L. Janssens

Modelling information-flow in formal organisations

Case study of the Dutch Military Air Transport Unit

TU Delft

Modelling information-flow in formal organisations

Case study of the Dutch Military Air Transport Unit

By

L.M.I. Janssens

in partial fulfilment of the requirements for the degree of

Master of Science
in Engineering and Policy Analysis

at the Delft University of Technology,
to be defended publicly on 05/12/2022

Supervisor:	Prof. Dr. T.C. Comes	TU Delft
Thesis committee:	Prof. Dr. Ir. J. Kwakkel,	TU Delft
	Drs. Y. Lont,	TU Delft
	E.van Shaik,	Ministerie van Defensie

Preface

The thesis in front of you concludes my final project during my master Engineering and Policy Analysis at the TU Delft. It is the product of 8 months of hard work. I would like to thank my supervisors for their support and help during the different steps that finally resulted in this thesis in front of you.

Thank you Tina, for your support during our meetings. Your questions and comments enabled me to get to the bottom of problems and look at so many different aspects of this thesis. Our meetings always made sure that I continued with renewed energy on the problem at hand.

Thank you Jan, for the different perspectives you brought in the meetings. Urging me to rethink the phrasing and wording that I used throughout the thesis. Your feedback allowed me to increase the quality of my research.

Thank you Yvonne, for all the support throughout the process. Our brainstorming sessions at the PKC helped me through some of the initial problems in defining the research and gathering data. Later on when trouble arrived during the implementation process our meetings helped me find solutions that I did not think off.

Thank you Eline and all people at the Ministry of Defence who helped me during my thesis project. Your help in finding my way through a complex organisation like the Ministry of Defence was critical. Special thanks for the people at the ATU for their help and support even though your schedules were already full.

Thank you to all my friends and family who supported me with this turbulent journey. Allowing me to rant when there was a bug in the model that I could not find, listen to me rambling when I found it but did not know yet how to solve it, and later when I could not stop talking about the fact that it worked. Your support was truly appreciated. An additional thank you is due to my dear friend Avi and my girlfriend Cheyenne who spent hours of their time reading and re-reading my thesis.

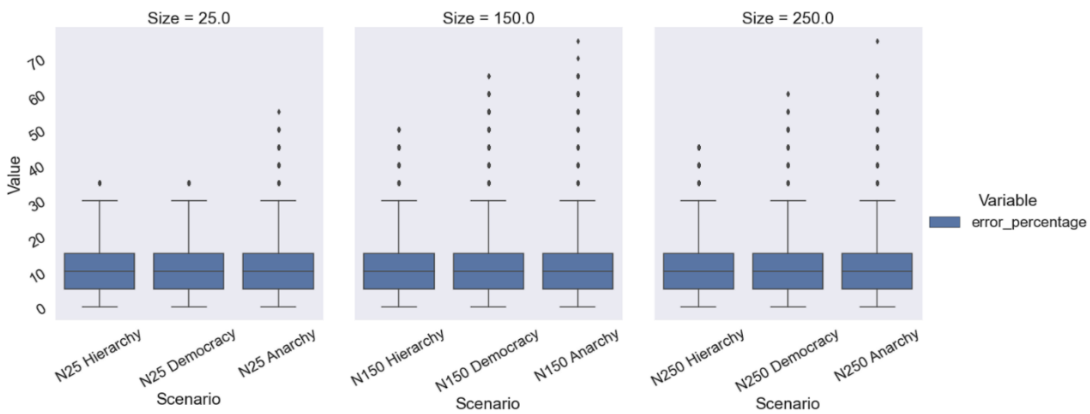
Please enjoy reading this thesis.

Luka Janssens

Executive Summary

The Dutch Ministry of Defence fulfils a crucial role in keeping the Dutch kingdom safe. From missions overseas to providing aid in the case of natural disasters, correct and efficient information-flow is critical for their operations. When a situation occurs, everything is dropped to make sure that they can handle it. The Royal Dutch Army, Navy, Airforce and Marechaussee make up the armed forces part of the military. Besides the branches of the armed forces, there are two support branches. Under one of them, Defensie Ondersteuningscommando, the Air Transport Unit of the Dutch military is located. This unit plans and advises on all forms of air transport for the Ministry of Defence, a unit where correct and timely information is flow critical. This thesis attempts to answer the following research question:

What are the effects of formal network structures within public organisations on their information-flow quality?

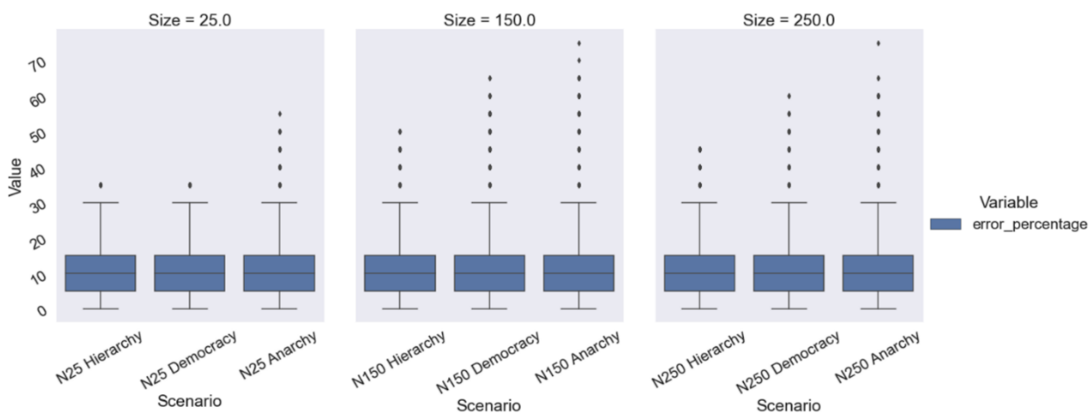


Attempting to fill gaps in literature surrounding the effect of formal network structures on information-flow quality. This study first defines the concept information-flow which is used for the KPIs. The effects of formal network structures on information-flow quality are studied using an ABM model developed in this thesis. The model is tested on the Air Transport Unit network and a diverse set of randomly generated networks. In this case, key findings from the model results indicate that in the case hierarchy, a more hierarchical structure positively affects their data correctness. However, this more hierarchical structure negatively affects their information timeliness, as is shown in the Figure above. Employee busyness does not seem to be affected by hierarchy. Other findings relating to the number of employees whose function it is to control data for mistakes, have significant effects on improving data correctness while at the same time, negatively affecting the timeliness of information. Indicating that the optimum, hierarchical, or less hierarchical organisation lies with the ambitions and goals of an organisation.

Executive Summary - NL

Het Ministerie van Defensie heeft een cruciale rol in het veilig houden van het Nederlands Koninkrijk. Van overzeese missies tot hulpverlenen bij natuurrampen, een juiste en efficiënte informatiestroom is belangrijk voor het uitvoeren van hun operaties. Wanneer een situatie zich voordoet, wordt het "alles laten vallen" om ervoor te zorgen dat het Ministerie van Defensie de situatie aan kan. De Koninklijke Nederlandse Landmacht, Luchtmacht en Marechaussee vormen samen de krijgsmacht van het Nederlandse leger. Naast de krijgsmacht zijn er nog twee supportonderdelen. Onder één van deze supportonderdelen, het Defensie Ondersteuningscommando, valt de Lucht Transport Eenheid van het leger. Deze eenheid is verantwoordelijk voor het plannen van, en het geven van advies over alle vormen van lucht transport van het ministerie van defensie. Voor deze eenheid is juiste en efficiënte informatiestroom cruciaal. Het onderzoek in deze scriptie probeert de volgende onderzoeksvraag te beantwoorden:

Wat zijn de effecten van formele netwerkstructuren binnen publieke organisaties op de kwaliteit van de informatiestroom?



Om de gaten in de literatuur te kunnen vullen wat betreft de effecten van formele netwerkstructuren op de kwaliteit van informatie stroom, definieert deze studie eerst wat er kan worden verstaan onder informatiestroom. De effecten van formele netwerkstructuren op de kwaliteit van informatie stroom worden onderzocht aan de hand van een ABM-model dat is ontwikkeld voor deze Scriptie. Het model is getest op een netwerk van de lucht transport unit en op verschillende willekeurig gegenereerde netwerken. Belangrijke resultaten van het model tonen aan dat in het geval van hiërarchie geldt dat een meer hiërarchische structuur een positieve invloed heeft op de correctheid van de data. Echter, resulteert een meer hiërarchische structuur ook in een negatieve invloed op de tijdigheid van de informatie. De drukte van de medewerken lijkt niet beïnvloed te worden door de hiërarchie van het netwerk. De hoeveelheid werknemers wiens taak het controleren van fouten in de data is, heeft significante effecten op het verbeteren van data correctheid maar tegelijkertijd ook negatieve effecten op de tijdigheid van de informatie. De resultaten in dit onderzoek geven aan dat een optimale hiërarchie in een netwerk afhangt van de ambities en doelen van een organisatie.

Contents

List of figures

List of tables

1 Introduction

The Dutch Ministry of Defence was founded in 1928 and is tasked with commanding and providing directions to the four main Royal Netherlands Armed Forces, being the Navy, Air Force, Army and the Marechaussee (Defensie, n.d.-b). To support the armed forces the Ministry of Defence also has three other organisations under its command. The Joint Support Command, the Defence Materiel Organisation and the Central Staff. These provide the necessary support for the main four organisations so that they can perform their societal tasks. With over 68.000 staff members, supply chain management is one of the important but difficult tasks the Ministry must deal with (Defensie, 2021a). They have to be certain that all of their personnel have the clothing, food, weapons and medicine they need and this is only a small example of the goods and supplies that are necessary.

Every branch is responsible for certain tasks within the Ministry. The Army is responsible for most tasks on land and alongside this for most types of vehicles. Besides these main branches, the two support branches, DMO (Defensie Materieel Organisatie) and DOSCO (Defensie Ondersteuningscommando) have their responsibilities within the organisation. All of these branches are highly hierarchical and there can be up to 6 layers above a high-level manager and as many below. Besides this hierarchical structure, the Ministry now also has a strong horizontal connection. Every branch now has to use services that are only provided by one of the other, e.g. air-transport, ground-transport, or maintenance of a vehicle

Societal Contribution

In July 2021 provinces in Germany, Belgium and the Netherlands were flooded due to heavy rainfall . This was a disastrous event that led to damages estimated between 350 and 600 million (Jonkman 2021). To provide aid during this event, the Dutch Military sent 300 troops, 25 transport trucks and 3 military ambulances to the region to help with evacuations, emergency dikes and the treatment of the wounded (Defensie 2021). This is only one of the tasks that the Dutch defence forces have. Besides providing domestic aid and support, they also partake in several international missions. From disaster support, anti-piracy missions in Somalia, or the enhanced Forward Presence mission where 270 Dutch troops are stationed in Rukla Lithuania (Defensie n.d.-a). This last mission is even more important now that tensions rise between NATO and Russia (NATO 2022). Therefore, communication in this organisation is so important. Information communicated has to be correct and fast allowing the Ministry to have effective operations. Information-flow effects their supply and demand which is important, without a good working supply chain the Dutch Military Forces would not be able to fulfil their responsibilities, like the disaster support in Limburg.

Scientific Contribution

Besides the societal relevance of this research, it will provide a new aspect from a scientific perspective. Most research that has been done in the field of information-flow is with focus on information diffusion behaviour. Gaps in literature were identified that will provide a new focus on timeliness, data accuracy, and employee busyness and how they are affected by formal structures inside organisations. The effects of hierarchy, type of controllers, network sizes, priority information and data correction. These aspects are not studied as of this moment. This is further established in the background section of this chapter.

Research Scope

The choice was made to focus more on one of the sub units of the Ministry of Defence. As an example case study, the Air Transport Unit or Dutch Military Air Transport Unit was selected. They are part of the DVVO which is part of DOSCO or Defensie Ondersteunings Commando (Dutch for Defense Support Command) of the Dutch Ministry of Defence. Traditionally DOSCO is more service-oriented than DMO and provides for example transport capabilities to the other organisations through a sub-organisation DVVO (Defence traffic and transport organisation or Defensie Verkeers en Vervoers Organisatie in Dutch). This sub-unit is responsible for all types of transport within the organisation, from post to personnel transport. With this comes a structure where information speed and reliability are of the essence. Material that is not received in time, due to delays in requests or problems that have not surfaced, are potential problems. This delay can be caused to a lack, or problems with the information that a department needs. Delays in orders, management information, or other types of information also troubles the decision processes within the Ministry of Defence, where processes are described as slow and almost syrupy (from the Dutch `strooperig').

1.2 Background

This sub section consists of a literature review that will explore current research on information-flow within organisations. First, the literature review method will be explained and afterwards there are several sections, each for one of the major topics considering the research scope.

Method

For this literature review several different search methods and databases were used. At the start of the process more general search methods like Google, Google Scholar, SCOPUS and the Search Engine of Leiden University Library were used.
These methods resulted often in too many papers to process. Because of this vast number of results, a systematic approach to reviewing all pieces of literature was not possible. Instead, a systematic literature review using several criteria was used. Papers were ranked in order from peer-reviewed, publication date, and finally

citations. After this, ten papers were selected based on their title. The criterium here was to check whether or not the title matched the queries. After this, the next step was to scan through the abstracts of the papers and look for relevant methods or processes that answered the question leading to the query.

The next step was to snowball from several of the more useful papers. Using their bibliographies more papers were identified. To further snowball from the originally identified papers, the ConnectedScience tool was used which creates a web of connected papers. This was only done for some papers that were deemed extra relevant. Based on their relevance to the research topic, e.g. information-flow in social organisations, or ABM information-flow modelling.

An overview of all relevant papers considered but not necessarily used as reference was kept using the Zotero reference manager.

Defining information and information-flow

This subsection will expand upon the core concept, information-flow. Before researching information-flow and what it entails it is necessary to define information.

Information is defined as "Knowledge communicated concerning some particular fact, subject, or event; that of which one is apprised or told; intelligence, news." According to the Oxford English Dictionary (Oxford English Dictionary n.d.). However, in literature information has many different definitions, some specifying it more than others. According to Checkland (1988) combining information systems and system thinking, he defines Information as usable data, inferences from data or data descriptions. Thus, defining information in relation to data. Rowley (1998), describes information using five definitions, where four are relevant for this study. Information as subjective knowledge, information as useful data, information as a resource and information as a commodity. Combining these definitions we come to the following definition for this research:

Definition*: Information is the communication of knowledge, data, or inferences to data, and can in itself be a resource or a commodity.*

Having defined information, the next step is to define information-flow inside organisations. Yazici (2002) discusses the importance of of organisational structure on information sharing systems. According to Yazici information-flow is the sharing of information between two actors. The findings of this study also indicate that there is a significant effect of organisation structure on information-flow, but that task complexity is one of the key factors. Noteworthy here is that at the time of this research, IT was still perceived as somewhat important, but not as important as traditional communication media (Yazici 2002). Pontrandolfo, and Scozzi, (2002) expand on information-flow inside production processes and add a mathematical model to describe information-flows inside organisations.

***Definition**: information-flow, is the distribution of this critical asset through a system or organisation. It is defined by the relation between two separated agents that are connected and share information based on structural and behavioural rules. (Albino, Pontrandolfo, and Scozzi 2002; Yazici 2002).*

Information-flow in organisations

This subsection will discuss the link between information-flow and organizations. In order to do this, first the difference between public and private organisations will be defined. Later, these will be linked to why good information-flow is critical to organisations.

Organisations

Public and private organisations are often put on opposite sides of one another. Portrayed as widely different organisations with different priorities and effectiveness. However, for this research, it is important to define the actual differences, and find out what makes public organizations public and what different strategies from the private world can be incorporated into the public domain.

Rainey, Backoff, and Levine (1976) discuss the differences and similarities between public and private organisations. Some of the major differences that are relevant to information-flow are: (1) more constraints on procedures, (2) less decision-making autonomy and flexibility (3) weaker authority, (4) more levels of review, greater use of formal regulations (5) greater tendency of goals to be conflicting (more trade-offs). Especially the hierarchical red tape, has been observed in public organisations (Rainey and Bozeman 2000). Rainey and Bozeman do mention that the hierarchy difference between public and private companies is mainly focused on personal and purchasing rules. Where public organisations are stricter and must follow protocol. In addition to the reduced perspective of hierarchy in public organisations introduced by Rainey and Bozeman. Hill and Lynn (2005), discuss whether the hierarchy is still present in public organizations. Through empirical research, they identify a possible trend for more horizontal governing instead of hierarchal. This however, is a trend in governance, not yet seen in the behaviour of public organisations in general.

According to Ben-Arieh and Pollatscheck (2002), there are three distinguishable types of organisations. The homogeneous organisations, semi-homogeneous organisations, and non-homogeneous organisations. Where every type has its properties on information-flow. They distinguish that every level inside the organisation, higher management etc. has its ownameters considering information processing, speed, generation of information and information correctness. They do conclude that most organisations tend to be semi-homogeneous, thus, every level in the organisation has the same production function but a different utility function. Meaning that information generation is often similar but how it is used or processed is different.

A research paper by Oliver (2008) discusses information culture in organisations. He defines several types of information management, and how information sharing in organisations affects their performance. The Dutch Ministry of Defence seems to be more like the Information Feudalism Model where individual business units manage their information. This can be seen in the difficulty of acquiring data. This is contrary to what might be expected of the MoD. One might have expected that it is more like the model Information of Monarchy, where information trickles up, and they decide whether to communicate this further.

Military organisations

Traditionally Military organisations have been very hierarchical organisations. The need for this hierarchy has been explained and argued for by the need to act quickly under high pressure situations like combat (Huntington 1981). These situations, and thus quick decisions are what make the Ministry of Defence an interesting actor. They operate under very different circumstances than most other organisations. The quick change from daily operations to missions or practices what makes the organisation special. Hierarchy is used as a tool in these situations to quickly change plans. Hierarchy is also used as a method to indicate experience and expertise in subjects (Soeters, Winslow, and Weibull 2006).

The change described by Hill and Lynn (2005) that organisations are changing was picked up on by Gøtzsche-Astrup, Brænder, and Holsting (2021). Their study goes into the details, of whether an organisation like the Ministry of Defence should change from an organisation ruled by hierarchy to one ruled by network. These network organisations define themselves by being more horizontal, flexible and even described as organic (Mintzberg 1980). The results of Gøtzsche-Astrup et al indicate that the organisation in itself is capable of the change, and that this style of management might improve employee satisfaction rates. However, this still raises the question how these changes in organisational structure will affect information-flow in organisations. As described in the next section, information-flow is affected by these new, 'network' style organisations.

Types of information-flow in organisations

Organisations such as the Dutch Ministry of Defence can be represented as a network (Durugbo, Tiwari, and Alcock 2013). Where every person or department is represented as a node and every interaction, or information transfer is a link between the nodes. A representation of a network like this can be used to study the flow of information in the current state of the network (Yazici 2002). When networks of the information-sharing pathways are created there are often two types of networks defined. Formal and Informal Networks.

Formal networks, or formalized information-flow networks are defined by Yang and Maxwell as the rules, guidelines and regulations of an organisation. These rules and guides often restrict the information-flow in a hierarchical manner (Kim and Lee 2006). Informal information-flow can then be defined as the less structured, more free forms of information sharing. An example of this was given in the new 'networking' style in the Danish military as described by Gøtzsche-Astrup, Brænder, and Holsting (2021)

The majority of research in public sector networks focuses on inter-organisational networks. When they focus on intra-organisational networks, most of them study informal dynamics of the social structure within the networks (McEvily, Soda, and Tortoriello 2014; Whetsell, Kroll, and DeHart-Davis 2020). Whetsell et al. (2020), showed in their research that formal structures have a strong effect on information search within public organisations, especially within local governments. They studied these effects in a small municipality in the US, specifically between the different departments. They call for further research that integrates formal organisation within social networking studies. In this case, information search, is relevant to this research, as it is a top-down approach for information-flow. Yang and Maxwell (2011) report on the effects of formal and informal networks within organisations and how this affects their information sharing capabilities. Formal systems are less effective compared to informal systems, the formal rules often restrict the spreading of information, and create longer travel times (Willem and Buelens 2007).

