject. This subject gives the tools necessary to develop simulation projects. All of the development process is seen on the subject. From the study of the system, passing from the development of the model, the experiment design and the verification and validation of the model. Simulation also gives tools and understanding of the generation and usage of the random numbers on computer science. All of the knowledge gathered on this subject has been key for the correct development of the project.

7.2 Speciality Justification

The Thesis' main topic is to design, specify and develop a prototype of a generic simulation model for crowd modelling. The application must be designed and implemented and with a approach rarely used on `ABM` that will allow the model to become a general one, while keeping the computational cost reasonably low. It must be designed to be incorporated on a bigger system. This process addresses directly the following topics:

- **Conceptualization of a real system**. The conceptualization will allow the specification and the approach to the proposal of a general tool.
- **Development of a simulation model**. Together with the design and specification. A simulation model is a software tool or application. It has to be correctly conceptualized, specified and implemented.
- **General tool simulation**. The tool will be able to represent crowd systems using a generic logic with dynamic obstacle detection algorithms.

All of the previous topics are within the Software Engineering, thus this justifies this Thesis appropriate for the Software speciality.

7.3 Technical competences justification

The technical competences associated with this project are the following ones:

CES1.1: To develop, maintain and evaluate complex and/or critical software systems and services.[Level: Enough]

CES1.2: To solve integration problems in function of the strategies, standards and available technologies [Level: In depth]

CES1.3: To identify, evaluate and manage potential risks related to software building which could arise. [Level: Enough]

CES1.7: To control the quality and design tests in the software production. [Level: Enough]

CES1.9: To demonstrate the comprehension in management and government of software systems. [Level: In depth]

CES2.1: To define and manage the requirements of a software system. [Level: In depth]

CES2.2: To design adequate solutions in one or more application domains, using software engineering methods which integrate ethical, social, legal and economical aspects. [In depth]

Justification

Now the selection of each one of the technical competences will be discussed and how the level of achievement will be reached.

CES1.1

This competence has been selected because a prototype of a simulation model, is a complex software system. To develop this prototype the system to simulate will be described and a state of the art will be developed on the subject. The various types of solutions will be analyzed and the one that suits better the project will be designed and implemented.

The development of the prototype is the main objective of this Thesis. But since the evaluation will be partial due to not having the correct data to validate the model the level of achievement will be enough.

CES1.2

The project has been developed using a software framework that only works for a certain Ubuntu distribution. The system has been integrated using container technologies to solve this issue.

The achievement level will be enough because multiple container technologies have been used to integrate the system and the models and create a usable tool that is non dependent of the environment.

CES1.3

The whole design and specification of the model is within the scope of the Thesis. This includes the detection of risks and the management and the reduction of them.

The achievement level will be enough because the prototype will be designed and it must not imply vulnerabilities to the future system where the model will be integrated.

CES1.7

The achievement of the quality control will be enough because during all of the development of the tool the code and the performed have been tested using default experiments and comparing how this changes affect the general model. Also on the design aspect, the quality assurance comes as teh result of the continuous discussion of the model with the co-director of this Thesis Dr.Iza Romanowska, expert on `ABM` modelling.

CES1.9

The project will use a container software system to run the simulation engine. The system has to be set-up and configured on various systems. The communication of this containers with the host systems has to be set-up as well.

Since all the development of the project has to be done on a container it has to be correctly installed and configured. Once it is done no more management is going to be done, thus the level will be a little bit.

CES2.1

On the model design the analysis of the requirements has to be done exhaustively to assure the most correct representation of pedestrian movement model. This is more important in this case because the model is a general one, and a wider approach is needed to be able to represent as many systems as possible.

The level of achievement will be in depth because the key feature of the prototype demands a exhaustive analysis of the possible requirements on various crowd simulation systems.

CES2.2

This competence has been selected because the social simulation aims to have a social impact on every model. Even if the model's objective is only to comprehend the system the simulation describes a social activity and with the information deviated from the model social changes can be achieved. Digital twins on the other hand aim to replicate a system, for better managing purposes, thus having a economical impact.

Being this competence one of the backbones of not only the project, but of social simulation in general the level of achievement will be in depth.

Level of achievement

The following table shows the level of achievement of the discussed technical competences in a more visual way:

	CES1.1	**CES1.3**	**CES1.9**	**CES2.1**	**CES2.2**
In depth				×	×
Enough	×	×			
A little			×		

Table 7.1: Level of achievement of the technical competences. Source: own compilation

Chapter 8

Conclusions

In this final chapter the conclusions extracted of the work done will be presented. First, the achievement of the initial objectives will be evaluated. Then, the contributions made by the work contributed in this Bachelor Thesis will be explained. Following that, the future necessary steps to complete the work started in this Thesis will be addressed. Finally, I will explain the conclusions reached by developing this project.