Measuring how formal an organisation is having its difficulties. When representing an organisation as a network, however, several tools are available that can help us measure hierarchy in organisations. One of these methods is the trophic incoherence parameter in networks. This parameter was first developed for representing food chains, where the trophic levels represent levels in the food chain (Johnson et al. 2014). The trophic difference of an edge is the difference in the trophic level between node X and node Y (Pilgrim, Guo, and Johnson 2020). Johnson et al then defines trophic incoherence, q, as the standard deviation of the distribution of trophic differences over all edges in the graph. Pilgrim, Guo and Johnson show in their paper how the trophic incoherence score of a network can be used to determine how hierarchic the network is. This is later confirmed and expanded by Moutsinas et al. (2021). Pilgrim, Guo and Johnson provide with this a method and scale to determine how hierarchic an organisation is based on the trophic incoherence score. Where an organisation with a q = 0, is described as a tyranny. There are no cycles and information-flows only one way, the structure is tree like. With a value of q ~ 1 is determined to be broadly hierarchical with more back-edges. They refer to this as a democracy. The final discussed value is q ~ 2, where there is little hierarchical structure, which would have an analogue of anarchy.

The importance of information-flow in organisations

In the private sector, where companies are driven by profit, information is critical for the company's growth potential (Krovi, Chandra, and Rajagopalan 2003). This is only one of the reasons why information-flow is important for an organisation. This subsection will expand on the importance of information-flow for organisations by looking at its effect on supply chains and decision making. Both are critical functions for organisations like the Dutch Ministry of Defence.

Besides these two topics, information-flow also affects organisations in more ways. Information-flow in organisations represents culture. Formal or informal organisations both have different information-flows, if only because of the hierarchical differences (Westrum 2014).

Information-flow and supply chains

An important core concept is the link between information-flow and supply-chains. Information is critical for an organisation's working and decision-making, as discussed in the next section. However, another way information-flow affects organisations is on their supply chain. In this case information can be orders that move between people or departments, but it can also represent data necessary to coordinate the supply chains. One example of this is in disaster relief supply chains. Day, Junglas, and Silva (2009) show that the lack of information-flow heavily affects organisations' disaster recovery efforts. Possible impediments in the information-flow can be unreliability of the data stream, or source identification difficulties. It affects the supply chain, by creating miscommunication between different departments, inside the same organisation, or between organisations. E.g. not aligning transport moments for disaster relief goods. In production, supply chains, information-flow is critical in informing other departments, about production results and providing decision-makers, later in the supply chain with critical information (Pedroso and Nakano 2009). This information is critical, for supply chain managers, to make sure that every single part of the supply chain is working optimally together. Delays, or problems in one part, will ripple through the supply chain if left unreported (Fiala 2005). This is also the case in military supply chains, where the stakes can be even higher if emergencies arise (de Waard, de Bock, and Beeres 2019).

Information-flow and decision-making

Decision-making within organisations can range from restructuring the organisation, to deciding what orders (products or services) are necessary to be able to perform their tasks.

Citroen (2011) shows that information is critical in strategic decision-making and that the lack or delay of information is harmful to this process. In addition, Ahituv, Igbaria, and Sella (2015) show that the lack of complete information and time pressure negatively affect decision-making. They tested this with Israeli Air Force commanders in a stressful situation. One reason that time pressure takes place is if, due to delays in the process, only a short amount of time is left for a decision. Thus, the flow of information in an organisation affects time pressure, as well as information completeness. As described in the exploratory interviews, requests, be it for information, or goods, get lost in the hierarchy. It can take months for requests to move through the organisation. The negative effects of incomplete information are also described in the work of Lurie and Swaminathan (2009). They also show that 'too' quick of an information-flow can negatively affect decision-making.

Ehsani, Makui, and Sadi Nezhad (2010) analyse the effect of decision networks using information theory. Studying dependence between subgroups in an organisational network, they study if placement of a subgroup affects the decision structure of an organisation. They find that organisational structure variants can be optimized so that the decision-making efficiency of that organisation will increase. This is mostly focussing on grouping agents that are dependent on one another. Thus, clustering information hubs together.

However, decision-making (e.g. restructuring) can also have significant effects on information-flow. Liu and Kumar (2011) show that different supply chain structures can be used to improve information-flow within the organisation and with this the process of decision-making. In terms of restructuring organisations, modelling approaches can be applied to identify problems of information-flow to improve vertical and horizontal information-flows (between actors at the same level in an organisation (Creti 2001, Eppinger 2001).

Information-flow Quality

Modelling information-flow for organisations is necessary because of the need to better understand how to: minimise the duplication of information, organise and coordinate processes and manage the sharing of intra-, inter-organisational information and reduce / eliminate redundant processes (Durugbo et al. 2013). When modelling information-flow, the definition of good information-flow, or information-flow quality is necessary. This way, it is possible to analyse and recommend on best practices based on model results.

As described in section 'Information-flow in organisations ', information-flow is critical for organisations. Information is necessary in all layers of an organisation and is vital for its processes. From order quantities and dates to management information on production quality. Information is vital for a healthy organisation. The first thing that is clear from the literature review is that there is not one view of information quality (Klischewski and Scholl 2006). Meaning that there might be shared definitions of what things to measure information quality as. Parameters like data quality, how intact the data is, whether any mistakes were made. Or the timeliness of data for its arrival. Actors in organisations will have different viewpoints of each of these, and other parameters. Which one is more important, and the value of a parameter that is desired. Already in 1988, (Seal 1988), information quality was defined with nine different dimensions: accuracy, objectivity, (cognitive) authority, reliability, relevance, precision, recall, timeliness, perceived value, and currency of information. The latter of them, being more relative. It defines the different characteristics to which we might measure information. For example, the need to receive information in real time or just at certain time intervals. These criteria can also have an adverse effect on one another, a high timeliness (e.g. when an actor is searching for information, it is able to receive this quickly) will decrease the precision and recall of that request (Ballou and Pazer n.d.; Klischewski and Scholl 2006). An empirical study by (Li and Lin 2006), found that trust between actors is a major contributor to information quality and information sharing between supply chain partners. Adding yet another dimension.

However, these are information quality dimensions, the literature review did not result in any papers that define information-flow quality. This means that from the definitions of information-flow, the importance of information-flow and the definitions of information quality, a definition of information-flow quality will be constructed. The first aspect of information-flow quality comes from the definition of information-flow, the flow of information from one actor to the next. This is an event that happens in real time, with time being the biggest constraint. Combining this with timeliness (an information quality dimension) our first information-flow quality indicator is the time it takes for a piece of information to travel from one point to its destination in an organisation, now called *timeliness*.

The second dimension of information-flow quality is a bit tougher to define for this study. As the definition of information-flow further specifies the transfer of information or data from one point to another, the data part is also critical. Information quality has many properties that can be measured. However, not all can be measured in this one study, and not all are as relevant, as their connection to organisational structure is unknown. What is known is the effect employees can have on data accuracy, mistakes that slip into the work that can add up and create big problems for an organisation (Seal 1988). This will be the second dimension of information-flow quality, *data accuracy*.

The third dimension is based on network properties of information-flow. When an information package must travel from one point to the next, bottlenecks in the organisational structure will lead to problems down the line. It will affect the timeliness of information-flow, but also the data accuracy of information-flow might be affected. As employees that are overworked are more likely to make mistakes (Jannati and Khalaf 2022). The third dimension will thus be *information-flow*, measured in average time and the lack or existing of bottlenecks.

Kim and Lee (2006) describe how social network style information sharing networks, result in better knowledge-sharing capabilities for employees. Whether this is true for information-flow quality is not yet described in literature. This begs the question if going from a hierarchical and formal organization to an open and network style organization will improve information-flow inside an organization.

Knowledge Gap

Chapter 1.2, *Information-flow in organisations*, describes the proposed effect of organisational structure on organisations. However, the effect of organisational structure on information-flow and especially information-flow quality is not yet studied. More specifically the effects of hierarchical networks on information delays within networks. In chapter 1.2, *Information-flow Quality,* Assumptions are made about the effectiveness of these organisations and that less layers will improve an organisation's effectiveness, but not how this business structure affects travel speeds of information within a network. A possibility for a model is identified here that studies the effects of different network structures or other behavioural scenarios on information speed through a (sub) organisation.

Gap 1: the effect of hierarchy on information travel speed in organisations.

Information quality has been widely documented. Information-flow quality has not. Defining information-flow quality and studying the effectiveness of certain organisational structures is something that until now has not been studied in literature.

Gap 2: what is information-flow quality? When is information-flow good?

Gap 3: the effect of organisational structure on information-flow quality.

The speed that military organisations have to react with at certain moments creates a unique situation in an organisation. When a threat or incident happens, everything in the organisation will shift to completing this goal. This influx in priority information possibly affects information-flow inside an organisation. However, effects like these have not yet been studied, especially in the context of information-flow quality.

Gap 4: the effect of an influx of priority messages on information-flow

These are the four gaps in literature identified in the literature review. These gaps in literature will be used to form the main research question in the next chapter.

Use Case:

One of the problems within the Ministry of Defence is the 'lack of inertia' associated with business processes that are in place now. They are even described as syrupy (or 'strooperig' in Dutch). In an organisation that is critically reviewing its processes, an analysis of its formal structure is necessary. The literature study has shown that a less formal or hierarchical structure might improve information-flow and business effectiveness. However, this begs the question if these 'new' business processes also would help the Ministry of Defence in battling the lack of inertia with information-flow in their processes.

Gap 5 can a more network style organisational structure influence the ministries of defence, air transport unit

If results from this study show that it might improve information-flow in an hierarchical organisation like the Ministry of Defence, lessons can be learned for other organisations. This is easier applicable to other public hierarchical organisations. However, results from the other organisational structures influence, types of employees, work speed, work accuracy will also benefit directly other organisations looking to make their processes more efficient.

1.3 Research Approach

This chapter describes the main research questions (RQ) and the sub-research questions (SQ) that are derived from the main question, and structured so that they enable answering the main research question. The chapter is divided into two sections, one describing the main research question and the second section describes the use case and methods used for each of the sub questions.

Main Research Question

The knowledge gap described in the previous section, was used in formulating the main research question. It is based on the hypothesis that information-flow performance inside organisations is affected by the hierarchical structure and that a flatter organisation will have a better information-flow quality. Organisational structures, information-flow, and public organisations, are some of the keywords derived from the identified research gap. Combining these leads to the following research question.

What are the effects of formal network structures within public organisations on their information-flow quality?

This research question will address the gap identified in the literature review by identifying by investigating the effects of organisational structure on information-flow performance. Where the organisational structure will entail network structure, employees, and employee performance.

Combining the RQ with a case study will further improve this research. It will allow us to focus more. The case study will be focussing itself on the Dutch Military Air Transport Unit. A sub-unit of the DVVO that focuses on air transport. The generally hierarchical structure of the Ministry of Defence will be used as an example of formal organisations. Where formal means the structure of the organisation, but also the stricter rulesets of who communicates with who, or the escalating a situation.

Sub-research Questions

This section will describe the sub-questions used to answer the main research question and expand on their relevance. To answer the main research question six sub-questions are used that each use a combination of different methods. Figure 1 shows the research flow diagram. This diagram combines the order, methods, and research questions into one organised process that was followed in order to answer the main research question.

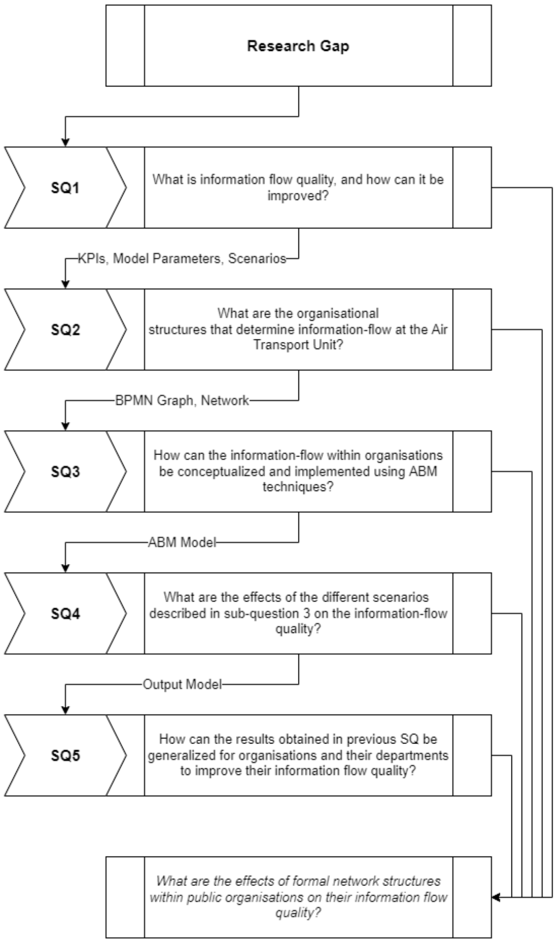


Figure 1: Research flow diagram

SQ 1: What is information-flow quality, and can it be improved?

Before we can research the effects of formal organisations on information-flow, information-flow quality must be defined. This will provide a framework for the other sub-questions, so that they can be focussed more on answering the main research question. It will be able do this, by framing the research domain within the identified definition. Using the definition for information-flow quality, KPIs can be developed that will help measure the information-flow quality in networks. The KPIs that follow should indicate the effects of organisational structures on information-flow quality. This in its turn, can then later be used to quantitatively measure the effects that formal organisational structure has on information-flow quality.

Defining information-flow quality, creating KPIs and thus enabling quantification of information-flow quality within the Ministry of Defence, will help to develop and test scenarios. These scenarios can then be used in recommendations for the Dutch Military Air Transport Unit.

SQ 2: What are the organisational structures that determine information-flow at the Air Transport Unit?

The second sub-question's function is to define the case and gather the necessary data and insights to use it to answer the main research question. Organisational structures are defined as the interaction network structure that is used to communicate, the types of employees, function that they perform, process times and if possible, error rates that affect information-flow quality at the Dutch Military Air Transport Unit. This SQ will help the RQ along by analysing and defining the case-study

SQ 3: How can the information-flow within organisations be conceptualized and implemented using ABM techniques?

In order to test the network from SQ2 and study the impact from different scenarios on the performance or quality of the information-flow inside an organisation a model has to be constructed. This model has to have the capabilities to form and analyse complex network structures and study the behaviour of information-flow inside these structures. To be able to do this, this question will provide the conceptualisation of an information-flow model that has these capabilities. Besides the ability to model the case study organisation, the model should have the ability to test randomly generated organisational structures, increasing the opportunities and insights for this research. The next step in answering this sub-question would be to test whether, a conceptual model like this can be constructed. Summarised this sub-question will provide a model conceptualisation and implementation that than can be used for experiments to do with organisational structure. Analysing how this would be implemented. It will help answer the boundaries that the main research question brings and test whether or not organisational structure might be of influence.

SQ 4: What are the effects of the different scenarios described in sub-question 1 on the information-flow quality?

Once the model is conceptualised and implemented it can be used for the next step in the model process, experimentation. This sub-question aims to answer the main research question using the model, the different scenarios as input. Thus, providing a vital contribution for this research. Testing the effects of scenarios that affect organisational structure, like hierarchy, size, type of employees, percentage priority messages.

SQ 5: How can the results obtained in previous SQ be generalized for organisations and their departments to improve their information-flow quality?

This sub-question will attempt to assess the different outcomes from the experiments and how they can be related to other organisations. Linking the results back to the case study and how they can be extrapolated to organisations that have similar structures or processes in their organisations. This will be achieved by combining a literature review and the outcomes of the different sub-questions.

2 Methods

This chapter will discuss the methods used to answer each of the research questions with the goal of answering the main research question. The theoretical approach to RQ3 is given in this chapter and the method used to implement it are described in chapter 3.

Research question 1

To define information-flow quality, an extensive literature review will be performed. This literature review will seek to define information, information-flow and then information-flow quality. This will provide an option to quantify and measure information-flow quality. Finding and defining KPIs to find methods to improve information-flow quality. The necessary data and knowledge for the scenarios will follow from the extensive literature review (section: background).

Research question 2

This sub-question will be answered using a review of internal documents and some exploratory conversations within the DVVO's Air Transport Unit. Internal documents like procedure descriptions, organograms, summative pieces, organisational descriptors are used to form a picture as detailed as possible of the organisation.

In the end, this should result in a description of the Air Transport Unit system. Details that are not found in the internal documents or interviews will be approximated in using literature or based on assumptions. This should finally result in a BMPN graph that describes the information-flow process accompanied with some descriptive statistics. This BMPN graph is then transformed into a network format that will be used as input for the model created as answer to research question 3.

Research question 3

Implementing an ABM model follows a certain process as described by K. H. Dam, I. Nikolic, and Z. Lukszo (2013). The modelling process is shown in Figure 2. It goes over the steps conceptualisation, specification, experimentation, and validation. What is important is the fact that it is a continues process. A model is never complete and can always be tweaked based on the validation results. Therefore, it is important to make clear what the specifications of the model need to be. The conceptualisation of the model will be based on the results of sub-question 1 and sub-question 2 combining information from both questions.

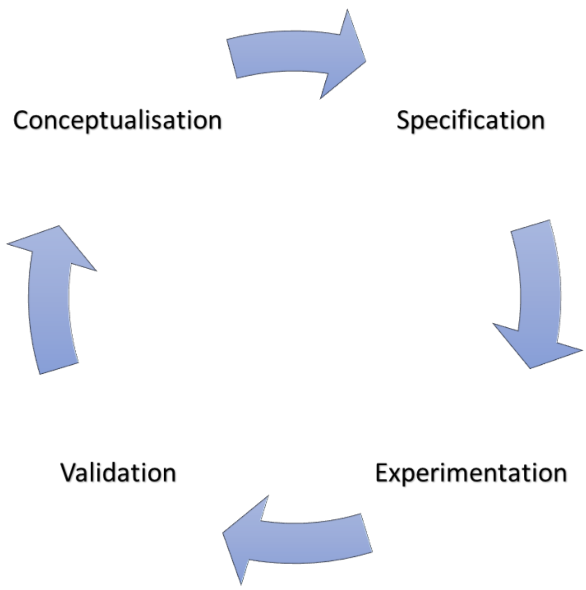


Figure 2: Modelling Cycle

The implementation of the model will happen in two parts. First a network model will be created as basis. The analysis of the Dutch Military Air Transport Unit and the resulting system description from sub-question 2 will thus firstly be modelled using Network Analysis methods. In Chapter 3 the steps and procedures will be expanded upon. Network Analysis will help this RQ by forming a model of the information-flow system that will provide the option for a preliminary analysis of the system using descriptive network statistics. This network model will also allow for the generation of random networks. Allowing the further study of the effects of organisational structure on information-flow. These theoretical networks will allow for easy differentiation in networks and to test a wide range of different structures, which would be limited if there was only a focus on the network from the case study as this is a small network.

The second part of the implementation asks for a method that allows for dynamicity and the interaction between the actors identified in SQ 2. The method that was deemed most useful for this is agent-based modelling (ABM). Strengths of ABM that make it the most suitable are for example the ability to: (i) capture emergent phenomena; (ii) dynamic, and (iii) the ability to describe systems bottom-up instead of top down (Crooks and Heppenstall 2012). As described by Figure 2, information-flow is a dynamic situation where interaction between actors is key to the information-flow definition. Thus when aiming to study these interactions, and the emergent behaviour that follows from them, ABM is perfectly suited to study actor interactions (K. H. Dam et al. 2013).

Verification of the model will happen during and after the model development. Methods for verification will exist of extensive code walk through, minimal model testing and tracking agent behaviour during these processes. After verification, and gathering the results of the experiments, validation of the model will be necessary. Dam et al list several options for model validation, historic replay, face validation, model replication and literate validation (K. H. Dam et al. 2013). To validate this model, a hybrid between historic and data validation will be used. The case study studied in this research has provided a range in processing times that if modelled correctly should have the same pattern as the output process times distribution.

Research question 4

The experimental design used for this research question will allow for scenario testing under various circumstances. The implementation and experimental design will be expanded upon in Chapter 3.2. The results of the conducted experiments will be analysed using different Python packages that will allow for visualisation of the results.

Research question 5

Research question 5 will be answered using the results following from the other sub-questions. Using randomised networks as a base for the experiments allows for greater generalizability of the results to other departments or organizations.

3 Model

This chapter will form the answer to sub research question three. Starting with a section that describes the process of conceptualising a model that can help answer the research questions. Section 3.2 describes the implementation of the conceptualised model, defining the implemented behaviour of agents and how the model works. The final section 3.3, describes the verification of the model.

3.1 Conceptualisation

This chapter describes the conceptualisation of a model that emulates information-flow in organisations. Especially information-flow in hierarchical organisations, and how this hierarchical structure affects the information-flow quality as described in Chapter Information-flow Quality. This chapter will partially answer SQ3. The conceptualised model and its workings will form the basis for the implementation of this model, as described in the next subsection, which will answer the rest of SQ3. A foundation for the model conceptualisation comes from the course Advanced Simulation, program EPA at Delft University of Technology. During this course freight transport over a road network is modelled and simulated. Wondering if this would also work for information over a network the idea was adapted and changed to fit people and information. The concept of traveling over a network from agent to agent that was used in the course, originated from a master thesis by Arga Jafino (2017)

Model Environment

Time and space are the two main concepts used to define the model environment. Implementing time in the model will grant the ability to track the time it takes for information to flow from one actor to another actor in the model. As information may take minutes, hours, or days to be processed, the passing of time in the model should allow for this. The time in the model is represented as steps, and since information may take minutes to be processed and send on their way, every step will represent a minute in real life.

The space of the model is of a more abstract level. As defined in Section Defining information and information-flow, information-flow is when information goes from one actor to the next. To bring this into the context of the model, there are two options, have actors meet one another and pass information along, making them move in 'real' space. The second option is to let actors stay put, and let information move from one to the other. As the goal of this study is to analyse the flow of information in an organisation due to organisational structure, the second option was chosen. A network representation of an organisation will be used to model the space of the simulation, allowing for information to flow from one point to the next.

Agent types, and behaviour

The goal of this research is to model information-flow in hierarchical organisations. Taking the formal organisational structures into consideration the agents in the ABM model will have to represent an organisation as closely as possible without creating a model that is too complex.
In order to achieve this, the decision was made to create two main classes of agent: *employee* and *information package*. The *employee* class will represent four different types of employees that can be generalised in an organisation. The *information package* class contains three different types of information package. Both main classes and their subclasses are shown in the UML inspired diagram shown in Figure 3. A larger version of that diagram is shown in Appendix C

Employee Class

The *employee* class exists of six different employee types. Each of these employees represents a role of functions in an organisation. The types of employee that will be necessary are based on the BPMN graph from Chapter 4, Figure **7**. In this section arguments for each subclass of employee will be made as well as what is required of each agent type. Every *employee* type will be assigned a place in the model space based on their degree properties. In the employee subsections, the degree requirements will be explained for each type.

Every employee class will need a basic functionality that every sub-class (employee type) will inherit. These tasks will have to do with general information and data processing. Based on the literature review, research gap and research formulation, requirements for the employee class have been listed. An employee needs to: (1) *receive an information package*. An employee will need the capability to receive information packages from other employees. This is critical when taking in mind the definition of information-flow. (2) *process that information package.* When an employee receives information, that employee will process that information with a goal in mind. This process will take a given amount of time before the employee is able to send their completed work to the next employee. (3) *modify the data*. During the process something might go wrong with the data, and some errors might slip into the information package. (4) *send information package onwards*. When the information package is processed and all tasks of the employee are completed, the employee will need to be able to send the information package to the next employee so that step 1 can be repeated. As added complexity to the model priority and regular information packages have a need to be processed. *Employees* will need to have the ability to process one before the other.

Gap 2 and 3 in the literature require the modification of the data in information packages. This will thus need to be a property that all employees can use. In practice data or information can have many forms. From purchase orders, request orders (like in the case study), data tables, letters, reports etc.. Every single one of these types have a multitude of problems or mistakes that can happen. From small spelling mistakes that cause problems in databases, to mistakes in dates on orders or requests. However, representing the information package in a form like this would be unnecessarily complex. This would add more text and data than would be necessary and increase the computational complexity of the model. Instead, an abstract form of data representation will be necessary that can still be modified but is not as complex as reports, letters, or orders.

Besides functions like sending, receiving, and processing, every *employee* will need to have a set of parameters that allow it to work at certain speeds and with a certain precision. These parameters should have the possibility to be randomly assigned from a range specified by the model user.

Autonomous

The *autonomous* class is based on employees in the BPMN diagram that have full autonomy, that is, if they finish their work, they themselves can decide whether to send on the information package to the next employee. In the diagram in Figure **7** this is for example represented as the first Employee MOVCON. They receive data

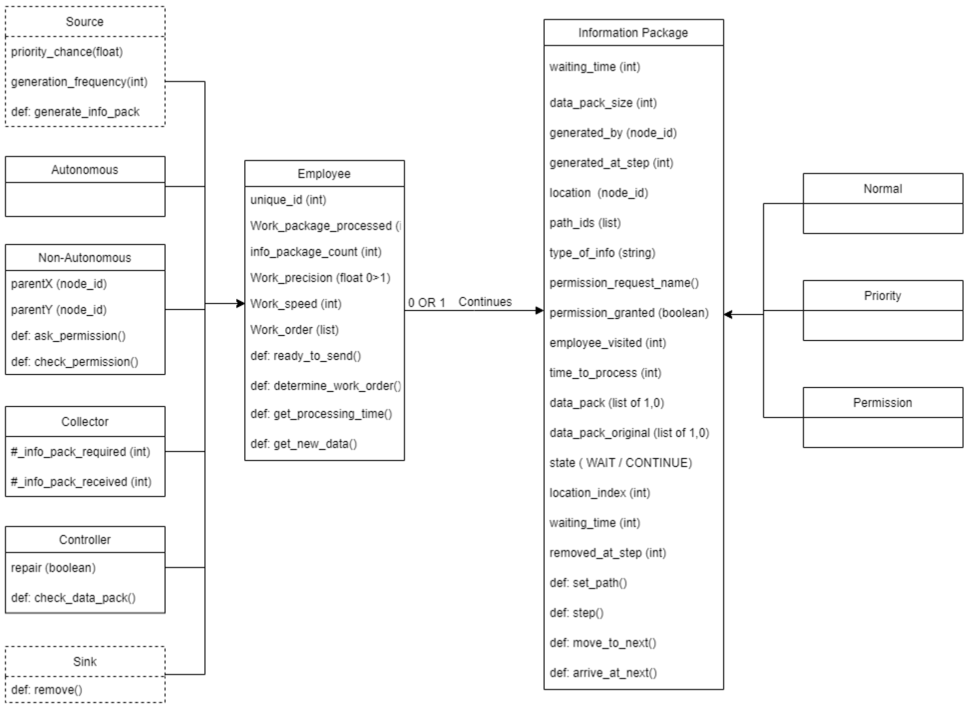


Figure 3: UML inspired diagram of model classes. Larger version can be found in appendix C

and send it onwards when they have finished their process. It is assumed that this employee type can be representative for many employees in various organisations and is associated with more informal or network style organisations.

The degree requirement of the autonomous class is $e_{in} = 1$ and $e_{out} >= 1$, since it does not need approval only one input edge is necessary. The *autonomous* class will not need any additional functionality on top of the inherited *employee* class functionality, since the basic functionalities already allow for receiving, processing, and sending information.

Non-Autonomous

The *non-autonomous* class is the opposite to the *autonomous* class, as the naming of the class might suggest. Instead of sending when ready the class represents employees who will receive information, process that information and then only send information when they receive permission from another employee. In hierarchical organisations a lot of red tape, or signatures are often necessary before someone can complete their task or send a request. Especially in the case of requests, or orders, often a supervisor, or someone higher in the tree will need to sign off before orders are send. An example in the network diagram comes in the form of order approval. Before the order information is continued and send into the network, approval of that purchase will be necessary. Example of this is described in chapter 4, case study.

The degree requirements for the *non-autonomous* class are very straight forward: $e_{in} = 2$ and $e_{out} => 1$ as the *non-autonomous* will require an input for permission and another for information input. *The non-autonomous* employee will need the functionality to ask for permission and check whether that permission has been received. As the model is an abstraction of an organisation and the hierarchy comes not from rank, but network structure (expanded upon in section model space) a different approach to approval structure will be necessary. Figure 4 shows how the permission process of the non-autonomous employee should work.

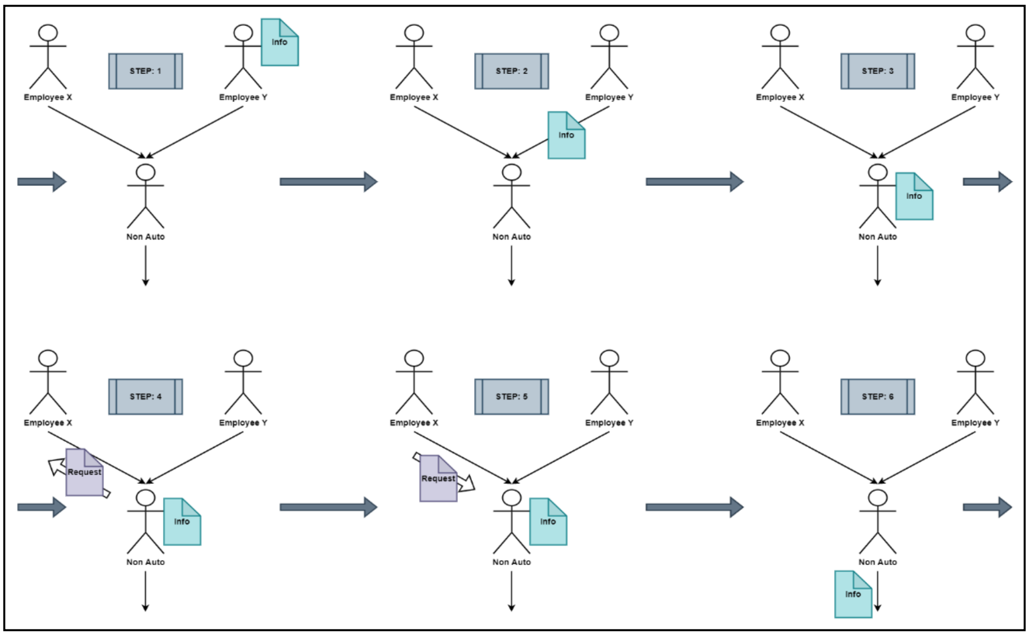


Figure 4: Non-Autonomous Actor

Step 1, Employee Y has finished processing an information package. Step 2, Employee Y sends an information package to the non-autonomous employee. Step 3, non-autonomous employee receives the information package and processes it. Step 4, the non-autonomous employee sends a permission request package to the other employee (employee X in this case). Step 5, employee X processes the permission in its work order and sends it back to the non-autonomous employee. Step 6, the employee receives permission and sends the information package onwards.

Collector

The *collector* class fulfils the role of information or data collector. In organisations there often is a step or person who collects all the necessary information, and if everything is collected, sends it to the next employee. This behaviour is also shown in the diagram in Figure **7**. There, the PN2 Planned has steps where they need multiple pieces of information before they can continue that step. Hypothesised is that this can create backlogs in information-flow. In order to test this, the *collector* agent was added to the model conceptualisation.

The degree requirements for the *collector* employee are: e_{in} => *3* and e_{out} => *1*. Information is thus collected from minimum 3 employees. For the collector to have the functionality necessary for the model, it will need one additional input parameter. This parameter, required packages, is picked randomly from a range specified by the user. This will allow for multiple scenarios down the line. As additional functionality the *collector* class will need to check whether every information package is present, and if everything is present, send the x number of packages that were collected onwards. Leaving any remaining packages in the work order.

Controller

The *controller* class fulfils an important function. It will process information sent to it and check whether or not it is still intact. This agent type represent people and tasks in organisations that make sure information is correct. Figure **7**, this step is shown in the lane of the Employee Order Management Cell. Where employees will check if data is correct, improve the data if they notice it, and then send it to the next employee.

The degree requirements for the *collector* employee are: $e_{in} = 1$ and $e_{out} => 1$. This overlaps the *autonomous* employee agent. The degree requirements of other agents are too specific to share with the controller in an abstract version of an organisation like this. The *controller* needs the ability to check the current state of data with the original state of data, collect information on this, and repair some or all of it based on a chance parameter specified beforehand.

Source and Sink

In addition to the four different employee type, two different helper employee classes will be implemented, *source* and *sink*. These two classes will help in the creation and removal of information packages in the model space.

The *source* class will be responsible for the creation of information packages, it will have input parameters to decide the number of packages generated each step and the division between *regular* and *priority*. The degree properties will be $e_{in} = 0$ and $e_{out} => 1$. The source will generate information packages and assign destinations (sinks) randomly from the network. Thus creating many information packages that will flow through the network. Generating a path from one point to the next. This path will be generated using a shortest path method as this will work well with a network structure.

The *sink* class will be responsible for the removal of information packages. It will be the destination of information packages in the network. The degree properties will be $e_{in} >= 1$ and $e_{out} = 0$.

Information Class

The information class exists out of one parent class, *information package* and three subclasses (*normal, priority, permission*). The parent class contains all the functionalities that information packages will need in the model. The sub-classes will be used to inform employees what type they are dealing with. The variables and functions that will be required are shown in Figure 3. As described and argued for in Chapter 4.1, information packages will flow from employee to employee in the network.

In order for the information package to flow from destination to destination, it will need to keep track of its position in the network and have a path to follow. The information package needs to know when it can move to the next location and how long it should wait (be processed) at a location. All these parameters will allow for the movement of the information package through the network. The information package will store data that can be processed and modified by employees. Figure 3 shows that the *information package* class does not contain many functions but will need many variables in order to store data for later analysis.

Network Generation

Since the space of the model will be a network representation, methods to incorporate this into the model need to be conceptualised, defined, and explained. To answer the main research question, as described in the research approach that there will be a case study and a theoretical analysis of randomly generated organisational structures that will allow for a wider study that can be generalised to more organisations. The next two subsections will explain what will be necessary for random networks and networks from a source.

Theoretical study random networks

In Chapter 1, hierarchical networks were defined as networks where there are few edges or, interactions between actors. Organisational structure can represent a tree structure when it is very hierarchical without any edges (interactions) between successor nodes. When an organisation is less hierarchical nodes will interact with one another on the same level. Forming a more horizontal organisation. The random networks will have to have the following properties in order to suffice for this research.

- A certain number of input and output nodes will be necessary, labelled Source and Sink in the section (Agent types, and behaviour).
- Every other agent type will need to be represented in the networks generated.
- The network will need to be weakly connected. Meaning that every node and part of the network must be connected by at least an edge in a single direction. In order to form an abstraction of a real-life organisation. This will make sure that there are no separate unconnected branches in an organisation, that if they are not connected have nothing to do with the organisation.
- A method will be necessary to check how hierarchical the network is. This way networks that are more or less hierarchical can be generated in order to test the effect of this organisational structure on information-flow quality.

- The method should allow for the number of nodes (employees) to be specified. This will allow for testing larger and smaller organisations/departments that differ from the case study sample.

When the network has these properties, it should make a for an appropriate model space for the simulation. Allowing all the agents to show their behaviour.

Network from source

Generating the network from source must allow some of the same characteristics as the random network generation. However, instead of random assignment of employee types, predetermined placing will be necessary. The assigned types of agents will be according to the same ruleset.

Model KPIs

Chapter 2 described the model KPIs as the data quality, timeliness and network flow. To measure these KPIs the information package class, employee class, and model class will need to gather information.

For the timeliness the step that information was created, as well as when it was removed will be necessary statistics. This will provide insight in the duration of the path that the information package took. The data quality should be measured with the original and final state of the data. A comparison can then be made on the differences between them. Resulting in an error percentage. This can then be averaged to gain a more model wide insight. For the last KPI, network flow, the model will need to collect employee workloads (*work_order*), and collect what nodes are the busiest. This will provide insight in possible bottlenecks that can harm the organisation. With the case study network, this can be an actual node, with the random study, information about the type of employee agent will provide insights into the effects of the different scenarios on busyness, and how it affects employee types.

3.2 Implementation

This subsection will describe the implementation of the conceptual model as described in the previous subsection: 3.1 Conceptualisation. Starting with model implementation, which gives a brief overview of the methods used to implement the model and the thought process behind some of the design choices during implementation. The model input subsection describes what input parameters are needed for the model to run, what their effects are and how they should be interpreted. The theoretical study network generation section describes how the random networks were generated and what checks or additions were used to achieve the required network statistics. The subsection Agent implementation describes how each of the agents is designed to work together to achieve the behaviour described in the previous section. Finally in the section Experimental setup the scenarios used in the experiments as well as how they were conducted are described. This subsection also describes the output of the model that is necessary to analyse the results of the model.

Model implementation

As described in chapter 2 in the subsection Research question 3, a modelling cycle was used to guide the model implementation. Input for the model implementation comes in the form of the model conceptualisation. The model itself was developed using Python's MESA package along with several support packages that all can be found in the requirements.txt file on the GitHub page. Generating the random networks and adapting them to the conceptualized requirements was possible using the NetworkX package. The model is a Python based program that is run from the terminal. The readme file describes what file contains what code. .The model, the requirements file, readme file and the scenario data used in this thesis is published on GitHub. (https://github.com/LJanssens97/MSc-Thesis-Information-flow-in-formal-organisations.git).

Instead of developing an interactive model, a model that is computationally efficient and easy to use was the goal. Keeping in mind that easy adaptability for repurposing should be priority. Therefore, the model was constructed over several files and classes and functions that allow for quick additions of agents or change in behaviour. With the focus on the adaptability of the model, the decision was made to not make the model interactive or visualise the process while the model is running. Options for debugging are included that enable tracking every agent's behaviour in the model, allowing for real time monitoring of the model.

Since the model is not interactive, there is also no need for model interface. The model is run using the *model_run.py* file which serves as the 'interface' of the model. This file sets the parameters and experiments of the model. Specifying the run length of a model run; input data; random or data-based networks and how many repetitions of each scenario are needed.

Model Input

Since there is no interactive model interface, model parameters must be set manually. However, a central approach was implemented where the model parameters are stored in a file that is used by the model to set its variables. Removing most variables from having to be changed in code.

When studying a network that is predefined, a network file has to be presented to the model. The model expects an edge list in csv format as input. This file should describe the Source and Target of an edge. Additionally, a file is required that specifies which employee or node is what type of agent. This approach was chosen as it allows for more streamlined empirical research where the researcher can specify what agent types are at what position inside an organization's structure. Modelling theoretical organisational networks requires input data that specifies the boundaries for the generated network. Their size, trophic incoherence, percentage of *controllers* and the probability value necessary for the random network generation function. This enables the modeller to generate networks that abstractly represent an organisation of the size and hierarchy specified by their respective research.

Other parameters in the parameter input file set boundary values for the agents of the model. Both theoretical-models and data-models need these values set for the model to function correctly. The parameters set values for the minimum and maximum time for the work speed of an employee; minimum and maximum values of the number of information packages required by a collector and minimum and maximum values for an employee's work precision. Besides these agent inputs, the input file requires a specification of the time interval in between information package generation and what percentage of information packages is classified as a priority package.

Network Generation.