8.1 Objectives achievement

In this section I will reiterate the original objectives in order to evaluate the level of completion for each one of them individually, following the explanation made throughout the document.

The first objective **Design and specify a prototype of a general `ABM` model that represents the pedestrian behavior within a given system to be implemented using the framework `PANDORA`** has been achieved successfully. Chapters 3, 4 and 5 describe the planning and development of the tasks in detail. The specification of the prototype has been developed using the UML to represent the whole system. Use cases have been specified to be able to identify the features that the tool must include. On the other hand, the execution logic of the model has been described on various flow diagrams. These diagrams represent the execution of the whole model, the decision making structure that the Agents follow each step. There is also one diagram representing the execution of each individual action.

Regarding the design, the two environments where the simulation can be executed, local environment and `HPC` environment, are described along with the container technology used of each case, Docker and Singularity respectively. After testing the technologies used, each one of the components of the model have been developed with their characteristics in mind. For the most complex ones, multiple strategies have been considered to implement these components.

Finally, section 5 describes the implementation of the designed prototype. Alternative implementations for each one of the prototype components has been tested. The final implementation of the multiple designed components has been the best performing one of all of the implemented and tested strategies.Nevertheless, the experimentation (section 5.6) with the prototype has shown the flaws of the model and some aspects that need to be refactored in future development.

The second objective **Define and implement the behavior of a single pedestrian** has also been achieved successfully. The behavior of the Agent is defined by its characteristics and the Actions performed. The Agents need to be defined individually to interact using Actions both of which must be specified and designed. The definition of an Agent comprise their decision making strategy and their defining characteristics. In this case the decision making strategy has been designed to feature a general crowd model. First the Agents explore their surroundings, then select the best position avoiding obstacles and then move to the best position in reach. This design's implementation is described on sections 5.1 and 5.4, addressing the Agent and Actions implementation respectively. In future reiterations of the model more detailed characteristics of agents - their local/tourist status, their age, gender and mode of participation (as part of a group or individually) will be explored.

To implement a realistic way of movement, the implementation of the model interprets the World as a cost function. This allows the Agents to dynamically avoid obstacles and the movement creates dynamic obstacles, agent aggregations, so that Agents must find a way around.

Strongly related is the objective number 3: **Define and implement the group dynamics among the pedestrians**. We need to keep in mind that the model's representation must be robust. Since we are working with a crowd model, the Agents are considered as individuals belonging to a larger group, all the pedestrians in the current simulation. To obtain a realistic movement of the pedestrians group dynamics must be added into the model. These group dynamics are applied to all of the Agents, not only the Agents that go in group. They have been taken into account during all of the achievement and implementation of the model. The intervention of group dynamics is explained in detail on sections 5.4.1 and 5.4.2. These restrictions described on Boids [25] have been worked into both the cost and utility functions of the movement actions. This restrictions lead the Agents to move in a more realistic way.

Going on to objective number 4 **Use real data to run the experiments with the model**. This objective has not been achieved successfully. Due to delays between the legal departments on the agreements of both `BSC` and `FCB` the transfer of data necessary to construct the scenarios has not been reached to this day. The author of this Bachelor Thesis was not involved in the data handover so could not prevent this difficulty. To work around this and achieve the following objectives open data from the studies acknowledged and other open data sources has been used. The analyzed data and the construction scenario addressed on detail can be found in section 5.5. As said before, real data has been used to calculate the entrances to the system, but the objective can not be classified as achieved. To summarize, the topological data has been extracted from Open Street Maps and the data for Agent characterization has been obtain from various open studies.

Despite the difficulties described above, objective 5 has been achieved successfully. **The no match day scenario has been defined and implemented**. These scenarios constructed on data are the defining properties of the general model. They are what define the model to a concrete system and drive the simulation. From the given topology and characteristics of movement gathered from the data the input of the general model has been constructed.

Once the scenario has been implemented correctly we can move to the final objective

Experiment and validate the no match scenario. This objective has been achieved successfully. A series of experiments, designed and explained in depth in section 5.6, have been performed with the model and the implemented scenario. Using sensitivity analysis during the experiments the model has been calibrated for a more realistic performance. This calibration aims to obtain the optimal values of the coefficients of the defined cost and utility functions of movement. Future work will further calibrate the parameter values using real-world data provided by the FCB.

Finally, the validation has been performed on two different approaches. The first one is the generation of `KPIs` to quantitative track the performance of the model. The second one has been the supervision on the design of the model by the expert on `ABM` modelling and co-director of this Thesis Dr.Iza Romanowska.

8.2 Contributions

On the one hand, the prototype of the general crowd model can be used to model multiple crowd systems. The purpose of the model is to offer a general tool to be able to represent multiple system designed as model inputs. This tool will help with the comprehension of these systems and will help to optimize them.

On the other hand, the implementation of the representation of the World where the simulation takes place is an approximation not often seen in `ABM` crowd models this big and with such a complex topology. This allows this simulation to represent dynamic obstacles that occur and disappear during the simulation. Following this topic, the dynamic obstacle avoidance algorithm is the main contribution of this Bachelor Thesis. With the arrival of the 5G and more `IoT` devices the amount of data necessary to characterize more systems as scenarios for this model will increase, making this model of use for a diverse range of systems.

8.3 Further work

Even if a lot of work has been performed, a whole lot remains to be done. The IoTwins project has not completed its first year yet and there are still two more years ahead. The data gathering infrastructure needs to be connected to the model. This will allow to define and implement the remaining scenarios described on this Thesis. The experimentation with this variety of scenarios will help to calibrate the model for a more accurate general representation. More features that can be part of a general crowd model will be identified during this process and added to the model to complete it. A more extensive study of the data has to be performed to achieve this. As seen in section 5.6 the amount of data currently available is insufficient to represent the model with the desired granularity.

Scalability tests have to be performed to determine if the approach of the model is the appropriate one for an `HPC` environment. With the validated scenarios various experiments will be performed to test the scalability and replicability of the model. Regarding the use of `HPC`, a new scheduler is being developed and the changes on the model required to use this more efficient scheduler must be implemented in the prototype.

The experiments detected some soft spots on the movement algorithm. Calibration can help improve the performance of the model, but as more scenarios are tested the necessary changes to the movement algorithms will be more obvious. Also, the simulation will be scaled to represent the space in $0.5m^2$. This new granularity will allow to the model to represent the systems on a much more finer way. These changes will allow the team to introduce a variable velocity system that will also help to better represent the movement of the Agents. It is almost certain that during all of the development of the project these algorithms will be evolving to better represent the pedestrian movement. All of the experiment sets 2 and 3 will be performed repeatedly to accommodate the changing algorithms and to perform more experiments of a full day and with the maximum Agent capacity. This new experimentation will allow us to keep gaining knowledge about the system. Also more evacuation situations must be studied, this time changing the topology of the system, as more data on the pedestrian flow becomes available.

Once the architecture of test-beds 5 and 11 has been validated by the `BSC` and `FCB`, the whole project will be available for the other partners of the IoTwins project, establishing the platform with the implementations of all of the test-beds. This will imply the access to many other systems that may be represented by the model. This will enable the team to evaluate the replicability of the approach proposed for the model. Equally, more features that the model may need to become generalised and reimplemented.

Finally, we have to consider one of the newer scenarios defined by the team with the meetings with the `FCB`. The reopening of the facilities after the covid-19 confinement ends. The club wants to simulate the reopening to assure that it will safe for both workers and for the audience. A epidemiological layer will need to be implemented to complete the model.

8.4 Personal assessments

On a more personal note, being part of a Horizon2020 project within a development team using agile methodologies and with experts on the topics, has been really enriching. Moreover, since the project will have an impact on the management of facilities of Barcelona using new approaches to well-known problems gave me an extra bit of motivation.

This project has permitted me to gain deeper knowledge on how a real simulation project is set up and how real complex systems can be conceptualized and worked in and represented into an application that simulates this system. Facing a problem this complex has certainly tested my abstraction capacity and my conceptualization skills. What I got from this project as well is a more in-depth comprehension of the C++ language and what it's capable of in a real project. The project forced me to understand new tools for me and made me constantly think on how it's going to be used is equally important as the code and logic quality. If I develop a tool and nobody can use it is the same as not developing anything at all.

Finally, this project and the `BSC` internship has showed me that the research and investigation world is not how I had imagined during previous years. This project also opened some ideas on me that I would like to investigate on the future. How the topologies affect the movement and the interactions between pedestrians that transit these urban spaces is a line of research that I find really interesting. Even more if the urban planning layer is added to the equation. I think one of the greatest challenges to come is how urban planning has to change to suit the ecological and social demands of our society. Seeing and planning the cities of the future using the simulation and IoTwins tool is a line of research I'm more than interested on following.

References

[1] IoTwins. Iotwins project webside. `https://www.iotwins.eu/`, 2020. [Online; last access 10-April-2020].

[2] Abdulmotaleb El Saddik. Digital twins: The convergence of multimedia technologies. *IEEE MultiMedia*, 25:87–92, 04 2018.

[3] Xavier Rubio-Campillo. Pandora: A versatile agent-based modelling platform for social simulation. pages 29–34, 09 2014.

[4] Cristina Montañola. Sim - fme aleatoritzacio-5. page 2, 04 2016.

[5] Wikipedia. Scrum (software development). `https://en.wikipedia.org/wiki/Scrum_(software_development)`, 2018. [Online; last access 18-january-2020].

[6] Wikipedia. Git version control. `https://en.wikipedia.org/wiki/Git`, 2018. [Online; last access 18-January-2020].