The network for the case study network was created using Gephi (Bastian, Heymann, and Jacomy 2009). This is an opensource tool that allows for the visualisation and creation of network graphs. Using this tool, the edge file and visualisation in Figure 8 were made. This edge list, as well as data that specifies which node is what type of agent is than processed by a function in the file *networks.py* this file generates both theoretical and data networks and returns them to the model.

Since theoretical random networks are the main source of data for this thesis, it is required that the networks to get as close as possible to the conceptualised standards. The parameters *nodes* and *probability* are used as input for the NetworkX function *fast_gnp_random_graph()* (Hagber, Schult, and Swart 2008). This functions returns a $G_{n,p}$ random graph or binomial graph that is then modified to fit the other parameters. NetworkX provides many ways to generate random networks but none of these functions' outputs fitted the requirements from the conceptualisation. Some functions generated only tree structures, thus creating similar hierarchical networks every time, did not allow for directionality, did not allow for the usage of a seed as input or took too long to generate a random graph. Therefore, the choice was made to adapt the generated network to fit to the other parameters in four steps.

- Step 1: Connect Loose nodes.
- Step 2: Connect Loose components.

- Step 3: Connect Source nodes.
- Step 4: Connect Sink nodes.

Since the function returns a network with loose nodes and loose components these must be connected for the network to be weakly connected. This achieved by first selecting one of the loose nodes and connecting it to another random node in Step 1. In Step 2, a list of loose components is generated and a random node from one of the components is connected to a random node from one of the other components creating one weakly connected graph. As it is now, the random network would exist of around 60% input and output nodes, which does not representative of an organisation. When thinking of organisations, one can image many input and output points for information but 60%, not being employees would be an overestimation. To compensate for this, in Step 3, nodes that would be labelled as *source* are connected to other source nodes. Step 4 follows similar logic and connects sink nodes to other random sink nodes until the threshold value is reached. This value has to be set in code in the *Networks.py* file.

The current stage of the network does not yet consider the hierarchy of the network. This, however, is achieved by considering the networks tropic incoherence. To ensure that a model meets a requested value, the trophic incoherence is compared to this requested value., and a new model is generated if it does not meet the threshold of (±0.1). If it passes this statement, the network is verified to be weakly connected and if there is a path from one of the sources to at least one of the sinks. Both control points were added after the model verification since they caused errors during trial model runs.

Agent implementation

The implementation of agents is characterized by the two main classed that were conceptualized. The UML diagram shown in Figure 3, shows the two main classes, their functions, and sub-classes. This section will discuss the implementation for each agent.

Employee Class

The employee class is responsible for storing and processing variables that all the employees share and provides functions such as getting the *time_to_process* for an information package at its location, keeping the work order, keeping track, and generating the new 'data', and keeping track of model statistics necessary for the experiments. The autonomous class is missing from this list as this is an empty sub-class and solely uses employee class functions. The work order is first in first out, except for priority messages. They are added to the front of the work order so that they can be processed before other messages.

<u>Source & Sink</u>:

The source employee generates information packages for the network. It does this at an interval specified in the scenarios file. At every interval a chance is set for the information package generated to be a priority package. This chance is also given in the scenarios file.

The sink agent houses the remove function, which removes agents (information packages) form the model schedule and from the model space. Before a package is removed, statistics about the package are stored in the model for later processing.

Non-Autonomous:

When an information package arrives at a non-autonomous agent (X) it calls their *ask_permission()* function. This function generates a permission request that is sent to the other employee above them (Y). It sets the path of the permission request and assigns a processing time for it. If that employee Y is a non-autonomous employee itself, a permission request is not generated for the permission request. The assumption was made that if you request permission with the person above you, they have the authority to give the okay. When the permission arrives back at employee X it is removed from the model. Before it is removed but added to the work order of Y. Every step the information package will check if the permission request has been granted by checking if their request has appeared in the work order of the employee X.

Collector:

The collector class has two functions. The function *everything_collected()* and *in_collection()*. The first function is called every step to check if the threshold of information packages that are required for this employee is reached. When this is reached, information packages at this location will call the *in_collection()* function that returns which information packages may continue to the next employee. The waiting time, or processing time, at the collector is modelled by making packages wait until everything is received. The employee does not process the information packages itself.

Controller:

The controller class has one main function. Reviewing and possibly correcting the data in an information package. The controller can access the original data that an information package has, and the current state of data. It will then compare the two and identify all the mistakes, for each mistake found, there is a set chance that the mistake is corrected by the employee. In the code itself it is also possible to specify the time a mistake found by the controller will add to the waiting time of an information package at this employee.

Information package Class

The information package class has three sub-classes that represent the types of information packages in the model. Each of them is empty and is used to identify the type in the model.

The information package class caries most of the models' functions. The class has three main functions, *step()*, *move_to_next()* and *arrive_at_next()*, all of which enable information packages to move through the network. The step function is a MESA function that is triggered by the model function at every step. The step function describes what must be checked by information packages when they are at what location. Checking if they may move to the next employee, and if not removing a step from their waiting time. When an information package may move to the next employee, their state is set to continue, and they are assigned new waiting times, their data gets processed by the next employee and they move to that location in the network using the *arrive_at_next()* function.

Experimental setup

The experiments in this thesis are all focused on answering the main research question through the sub-questions. Studying the effects of organisational structure on information-flow quality. Every model run has a set of parameters that are set for each agent. In the case of this study the employee agents and information-package agents have certain parameters that do not change in every scenario.

Table 2: Stable model parameters

Rather the focus is on changing the network style and the number of priority requests that go through an organisation. These values are shown in Table 2. Each scenario has these values set for their agents. The values are boundary values that define a range that is used to do a uniformly random selection of a value for an agent. The greyed out *processingtime* variable shown in Table 2are artefacts from an earlier version of the model. The possibility to use this is still there, but since the goal of the theoretical networks is to compare them to the case study as well, the choice was made to base the distribution of the case study. The work precision is based on data from the Niagara Institute. When employees are working with

Table 1: Stable parameters over the experiments

Agent	Min. Processingtime	Max. processingtime	Min. workprecission	Max. workprecission	Min. workspeed	Max. workspeed	Min. #infopacks	Max. #infopacks
Info-regular	60	240	-	-	-	-	-	-
Info-priority	60	240	-	-	-	-	-	-
Info-permission	15	35	-	-	-	-	-	-
Empl-auto	-	-	0.85	0.95	0.8	1.2	-	-
Empl-nonA	-	-	0.85	0.95	0.8	1.2	-	-
Empl-Coll	-	-	0.85	0.95	0.8	1.2	3	7
Empl-Cont	-	-	0.85	0.95	0.8	1.2	-	-

following common practices 10 to 30 human errors per 100 opportunities happen (Niagara Institute n.d.). When working very precise between 5 and 10. Resulting in the assumption that a chance of making mistakes ranges from 5-15%. In addition to this the case study showed that there are employees at the ministry of defence that must manually enter data from one information system into another information system increasing the opportunity to make mistakes.

The *workspeed* values are based on the processing times from the case study as well. Taking into consideration that an information package will take the time it was assigned to with a *workspeed* of 1, we introduce variance with the employees by increasing the range from 0.8 to 1.2 where the actual work speed assigned to an employee is picked with a normalised random distribution.

Collectors need a minimum of information packs and a maximum of information packs assigned to them. These are also kept constant over all the experimental scenarios. The minimum number of information packages for the theoretical networks is set to three, as per design collectors have a minimum in-degree of three. A maximum of seven was chosen since degrees higher than seven rarely occurred in the network test phase. These values were also used for the case study experiments to allow for comparison of results.

3.3. Verification

During and after developing the model, the behaviour and outcomes of the model were constantly analysed. The verification of the model consisted of extensive code walkthrough, tracking agent behaviour, and testing a minimal model and comparing the results. For the latter, the case study was used. This is a predefined network where information on real life processing times is known. The description of this process is more in depth explained in appendix D. After verification the model seemed to work as intended. Generating results that are to be expected based on the data known from the case study.

4 Case Study Description

In addition to the theoretical study of theoretical randomly generated networks a case study of the Air Transport Unit will be added to the experiments later. To be able to do this the ministry of Defence, the DVVO, and the system surrounding it will be described. This should enable creating a BPMN graph that is used to create an organisational network representation. This results in a more abstract representation that will serve as input for the model.

The Ministry of Defence

As mentioned in chapter 1, the Dutch Ministry of Defence exists out of six major branches. The Royal Navy (koninklijke marine), Royal Military (Koninklijke Landmacht), Royal Airforce (Koninklijke Luchtmacht), Defence Support Command (Defensie Ondersteunignscommando, DOSCO), Defence Material Organisation (Defensie Materiaal Organisatie, DMO) and the Royal Marechaussee (Koninklijke Marechaussee). Of these six branches two of these branches are support branches, being DOSCO and DMO.

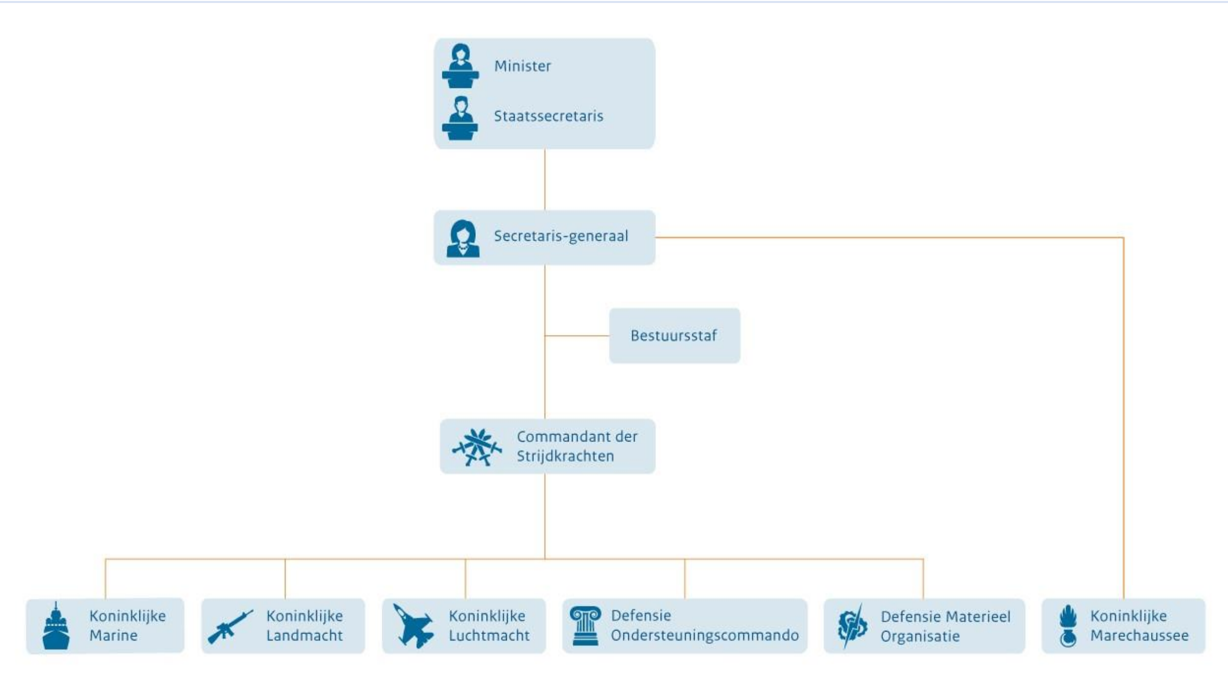


Figure 5: Top structure of the Ministry of Defence. The separate branches and how they are connected.

These branches each have their own tasks and responsibilities to the Dutch Public. Besides their separate tasks each of the branches has tasks to fulfil for the other branches. For a long time, every branch prioritized itself mostly, argued for their own importance and prioritized their own. Before the creation of the two support branches, DOSCO and DMO, in 1996, every branch took care of their own logistics. They made sure that they had the supplies that they needed, provide facilities that they needed, and more. Now, after 1996 this is taken care of by DOSCO and DMO. The phenomenon described above is according to the MOD a phenomenon that has been in decline (Defensie n.d.-b). The new structures introduced within the Defence system tried to break the barriers between these branches and succeeded partially. During conversations with employees at the MOD, people mentioned that they still notice the tension between departments and that one helps their own branch first.

One of the defining factors and that separates the Ministry of Defence from many other organisations is their task to respond to emergencies and crises, where they must drop everything, they are currently doing to deal with this. Taking the hypothetical example of a mission that abruptly ends. In this case everything is put to work to make sure that the boots on the ground can safely return home. A similar hypothetical situation is when there is a threat to one of the tasks that the Ministry of Defence is tasked with. If the situation requires it, everything will be dropped to handle this priority event.

Air Transport Unit

One of the sub-units of DOSCO is the division 'Division Facilities, Logistics and Security (Divisie Facilitair, Logistiek en Beveiliging), this division is responsible, as the name suggests, for a wide variety of services that they provide for the entire organisation. These services are all ordered within several sub-units, each belonging to their own branch of this division.

One of the sub-units is the Defence Traffick and Transport organisation (Defensie Verkeers- en Vervoersorganisatie, DVVO). Within the MOD this unit is responsible for all forms of transport when materials, goods, or personal are considered, but also postal services and the organisations car park. This sub-unit exists of around 380 employees (DVVO n.d.). The main task is to plan, and perform (if inhouse) transport by truck, ship, plane or train. Expanding on the hierarchical structure, the sub-unit itself has six departments with each their own teams. The structure of the DVVO can be seen in Figure 6. The department that holds the team that will be the focus of the case study is the NMCC (National Movement Coordination Centre). The NMCC, and with it the Air Transport Unit is responsible for the worldwide preparation, planning and delivery of traffic and transport services to aid the Dutch armed forces in their tasks.

The Air Transport Unit (ATU) plans all types of transport, goods or personal that use an airplane as the mode of transportation. It is a complicated process where around twelve people work. The process gets its complexity due to the time-pressure, shifting timelines, expenses and legal or bureaucratic difficulties. In the several planning steps, these difficulties can arise. Difficulties in the process like permission from sources outside the ATU or even the DVVO are necessary.

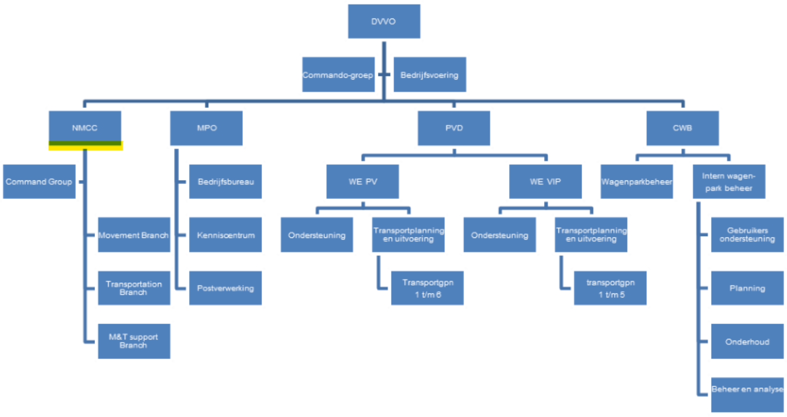


Figure 6: DVVO and its substructure. Yellow marking indicates the branch/location of the Air Transport Unit.

The process is described in a BPMN inspired diagram in Figure 7. The process starts with an authorized requester, this can be a commander of a unit or another authorised staff member of one of the four main armed forces branches. This authorised requester uses a so-called ATB to form a request. When the request enters the system, an employee of the order management cell will pick up on it. Their tasks are to check whether the order is correct, contains all necessary information, and what type of transport might be necessary. If the order is correct, it will be processed into SAP and get a BA code. If an order needs multiple forms of transport, instead of going directly to the PN2 planner, it will go to a project leader who will oversee the project. In the BPMN graph the project leader would be in-between the PN2 planner and the Employee Order Management Cell. What types of transport will be necessary, who will plan what, and then send the tasks to the specific planners.

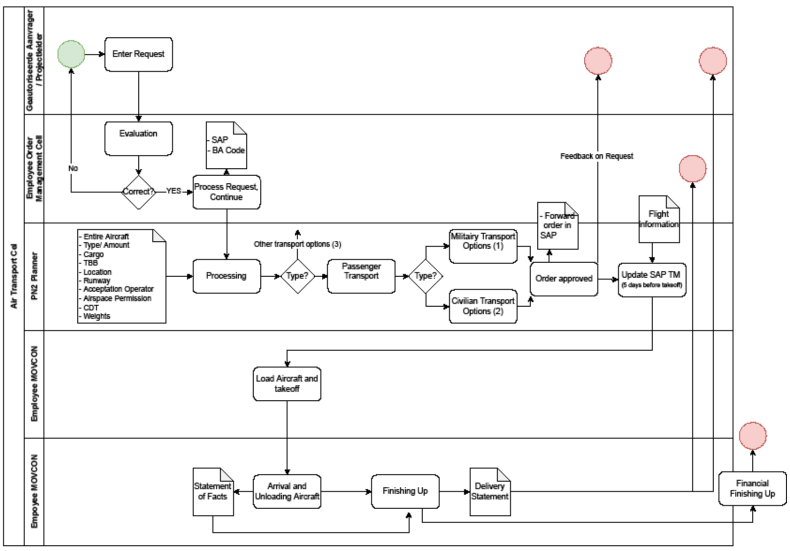


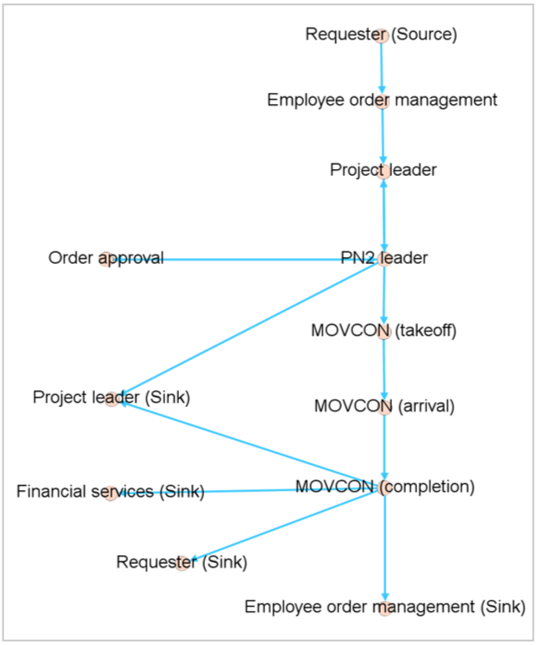
Figure 7: BMPN inspired diagram of the Air Transport order process. See Appendix B for a larger figure.

The PN2 planner requires that every part of information is present. Shown in the file style box in the diagram. Based on the type of cargo, the number of people that need to be transported and/or they will require to bring military supplies, different transport options for airplane travel are required. Each of the options has its own caveats due to complexities in international law. Besides this, the different systems each have their own information sharing system (MEAT, SAP, HAW, and more). These different types of information sharing systems and their interactions with MOD systems, can create difficulties for the employees that work at the ATU. The systems are not integrated, and information must be processed manually to different systems. The ATU will always aim to use transportation options provided and operated by one of the MOD departments. If this is not possible, chartering a plane, or seats, with a private company is the next option. Costs of chartering a private airplane, like a Boeing 737-300 can be achieved from $17,350 an hour (Paramount Business Jets n.d.). The MOD has as a practicing rule with the ATU that when a purchase, like a charter, costs €30.000 or more will need to be approved. This means that the ATU planner will be required to draw up an ATB that needs to be sent to Order Approval. In practice, the chartering of an airplane, or seats happens often. Which means that the approval process happens more than once.

After the PN2 planner, the process continues, the PN2 planner communicates to the project leader that the order has been processed and forwards the information to an employee MOVCON. Multiple MOVCON employees work at multiple steps in the process, making sure the transport takes off, arrives, checking if everything is in order when it arrives and finishing up the order by entering everything in the involved systems. The final MOVCON employee has the task of communicating the delivery statement to the employee order management cell, the project leader, the financial services of the MOD, and the requestor.

This process, and the BPMN graph provided the information that was necessary to construct a network diagram that can serve as a model input. Communication lines are visible in the BPMN graph and are combined with structures that were mentioned during conversations with employees. Take for example the order approval process, this is not present in the BPMN graph but is a clear example of hierarchy and takes away autonomy from the employee. The diagram of this network, including the table that links employees to agent types is shown in Figure 8.

The requester in BPMN graph will act as a *source* as they provide information. Financial services, Project leader, Requester, Employee order management all act as *sink* in the network. The employees Project leader, Employee order management cell are represented twice in the network, as besides providing a *sink* function, they also act as *collectors* and *autonomous* agents. Taking into consideration the fact that the PN2 planner has to request permission for an order when costs are above €30.0000, the PN2 planner is labelled as a *non-autonomous* *agent*. As the function of the project leader in the order process, to communicate information, collect it, and send it to the right planners, the project leader was labelled as a *Collector* agent.



Employee	Agent Type
Requester (Source)	Source
Employee order management	Controller
Project leader	Collector
PN2 leader	Non-auto
MOVCON (take-off)	Auto
MOVCON (arrival)	Auto
MOVCON (completion)	Auto
Financial services (Sink)	Sink
Requester (Sink)	Sink
Employee order management (Sink)	Sink
Project leader (Sink)	Sink
Order approval	Auto

Figure 8: Case study graph based on BPMN diagram. And table that explains what agent type is at what position.

This process takes time. The duration from when an order arrives until it is approved and sent to MOVCON can be between 60 minutes and 600 minutes. This is however not a normal distribution. Transport type, cargo type or the date, all influence how long the process takes. Most are processed within several hours; some take up to the 600 minutes and in rare cases can take up weeks or months. Since this is a wide range, and contains

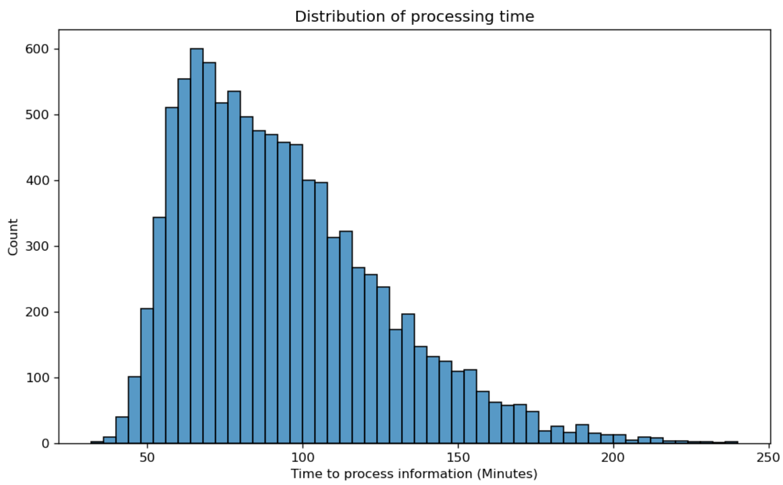


Figure 9: Distribution of processing time information within the Air Transport Unit. Approximation based on timeliness estimation.

multiple steps, the size of an information package, and thus how long an employee might work on it is linked to a right tail distribution where each information packet is between 30 minutes and 240 minutes of standard speed time, approximating the data from conversations with employees at ATU. This distribution is shown in Figure 9.

Conclusion

The Ministry of Defence is a complex, wide, but also hierarchical organisation. It is complex in its tasks, ranging from defending the country from foreign invaders to providing support to its citizens all over the world when emergencies take place. In these scenarios daily tasks are dropped, and priority is given to the task at hand. The organisation is wide it its links with outside organisations, contracts for transport ranging from truck, train to ship, or chartering planes. All the other different services that are all 'inhouse' make it a wide organisation as well. From maintenance to production, to catering, to music, everything is done inhouse. The organisation is hierarchical in nature, describing the place of the ATU in the organisational structure the number of levels above it rose quickly. Only including organisational structures, not the employees, and the hierarchy inside the other bodies that are above the ATU or are behind the requester process. Unfortunately, it was not possible to model the processes superseding the ATU, or the different branches that communicate with the ATU. Information collection proved difficult as employees are structurally in shortage of time to finish their own tasks, let alone aid in research by answering questions. Nevertheless, the information that was gathered provides a case study that can be used in the model to test whether behaviour seen in the theoretical models is visible in networks within the Ministry of Defence.

5 Results

This chapter depicts the results of the different scenarios tested on the model. To accommodate for limited computing power, the experiments with corresponding scenarios were split up to run on two different machines. The first machine is an Intel® Core™ i5-10600k based machine with 32GB of DDR4 3200MHz RAM. The second machine is an AMD Ryzen™ 5 3600 based machine with 16GB of DDR4 3600MHz RAM. Every scenario was modelled 10 times to reduce randomness in the results.

5.1 Case study results

This subsection describes the results of the case study. It uses scenarios to study the effects of the percentage of priority messages in the network and how this affects the information-flow quality in an organisation. First, the statistics of the network will be discussed and expand later on the results of the different scenarios. The network statistics of the case study do not differ per scenario since the network itself does not change. The density value indicates the relationships present in the current state compared to all possible relationships. As the density score is low, few of the possible connections are present. The relevant network statistics are shown in Table 3.

Now for the KPI data correctness. The differences in percentage of the data string that is incorrect for every information package when it arrives at its destination are small. The mean percentages at the scenarios: scenario 0: 8.62%, scenario 1: 8.59% and scenario 2: 8.62% are small. Scenarios 0 and 2 share similar error percentages among them. The error percentage of scenario 1 is 0.03% smaller than the other scenarios. However, the mean value is more meaningful as a result when considering the distribution of the error percentages among the scenarios. This distribution is shown in Figure 11, showing the effects of the different scenarios. Again, the differences between the scenarios that can be observed are minimal. What can be observed is the difference between error percentages found in the distribution. The statistics show that the majority of information packages that have arrived and their destination have an error percentage of 10%.

Table 3: Network statistics of the case study.

statistic	value
Num. nodes	12
Density	0.11
Trophic incoherence	1.78
Avg. Shortest path	1.48

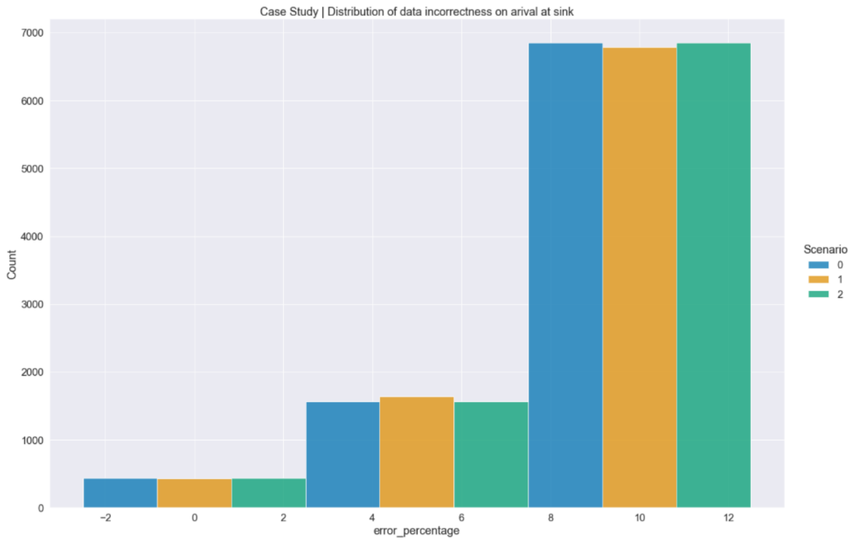


Figure 11: Distribution of error percentages over scenarios. Plot shows error percentages of 0, 5 and 10 percent.

The next KPI of information-flow quality is travel time, or time spent in the network by information packages. This is plotted using the normalised travel time, time divided by employees visited, and the mean travel time. The data is shown in Figure 10. The figure on the left shows the mean travel time/step (normalised) for each of the scenarios, wheras the right figure, shows the traveltime.

Comparing the left and the right figure, it can be observed that the behaviour between scenarios does not change when looking at information travel speeds in total normalised travel times. The traveltime distribution of the case study is shown in appendix D in the verification section as a means to verify the model. However, the mean travel time and distribution of traveltime show their behaviour at the end of the model, and how its behaves for every information package. It does not show how it changes over time due to the behaviour of the agents. To visualise this behaviour Figure 13 (upper and bottom) were plotted. These figures show how travel time changes over the course of simulated work month. Both figures show high similarities. Besides the mean travel time having different values, the regression lines in both figures show simmilar behaviour. It

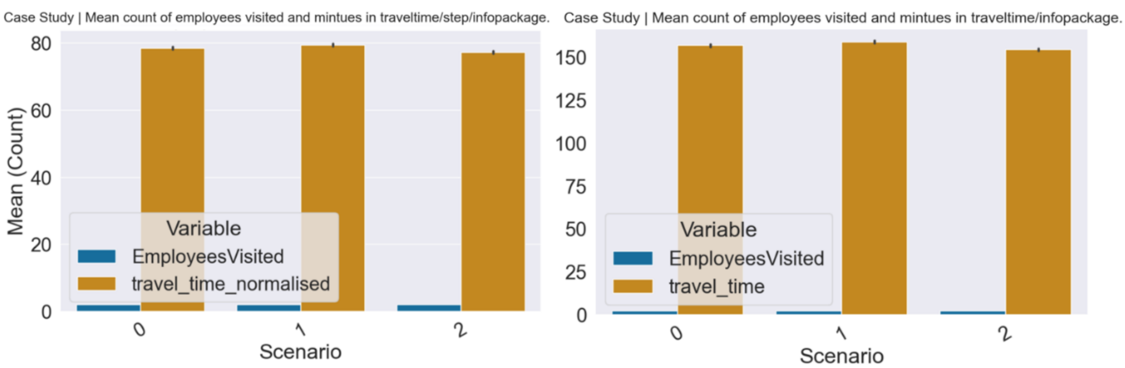


Figure 10: Case study | mean travel time

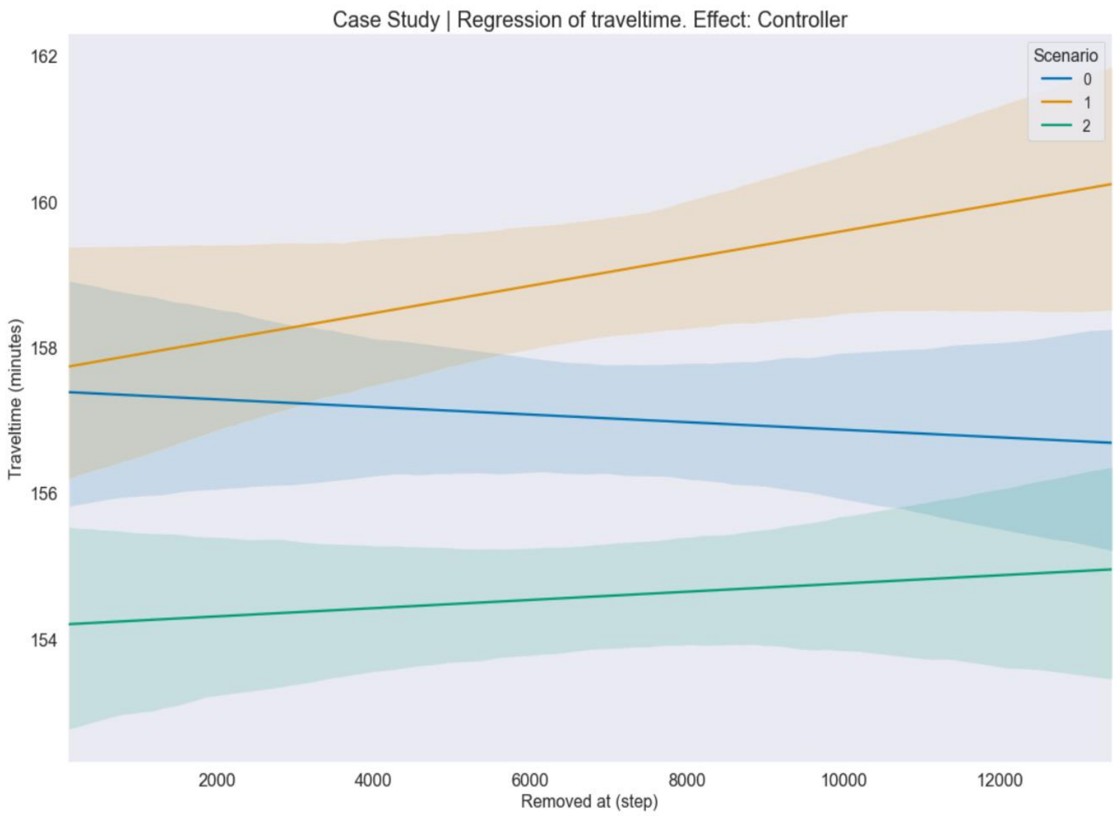


Figure 13: Case study | Travel time regression

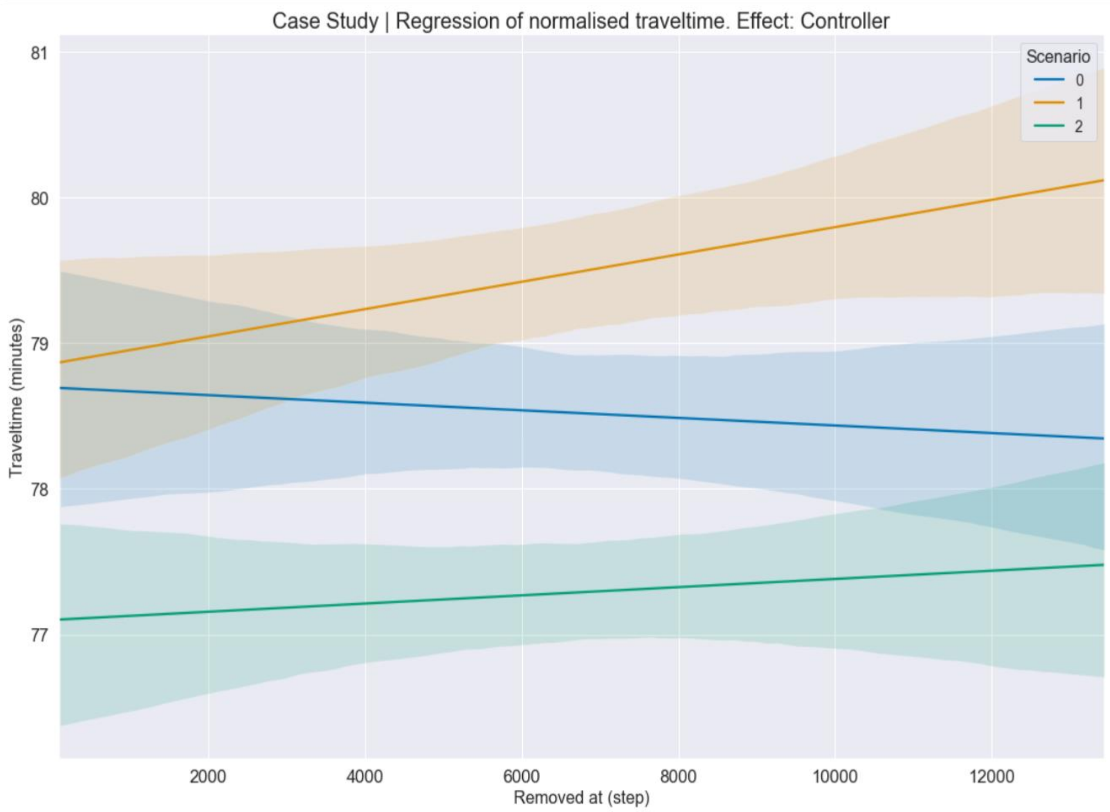


Figure 13: Case study | Normalised travel time regression

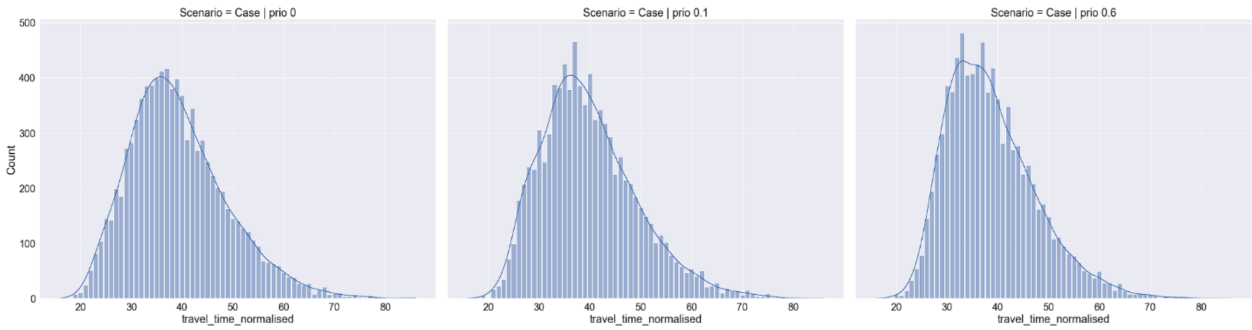


Figure 14: Distribution of travel times of case scenarios

can be observed that for scenario 0, the trend seems to be that over time the mean travel time decreases wheras for scenario 1 and 2 the mean travel time increases. The confidence interval in the plot is high for the different scenarios. However, taking this into account, it can still be observed that all three of the scenarios show different behaviour over both figures.

Table 4 shows the busyness statistics for the case study scenarios. In scenario 1 and 3, the mean busyness of all the employees is higher after 4 work weeks than compared to 1 work week of time. The standard deviation shows that the mean value is close to the median. The table also shows that scenario 2 and 3 have a lower mean value for step 480 than the results of an organisation without any priority messages.

Table 4: Mean busyness of employees in the network.

Scenario	1	2	3
480 std	1.60	1.44	1.68
480 mean	5.65	5.42	5.55
13439 mean	6.01	5.42	6.04
13439 std	1.82	1.44	1.83

Table 5 shows the mean busyness, or workload for each employee type in the model. Auto and Non-auto employees both have the same busyness over all three scenarios. The NaN values at the collector row indicate that at the final step of the model, the collector had no information packages left in its work order list.

Table 5: Mean busyness for each employee type at step 13439

Scenario	1	2	3
Auto	7.10	6.25	7.20
Non Auto	7.10	6.25	7.20
Collector	NaN	NaN	NaN
Controller	3.89	4.11	3.69

Summary

This subsection describes the results obtained from the case study experiments. As shown, the effects of priority messages in the case study network do not differ much. What this means for the study and network will be discussed in the next chapter.

5.2 Theoretical study results

This section will discuss the results of the experiments on the theoretical study, similar to the case study results, every scenario has been run 10 times to account for randomness between the iterations. In the processing of the results an artefact in the random travel time data was identified. The method, and the artefact itself, are explained in Appendix H.

Scenario description

To be able to answer the research questions, 30 scenarios were implemented and tested on the model. A table with these scenarios and the corresponding values is shown in Table 7. The table shows several blocks of rows that are colour coded different. Each block helps identify a different set of network sizes with ease. The goal of the experiments is shown in the following table. Studying the effects of the left column on all the measurements in the right column.