[7] GitHub. Github. `https://github.com/about`, 2018. [Online; last access 18-January-2020].

[8] Trello. Trello. `https://trello.com/en`, 2018. [Online; last access 18-January-2020].

[9] Michael Batty. Agent-based pedestrian modeling - editorial. *Environment and Planning B: Planning and Design*, 28:321–326, 02 2001.

[10] Albert Gutierrez-Milla, Francisco Borges, Remo Suppi, and Emilio Luque. Crowd turbulence with abm and verlet integration on gpu cards. volume 80, pages 2428–2432, 12 2016.

[11] Pau Fonseca i Casas, Josep Casanovas, and X. Ferran. Passenger flow simulation in a hub airport: An application to the barcelona international airport. *Simulation Modelling Practice and Theory*, 44:78–94, 05 2014.

[12] Fasheng Qiu and Xiaolin Hu. Spatial activity-based modeling for pedestrian crowd simulation. *SIMULATION*, 89:451–465, 04 2013.

[13] Maddegedara Lalith, Wasuwat Petprakob, Leonel Aguilar, Muneo Hori, and Tsuyoshi Ichmura. Scalable hpc enhanced agent based system for simulating mixed mode evacuation of large urban areas. *Transportation Research Procedia*, 34:275–282, 01 2018.

[14] Soraia Musse and Daniel Thalmann. *A Model of Human Crowd Behavior : Group Inter-Relationship and Collision Detection Analysis*, pages 39–51. 01 1997.

[15] Paul Richmond and Daniela Romano. A high performance framework for agent based pedestrian dynamics on gpu hardware. 08 2009.

[16] Norman Johnson. *Flocks*, pages 231–252. 10 2019.

[17] Docker. Docker. `https://www.docker.com/resources/what-container`, 2018. [Online; last access 01-March-2020].

[18] Docker. Docker volumes. `https://docs.docker.com/storage/volumes/`, 2020. [Online; last access 04-April-2020].

[19] GDAL. Gdal. `https://gdal.org/`, 2020. [Online; last access 24-April-2020].

[20] QGIS. Qgis. `https://qgis.org/es/site/`, 2020. [Online; last access 24-April-2020].

[21] theHDFGroup. Hdf5. `https://portal.hdfgroup.org/display/support/Documentation`, 2006. [Online; last access 20-April-2020].

[22] Parth Mehta, Hetasha Shah, Soumya Shukla, and Saurav Verma. A review on algorithms for pathfinding in computer games. 03 2015.

[23] Wikipedia. Dijkstra's algorithm. `https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm`, 2020. [Online; last access 3-May-2020].

[24] Wikipedia. Greedy algorithm. `https://en.wikipedia.org/wiki/Greedy_algorithm`, 2020. [Online; last access 3-May-2020].

[25] U. Wilensky. Netlogo flocking model. `http://ccl.northwestern.edu/netlogo/models/Flocking`, 1998. [Online; last access 4-May-2020].

[26] FCB. Fcb museum visitors. `https://www.fcbarcelona.com/en/news/814066/fc-barcelona-museum-preparing-for-30-millionth-visitor`, 2020. [Online; last access 29-April-2020].

[27] Andrew Crooks, Arie Croitoru, Xu Lu, Sarah Wise, John Irvine, and Anthony Stefanidis. Walk this way: Improving pedestrian agent-based models through scene activity analysis. *ISPRS International Journal of Geo-Information*, 4:1627–1656, 09 2015.

[28] Satish Chandra and Anish Bharti. Speed distribution curves for pedestrians during walking and crossing. *Procedia - Social and Behavioral Sciences*, 104, 12 2013.

[29] Alessandro Pluchino, Cesare Garofalo, Giuseppe Inturri, Andrea Rapisarda, and Matteo Ignaccolo. Agent-based simulation of pedestrian behaviour in closed spaces: a museum case study, 2013.

[30] Cristina Montañola Sales, Bhakti Onggo, and Josep Casanovas. Agent-based simulation validation: A case study in demographic simulation. 01 2011.

[31] Microsoft. Windows 10. `https://www.microsoft.com/es-es/windows/features`, 2018. [Online; last access 01-March-2020].

[32] GNU. Ubuntu operating system. `https://ubuntu.com/about`, 2018. [Online; last access 01-March-2020].

[33] Overleaf. Pverleaf. `https://es.overleaf.com/about`, 2020. [Online; last access 01-March-2020].

[34] Microsoft. Microsoft visio. `https://products.office.com/es-es/visio/flowchart-software`, 2020. [Online; last access 01-March-2020].

[35] Microsoft. Visual code studio. `https://code.visualstudio.com/docs`, 2020. [Online; last access 01-March-2020].

[36] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide.* Springer Publishing Company, Incorporated, 1st edition, 2017.