Table 6: Describing the goal of the scenarios and experiments

Effect of	On:
Hierarchy	Employee busyness
Network size	Travel Time
Priority	Data Correctness
Number of controllers	Controller behaviour

To study the effects of hierarchy, we opted for the values for trophic incoherence (q): 0.6, 1.0 and 2.0. These values were chosen carefully based on the results from the paper by Pilgrim et al. (2020). They show that a q value of 0.4 would be the structure of the US government in a simplified diagram. They theorize however that adding these links would increase the tropic incoherence of the network. The choice for q = .6 is also due to technical limitations. As described in Appendix E, network generation probability, the network generation algorithm only minimally goes to a q of .6 that is consistent for all network sizes tested. Since this is close to the value of the US government, when the network would be expanded, .6 was chosen as the most hierarchical value. The value of q=1.0 was chosen since their research shows that this would represent a democratic organisation, where people have a say in matters and are heard. The value of q=2.0 depressants organisations where no structure is present. Thus, allowing for a hierarchical, flat, and chaotic organisation to be generated.

The second parameter tested in the scenarios is network size. Three different network sizes are tested. A small, medium-sized, and large organisation or organisational branch are tested. The sizes are based on the sizes of the biggest group of organisations in the Netherlands, the small-medium companies (MKB). These are private companies and not public companies but provide a reference in size that make the results generalizable to other organisations. Besides this, the size of larger units or branches inside the MOD can rapidly increase to these numbers as well.

The third variable in the scenarios is priority chance, which is the percentage of information packages generated that is labelled as a priority message. As explained in the case study, when something is a priority, other tasks are put on hold for a while until the crisis is handled. Three different values were included. 0%, 10% and 60% with the reasoning to have a base line, an organisation where there are constantly small priorities and an organisation experiencing a crisis generating a large influx in priority messages.

Table 7: Scenarios and the network parameters that accompany them

.	Scenarios new names	nodes	tropic incoherence	controller chance	infopack gen.	priority chance	probability
1	N25 Hierarchy	25	0.6	0.15	15	0	0.005
2	N25 Hierarchy | prio 0.1	25	0.6	0.15	15	0.1	0.005
3	N25 Hierarchy | prio 0.6	25	0.6	0.15	15	0.6	0.005
4	N25 Democracy	25	1	0.15	15	0	0.005
5	N25 Democracy | prio 0.1	25	1	0.15	15	0.1	0.005
6	N25 Democracy | prio 0.6	25	1	0.15	15	0.6	0.005
7	N25 Anarchy	25	2	0.15	15	0	0.005
8	N25 Anarchy | prio 0.1	25	2	0.15	15	0.1	0.005
9	N25 Anarchy | prio 0.6	25	2	0.15	15	0.6	0.005
10	N150 Hierarchy	150	0.6	0.15	15	0	0.005
11	N150 Hierarchy | prio 0.1	150	0.6	0.15	15	0.1	0.005
12	N150 Hierarchy | prio 0.6	150	0.6	0.15	15	0.6	0.005
13	N150 Democracy	150	1	0.15	15	0	0.005
14	N150 Democracy | prio 0.1	150	1	0.15	15	0.1	0.005
15	N150 Democracy | prio 0.6	150	1	0.15	15	0.6	0.005
16	N150 Anarchy	150	2	0.15	15	0	0.005
17	N150 Anarchy | prio 0.1	150	2	0.15	15	0.1	0.005
18	N150 Anarchy | prio 0.6	150	2	0.15	15	0.6	0.005
19	N250 Hierarchy	250	0.6	0.15	15	0	0.001
20	N250 Hierarchy | prio 0.6	250	0.6	0.15	15	0.6	0.001
21	N250 Democracy	250	1	0.15	15	0	0.001
22	N250 Democracy | prio 0.6	250	1	0.15	15	0.6	0.001
23	N250 Anarchy	250	2	0.15	15	0	0.001
24	N250 Anarchy | prio 0.6	250	2	0.15	15	0.6	0.001
25	N25 Hierarchy | Cont. 0	25	0.6	0	15	0	0.005
26	N25 Hierarchy | Cont. 0.15	25	0.6	0.15	15	0	0.005
27	N25 Hierarchy | Cont. 0.30	25	0.6	0.3	15	0	0.005
28	N150 Hierarchy | Cont. 0	150	0.6	0	15	0	0.005
29	N150 Hierarchy | Cont. 0.15	150	0.6	0.15	15	0	0.005
30	N150 Hierarchy | Cont. 0.30	150	0.6	0.3	15	0	0.005

Since data quality and timeliness are both KPIs that could be affected by the number of controllers in an organisation, the choice was made to include this in the scenarios. Scenarios 25 till 30 model the effect of controllers by changing the percentage of otherwise autonomous agents to be controllers. Percentages of 0, 15 and 30 were chosen. These are based on assumptions that in the abstract representation of an organisation, this model needs enough autonomous employees to perform actual work with information.

For the baseline scenario, scenario 10 was chosen. This scenario is in the middle of MKB organisations or larger branches within the Military in terms of size, it has the same trophic incoherence as assumed with the MOD, no priority messages as a baseline and 15% controllers. The latter, since no organisation has no control points.

Effect of hierarchy

The first effect studied is the effect of hierarchy. The scenarios analysed to study the effects of hierarchy are the scenarios N25 Hierarchy, N25 Democracy, N25 Anarchy for N = 25 and similar scenarios for N=150 and N=250. Three network sizes were tested for the effect of hierarchy to determine if the effect of hierarchy is different between different node networks.

KPI - Data Quality

The effect of hierarchy on data quality is visualised using the boxplots below. The median number of mistakes and repairs by controllers, the mean percentage of data incorrectness for each of the scenarios, as well as distribution of error percentages, are analysed. Figure 15 shows the controller data. The number of mistakes increases with every scenario that increases the value for the trophic incoherence. A peak in the number of mistakes can be seen with N25 Anarchy. Where the number of mistakes encountered by a controller agent in the model is double of that of an agent controller in scenario N25 Democracy.

The next statistic for measuring how Data Quality is affected by hierarchy is shown in Figure 16. The boxplot shows the different scenarios and the median error percentage values. The plot shows that there is an increasing trend in error percentages when a network has a higher tropic incoherence. However, as can be observed, the differences between the scenarios are small. The appendix G, effect of hierarchy, shows the distributions for all of the scenarios used in this analysis. The scenarios for networks of N = 250 were not plotted in the distribution as their median values show that they are more or less the same. The figures show that, both size networks show a similar pattern in the distribution. However, the N = 25 plot has a lesser range, only up to 35% of the data string is corrupted at arrival. For

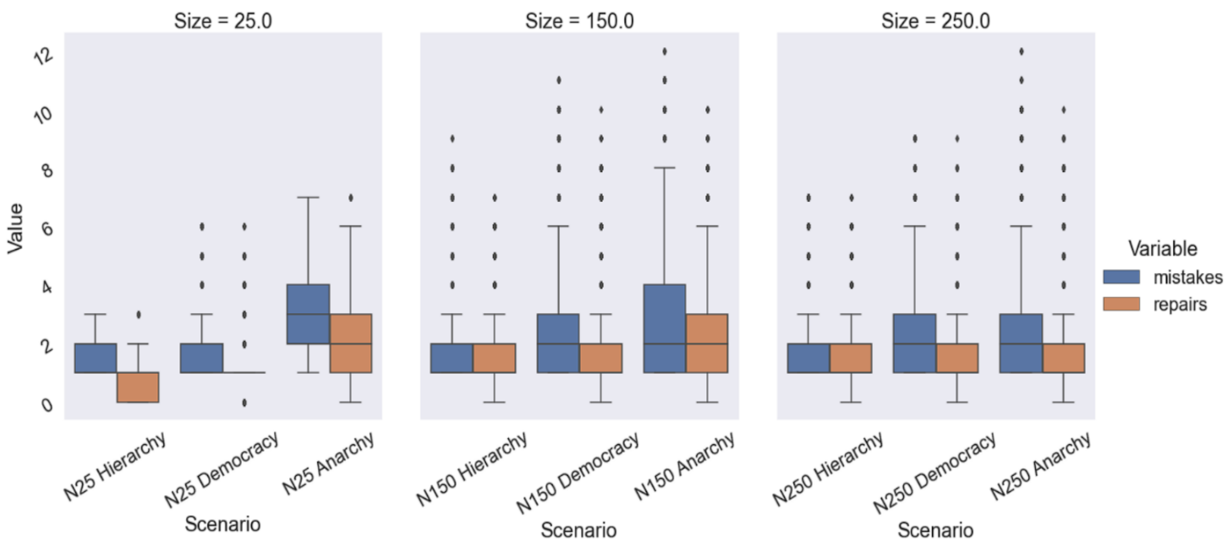


Figure 15: Mean number of mistakes encountered by controllers.

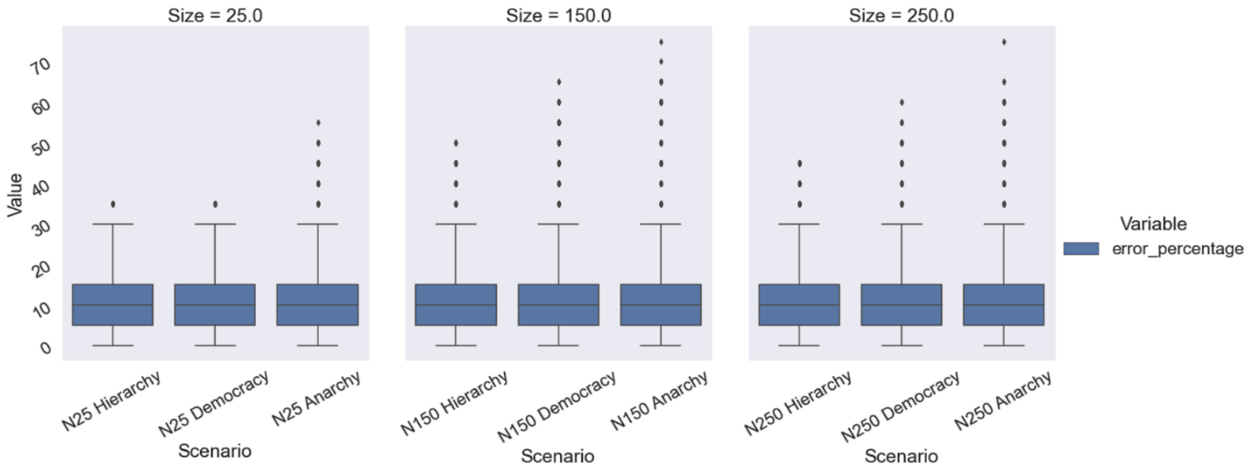


Figure 16: Mean percentage of incorrect data. The effect of hierarchy

the N = 150 plot, the maximum error percentage encountered is 50%. Scenarios 1, 4, 7 and scenarios 13 and 16 have their peak in the distribution at 10% data corruption. What also can be observed is that the maximum number of mistakes and error percentages increase when going from hierarchy to anarchy. As can be seen in Appendix G: effects of hierarchy. The distributions of the scenarios are very different. The outliers seen in Figures 15 and 16 are visible in these figures as well. Statistically all scenarios are significantly different, as is shown in Figure 59 and Figure 60 in appendix I, except for Scenario N25 Hierarchy and N25 Democracy.

Hierarchy effects on travel time

Figure 17 shows the travel time and how it is affected by hierarchy. The choice was made to plot normalised travel time. Since this enables comparing all scenarios on travel time alone. The normalised travel time shows that for a network where N = 25, there is no real trend, having a peak travel time/step at democracy. The figure also shows that for N = 150 and N = 250 that there is a downwards trend in normalised travel time when the value of q increases. This can be seen in both the median and 3rd quarters. It can also be observed that for N = 250 the highest outlier decreases when going from hierarchy to anarchy. This behaviour is confirmed by the statistical analysis in Appendix I and the distributions in Appendix G.

Figure 18 shows the effect of hierarchy on travel time, and how this evolves over the course of time in the model. Normalised travel time shows an increasing time/step over the span of the model, except for scenario 1. Figure 18 shows that scenario 10 has the highest travel time per step followed by scenario 13. The confidence intervals for scenarios 1 and 4 are large compared to the other scenarios, adding uncertainty to their results.

Also, it can be observed that scenario 7 results in the lowest travel time per step and in total. However, as can be seen in the travel time figure, scenarios 1 and 7 start converging at the final model step.

The highest travel times differ between both. Per step, the highest travel time is with scenario 10, increasing from 72 to 73 minutes. When looking at total travel times, the scenario that takes the longest time, is scenario 16. It starts around 187 minutes and ends around 200 at the end of the model run.

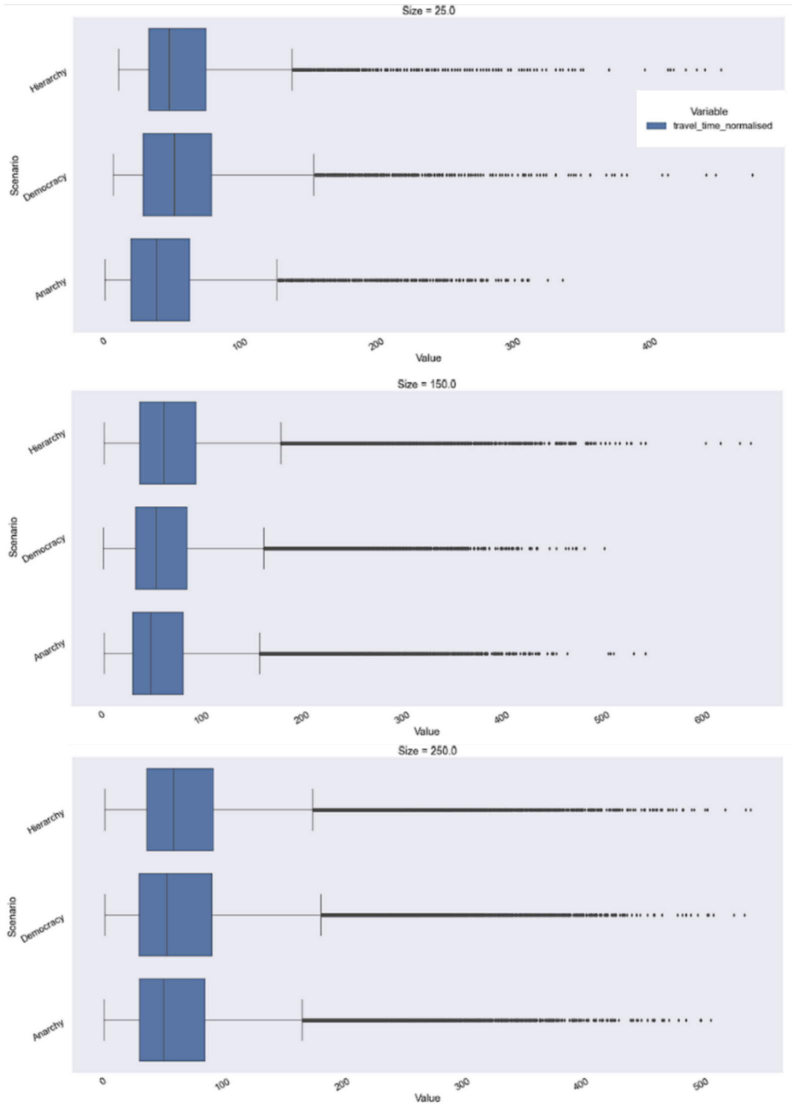


Figure 17: Mean travel time affected by hierarchy. (left) normalised travel time, (right) travel time

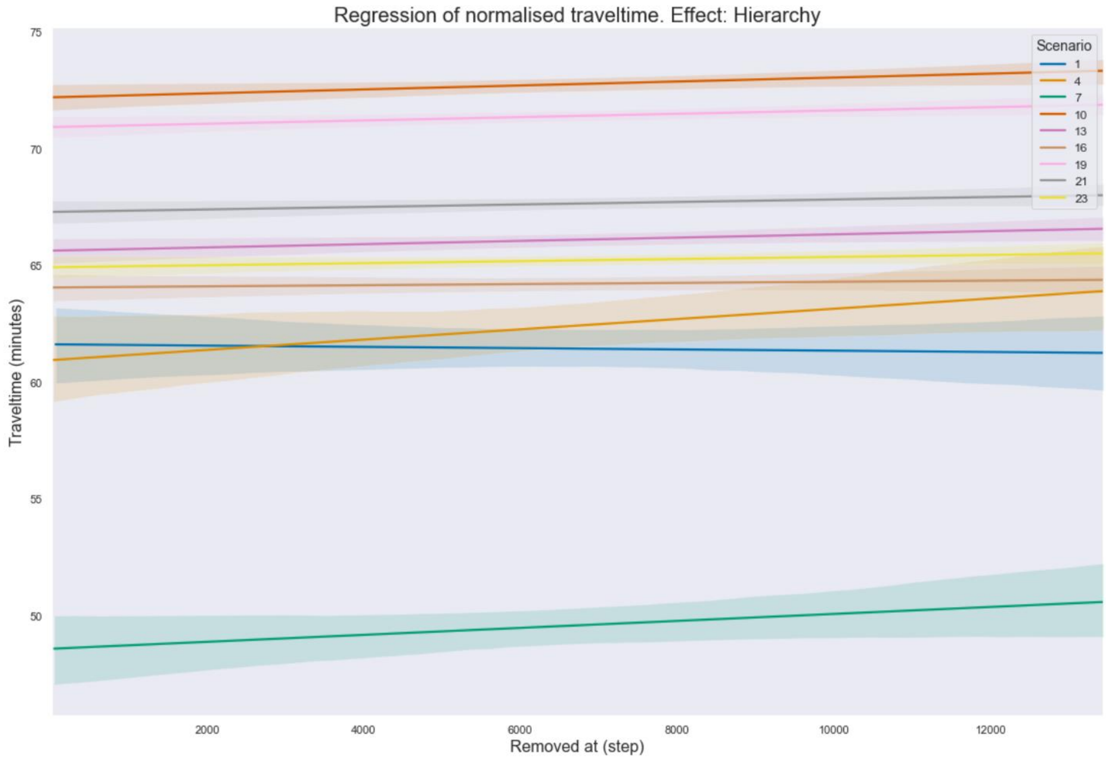


Figure 18: Effect of hierarchy on normalised travel time

Busyness

Figure 19 shows how the busyness of every employee in the model at step 13439, or after 4 workweeks. For the network size N=25 the behaviour of the different scenarios does not differ much. Scenario N25 Hierarchy and N25 Democracy are very similar to each other. As is also indicated by the KS table in Appendix I . The N=150 and N=250 scenarios both show similar behaviour where the busyness of employees increases when the q of an organisational structure increases.

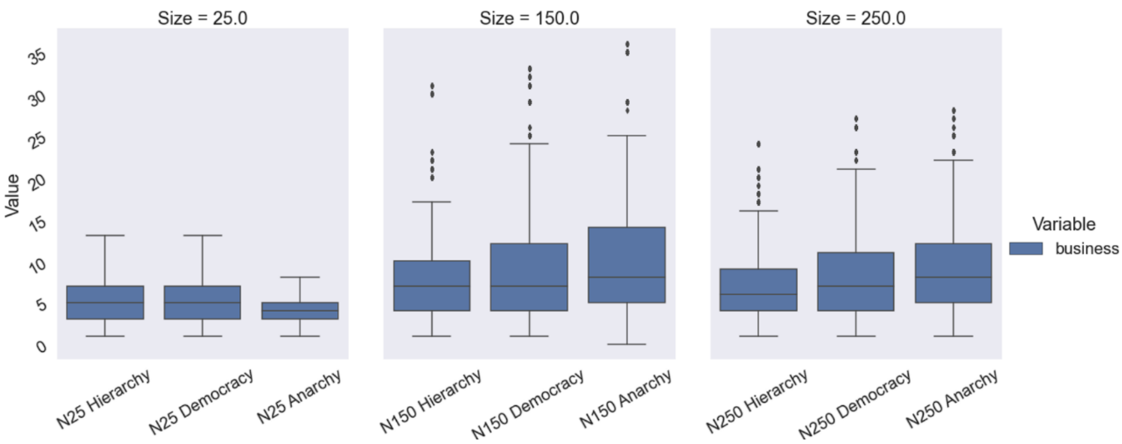


Figure 19: Effect of hierarchy on busyness

Especially in the extremities of busyness values an increase can be observed. The median values do differ, but not by much.

Regular vs priority messages

Additional comparisons were made to study the effects of hierarchy on travel times among regular versus priority messages. To achieve this, different scenarios were plotted as the previously plotted scenarios did not contain any priority messages. Figure 20 shows the results of these scenarios. Columns indicate the percentage of messages created that results in priority messages and the row shows travel time and normalised travel time results.

All four barplots show different behaviours among the scenarios. However, for this section the focus is to see if the behaviour between priority and regular messages changes when hierarchy values change.

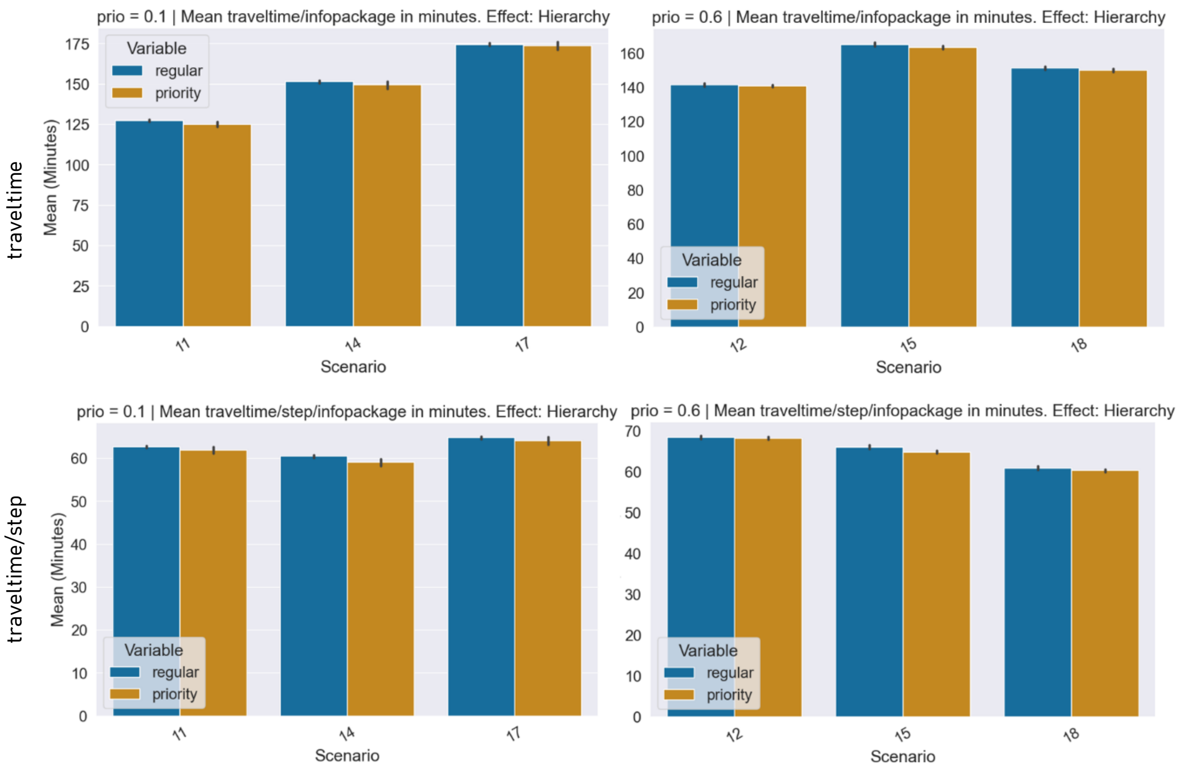


Figure 20: Effect of hierarchy on regular or priority messages.

Effect of network size

The first effect studied is the effect of hierarchy. The scenarios analysed to study the effects of hierarchy are the scenarios 1, 10 and 19, respectively N = 25, N = 150 and N = 250. The effects of network size are visualised and presented based on the three model KPIs. Since the base scenario is 10, the choice was made to visualise and discuss only the scenarios with similar values for the other parameters.

Data quality

As can be observed in Figure 21 (left) the median error percentages for every information package for each of the scenarios do not differ. Nor do the outer quarters. The outliers of the scenarios do differ. With the outlier of the N150 Hierarchy scenario being higher than the N250 Hierarchy value. The differences between the distributions are however significant, as can be observed in the tables in Appendix I, Figure 42. The outliers that show the difference between 50% and 45% is not a large difference, but one, nonetheless.

Figure 21 (right) shows that the controllers for each of the scenarios have comparable results with the median error percentages in the data. The medium network's controllers encounter slightly more mistakes on average than their colleagues in larger or smaller organisations. The distribution of these results is shown in Appendix G: Controller, where can be seen that scenario N25 Hierarhcy's controllers do not encounter more than 2 mistakes in data. Whereas N = 150 controllers encountered data with 9 mistakes present. The differences between the distributions are significant according to the KS tables shown in the Appendix. With P values lower than 0.5.

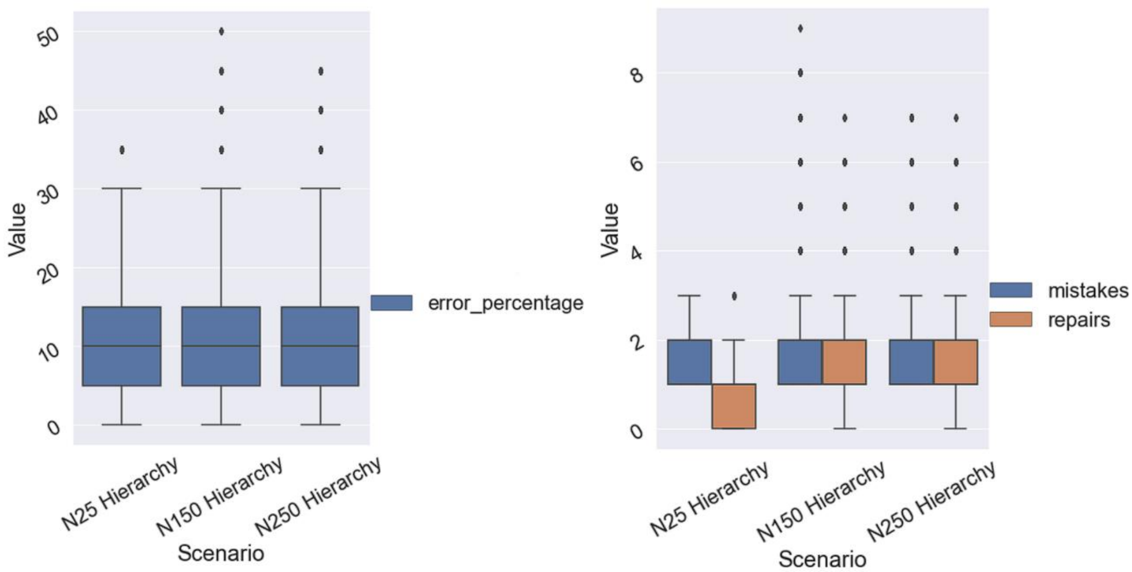


Figure 21: Error percentage (left), Mistakes encountered by controllers (Right)

Travel time

The boxplot (Figure 22) shows that the median travel time increases from scenario 1 to scenario 10 but decreases going from the 150 nodes to 250 nodes in network size. The same behaviour is noticeable in the right plot where travel times are normalised. Scenario 1 in the right plot does have a slightly bigger error bar than the other scenarios but still stays within minutes.

Figure 22: Effect of network size on mean travel time shows the regression plot of employee travel times. The mean travel time of the N = 25 scenario does not seem

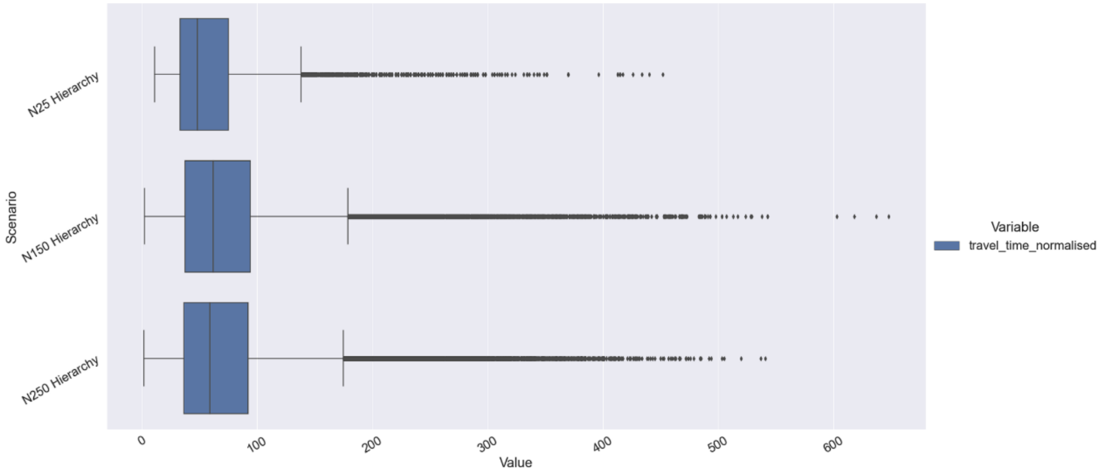


Figure 23: Effect of network size on normalised travel time

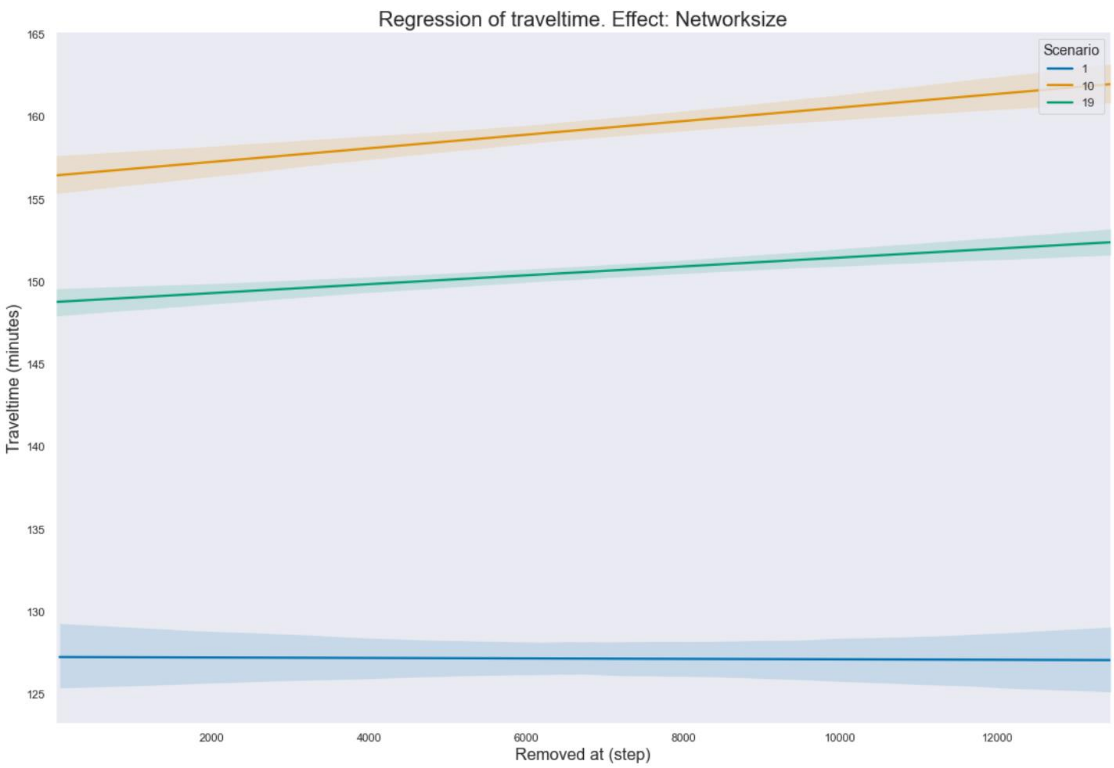


Figure 22: Effect of network size on mean travel time

to increase or decrease over time. However, the confidence intervals for this scenario leaves room for interpretation. Scenario 10 and 19 both show that travel times increases over time. With the travel time increasing more quickly for scenario 10, from around 157 minutes to 163 minutes of total mean travel time.

Busyness

Figure 24 describes the effects of network size on the busyness of total employees. There is a difference observable when comparing the values between scenario N = 25, N = 150 and N = 250. When going from scenario N = 25 to scenario N = 150 the mean busyness increases, while going from scenario N = 150 to N = 250 decreases the average busyness along with the standard deviation.

Interesting are the outliers that can be observed. Where the scenario N25 Hierarchy has no observable outliers in the boxplot but N150 and N250 have plenty of outliers. The outliers of N = 150 are very different from N = 250, with the highest of the latter being 24 and the highest of N = 150 being 31 information packages. As is also indicated by the KS table in Appendix I. The N = 25, N = 150 and N = 250 scenarios are all statistically different and do not come from the same distribution or can be achieved by randomness.

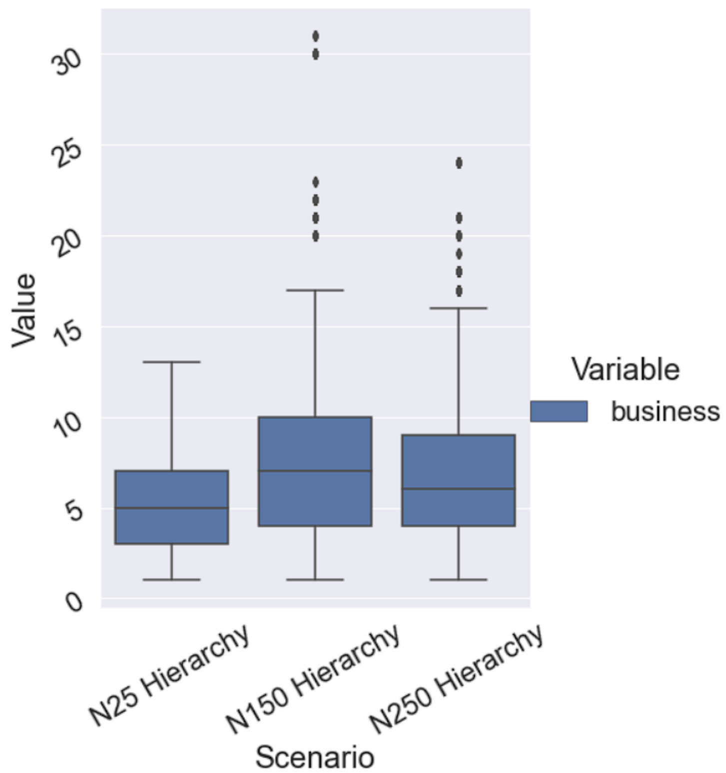


Figure 24: Busyness of employees | Effect of networksize

Effect of priority

The first effect studied is the effect of hierarchy. The scenarios analysed to study the effects of hierarchy are the scenarios N150 Hierarchy, N150 Hierarchy | prio 0.1 and N150 Hierarchy | prio 0.6 (10,11,12), respectively 0.0, 10% and 60% of the generated information packages being priority packages. The effects of priority percentage size are visualised and presented based on the three model KPIs. Since the base scenario is 10, the choice was made to visualise and discuss only the scenarios with similar values for the other parameters and leave out the smaller and larger network.

Data quality

The median error percentages for every information package for each of the scenarios do not differ. The mean percentage of the effects of priority percentage on data quality are scenario N150 Hierarchy: 9.92%, scenario N150 prio 0.1: 9.26% and for scenario N150 prio 0.6: 9.71%. However, when looking at the distributions, or the boxplots, we can observe that the differences are not visible. Only the outliers differ a little. The distribution shown in Appendix G, illustrates that there is no clear peak when comparing the three scenarios. Both error percentages 5% and 10% appear more often in the distribution.

The controller data shown in Figure 25 shows similar behaviour as the error percentages. In the scenario with 10% of the information being priority information, less mistakes are identified in the network. The boxplot shown in 25 (left) shows that scenario 10 has a maximum number of mistakes found of 9, scenario 11 find a maximum of 5 mistakes and scenario 12 finds 6 mistakes.

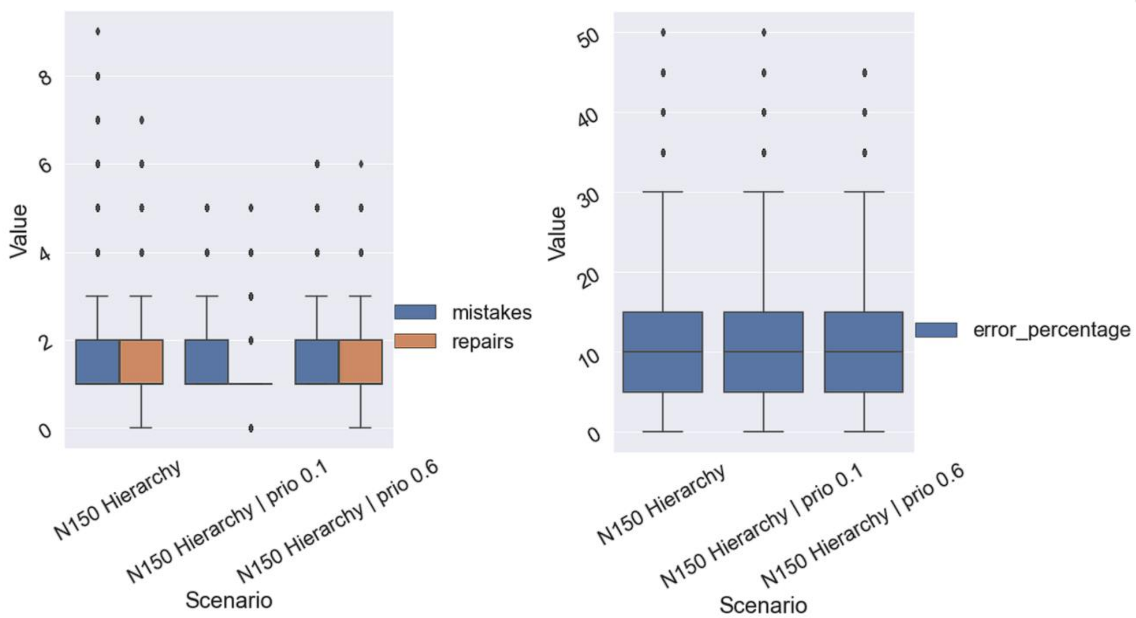


Figure 25 : Error percentage (left), Mistakes encountered by controllers (Right)

Travel time

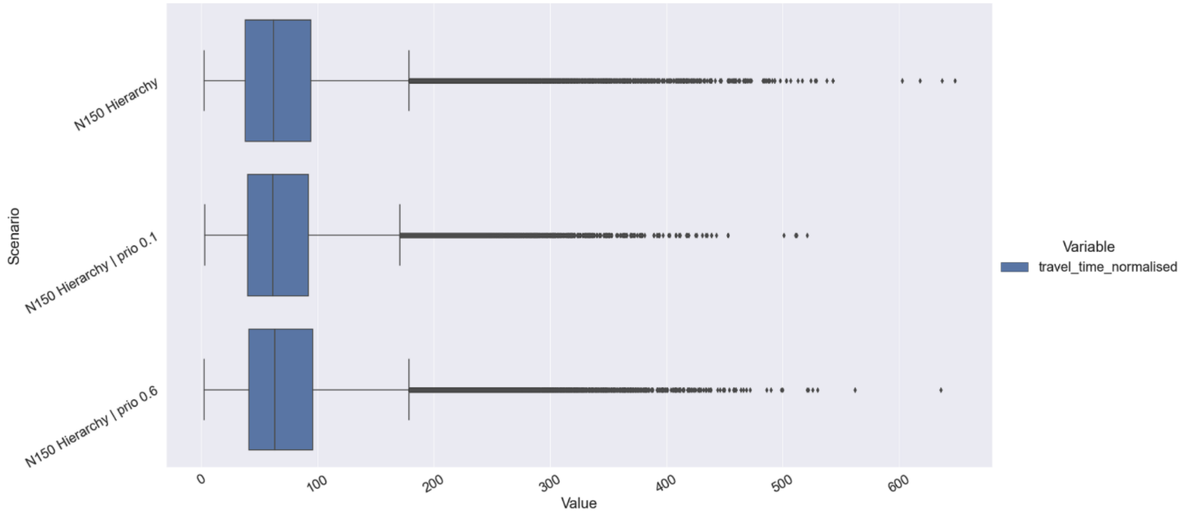


Figure 26: Effect of percentage priority on travel time. Normalised (right)

Figure 26 illustrates the effect of priority messages on travel time. The normalised travel time shows the travel time with scenario 11 having the lowest travel time, total and per step. Looking at the results from the per/step perspective, scenario 12 has a longer mean travel time than scenario 10. Figure 27 shows that all three scenarios have a travel time that increases over time. The increase for scenario 11 however is not as quick as for scenario 10 and 12, having a regression line that has a steeper incline.

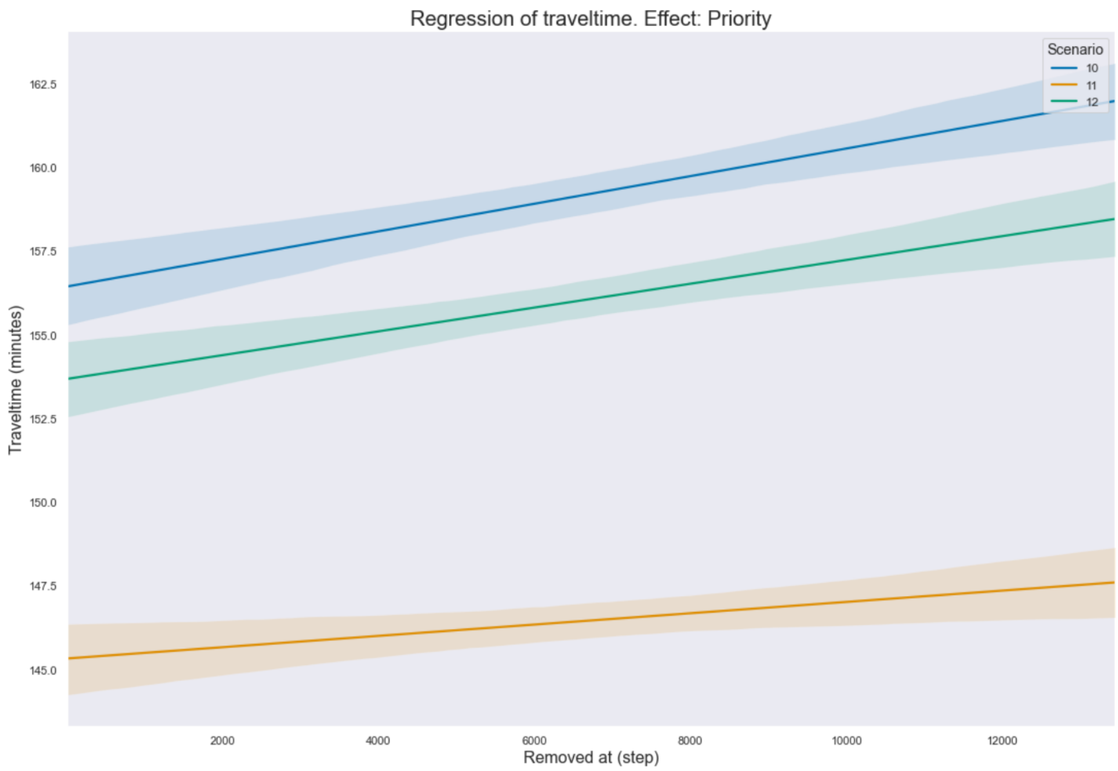


Figure 27: Effect of priority on travel time

Busyness

Figure 28 shows the effect of an increase in priority messages on the busyness of employees in a hierarchical organisational structure. There is a difference to be seen between the three shown scenarios. The differences however do not show a trend amongst the scenario's. As seen in Appendix I, the differences between the scenario's are statistically relevant. The outliers of scenario N150 Hierarchy and N150 Hierarchy | prio 0.1 have similar maximal values. The scenario with 0.6 percent of the messages being a priority message has a lower outlier of 27 compared to 32.

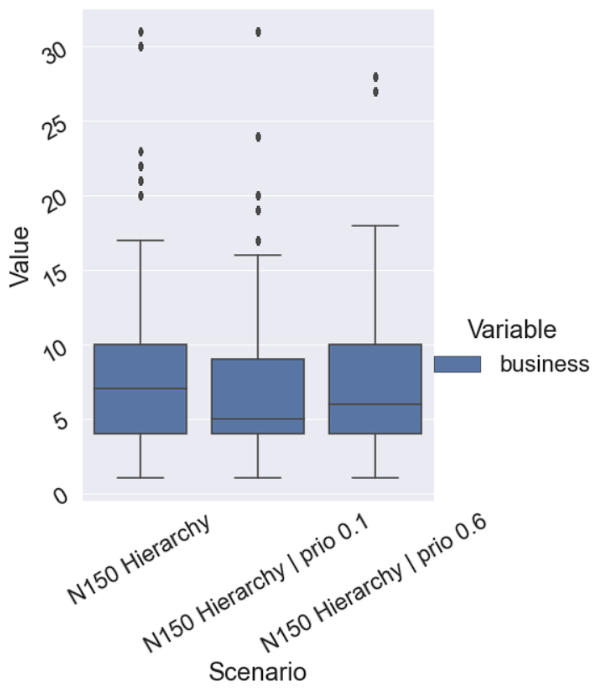


Figure 28: Busyness of employees | Effect of priority

Effect of controller percentages

The first effect studied is the effect of hierarchy. The scenarios analysed to study the effects of hierarchy are the Scenarios 28, 29 and 30, respectively 0, 15% and 30% of the normally autonomous agents is a controller. The effects of controller percentage are visualised and presented based on the three model KPIs.

Data quality

Increasing the number of controllers in an organisation shows the following effects on data quality measured with the error percentages. For the first scenario, 28: 11.14% the second scenario, 29: 9.42% and the final scenario 30: 8.48%. These however are the median values. As this does not say everything, the boxplot of these

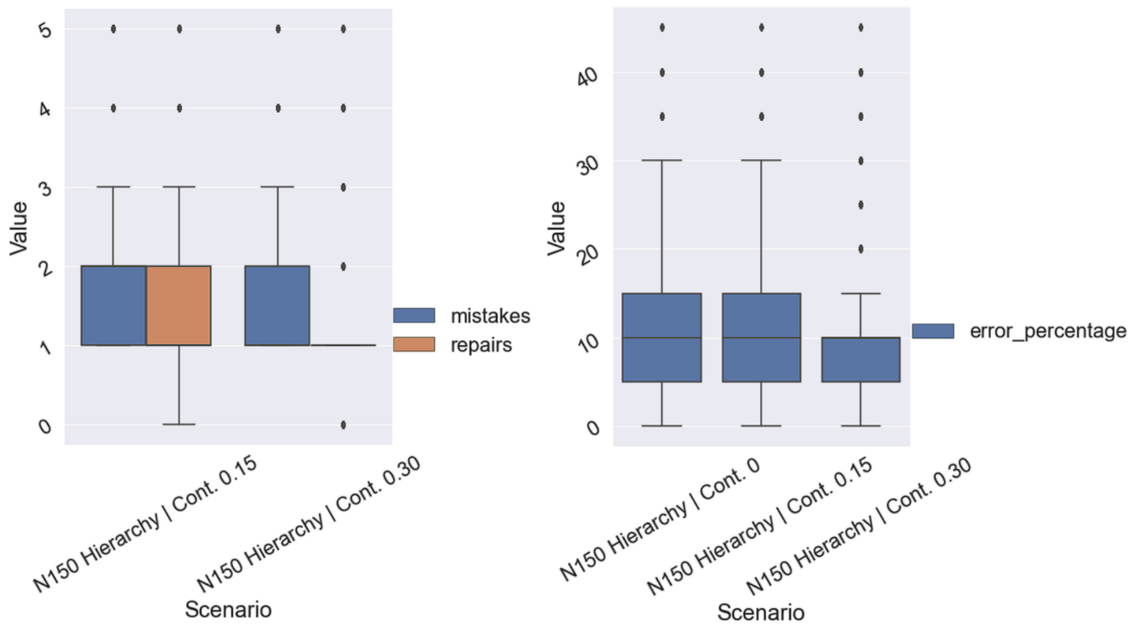


Figure 29: Error percentage (left), Mistakes encountered by controllers (right)

experiments are shown in Figure 29 (right). The figure shows peaks at 5% for scenario 28 and 29 and a peak at 10% for scenario 30. With all scenarios reaching a maximum value of 45%. The control 0.15 and control 0 show similar behaviour in Figure 29. In the distributions in the Appendix G, some slight differences can be observed that are not distinguishable in Figure 29. The controller data shown in Figure 29 (left) shows similar behaviour as the error percentages. In the scenario with 10% of the autonomous agents being controllers, more mistakes are identified in the network. Scenario 28 shows no values for controllers. The Figure 29 (left) shows that the scenarios have a maximum number of mistakes found of 5.

Travel time

Figure 30 shows the results of controller presence on travel time. Both the total travel time and the normalised travel time show the same trend in their median travel time with scenario 29 having the lowest median travel time, total and per step. Looking at the results from the per/step perspective, scenario 30 has a longer median travel time than scenario 29. However, from the perspective of total travel

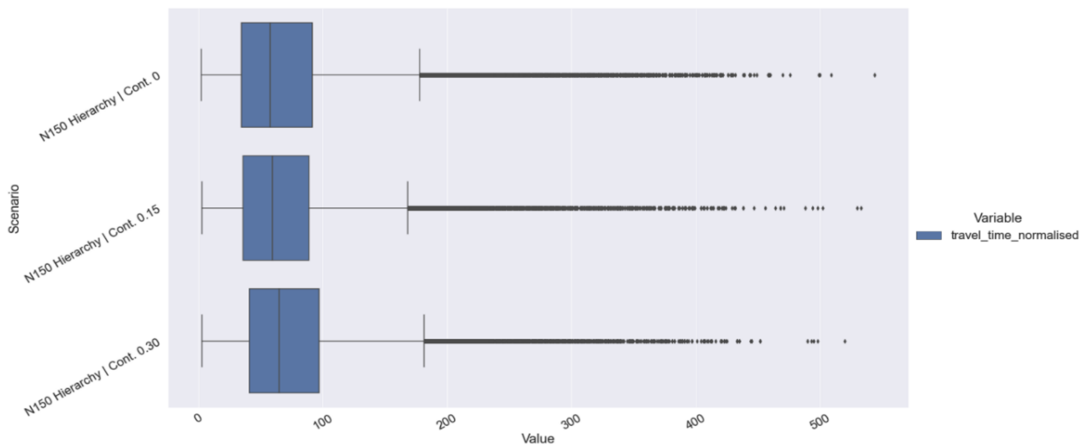


Figure 30: Effect of controller percentages on travel time. Normalised (right)

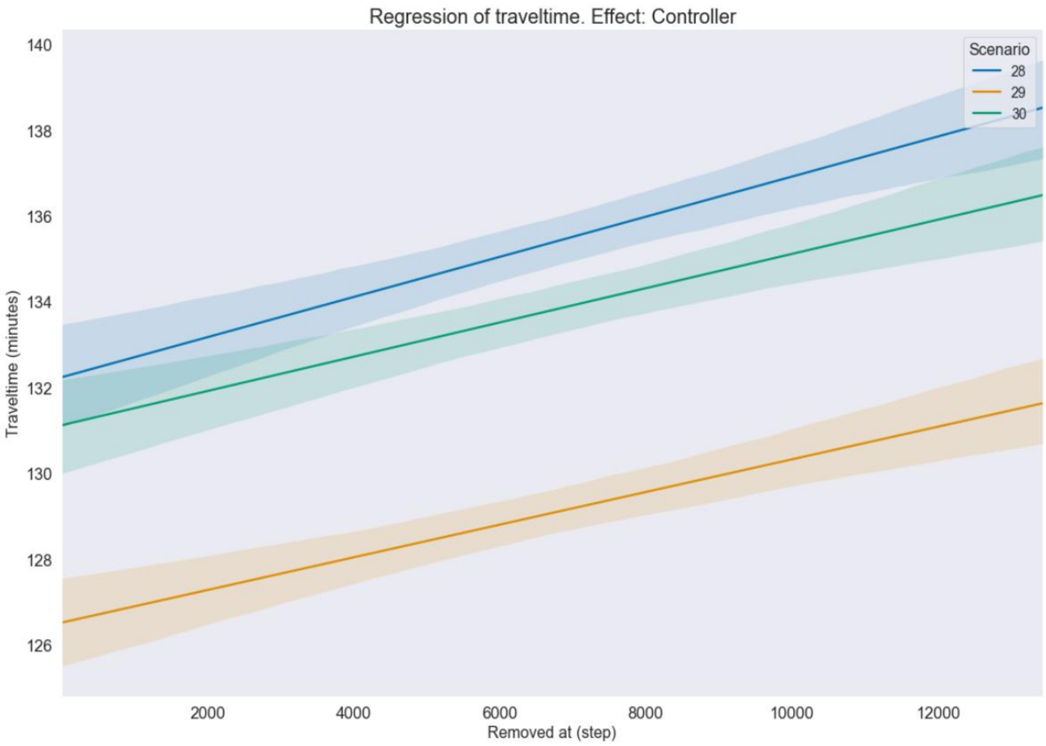


Figure 31: Effect of controller chances on normalised travel time

time, scenario 28 has a longer average travel time than scenario 30. Figure 31 shows that all three scenarios have a travel time that increases over time. The confidence intervals for all three scenarios are relatively wide and show uncertainty in the results.

Busyness

Figure 32 shows the effects of scenarios that contain more controllers in an organisation with a hierarchical organisational structure. There is a difference to be seen between the three shown scenarios. The differences however do not show a trend amongst the scenario's median or trends. However, what can be observed is that there is a downwards trend in maximum outliers from the three scenarios. As seen in Appendix I, the differences between the scenarios are statistically relevant. The outliers of scenario N150 Hierarchy and N150 Hierarchy | cont 0.15 have similar minimal values.

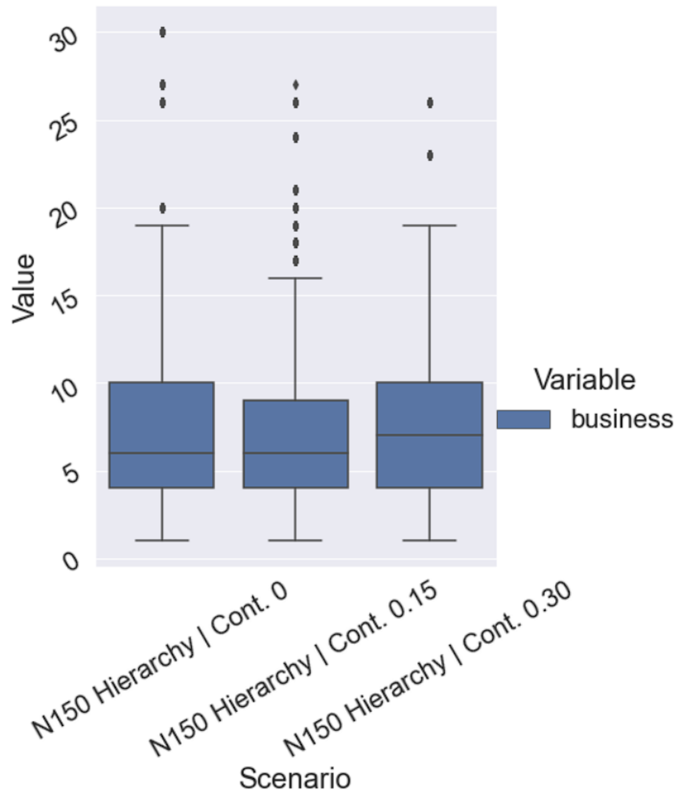


Figure 32: Busyness of employees | Effect of priority

6 Discussion

The results presented in the previous chapter are discussed in this chapter as well as the interpretation of the results, what the implications are of the results for the Ministry of Defence and other organisations and discussing the limitations of the study that arise due to assumptions and implementation of the agent-based model.

The model was labelled as verified, concluded based on the results of the model on the case study representative of the data from the ATU. The hybrid validation of the model used the data shown in Appendix E, verification. Validation of the results of both the case study, as well as the verification of the model used the same results from the case study model. Unfortunately, it was not possible to validate the theoretical models as there is no expert data or historical data.

6.1 Analysis of the results

This subsection will analyse the results from the experiments. Discussing their interpretation and implications. The subsection is divided into the subdivision among tested scenarios and how they affect the model's KPIs. Discussing the effects of the scenarios on data quality, travel time and employee busyness.

Effect of hierarchy

The scenarios that tested the effects of hierarchy, modelling hierarchical, democratic, and anarchic organisations are discussed in this subsection. The results in Figure 16 show that hierarchy influences the data correctness of information packages in the model. Networks that have higher q values show that information packages travelling over them have a bigger fraction of their data corrupted. This behaviour is also visible among controller employees. Regardless of network size, the average number of mistakes encountered by controllers rises, indicating that there is correlation between trophic incoherence and the number of mistakes. Figure 16 also shows that not only the median value increases but the maximum number of mistakes in an information package increases when q increases. Similar behaviour is visible in Figure 15, where one can observe that the number of mistakes identified by controllers increases when q increases. As mentioned in the results section, the findings are statistically significant (Appendix I).

Hierarchy and travel time have a more complicated relationship, as is visualised in Figure 17. For a network size of 25 nodes, travel time per step shows the lowest median travel time for a q = 2, but q = 1, has the highest travel time. For network sizes N = 150 and N = 250 the higher the q value the lower the median travel time per step gets. The outliers visible in the boxplot also cohere with this trend. Analysing Figure 18 (mean step travel time) shows different behaviours in the increase or decrease of travel time while an organisation is working. For example, scenario 16, where it can be seen that a regression line curves upwards, indicating that information travels slower later in the model. However, it can be observed that this scenario only has a slight slope upwards that is barely noticeable with a visual inspection. This can probably be explained by the random generation of paths for information packages. There is a possibility that the model, later during the process, generated more information packages that travelled a longer distance in the model, increasing total travel time but not the normalised travel time. Scenario 1, the network of size N = 25 and q = 0.6 shows an almost stable travel time over the course of the model-run. Besides scenario 1, where a stable travel time is visible, all other scenarios show that travel times increase over time, some faster than others (travel time/step). The normalised travel time also shows that networks with a q of 0.6 for networks of N = 150 and 250 have an overall higher travel time per step. These results seem to indicate that organisations with an organisational structure of 25, 150 and 250 nodes have lower travel times/step when they are less hierarchical.

Organisational structure also effects the busyness of an employee in the model. The results are visualised in Figure 19, showing that for smaller businesses (N = 25) the results are not statistically different enough for Democracy and Hierarchy of N=25. Thus, eliminating the results from this analysis. However, for the network sizes N=150 and N=250 a trend can be observed. Employee business, median and maxima, increase when q increases, or to phrase it differently: When the network structure changes from a hierarchy to a democracy and then to an anarchy the busyness of employee increases.

Summarizing, hierarchy does clearly seem to affect the data correctness of information-flow in the model environment. Showing that an increase in trophic incoherence increases error percentages and mistakes in an organisation. If we look at the normalised travel time, the time that it takes for an information packet to go to the next employee, we see that less hierarchy increases travel times. As shown in the distribution in Appendix G the normalised travel times removes most of the outliers created by randomness over creating certain length of paths. The results also show a consistent effect of hierarchy on busyness. Decreasing the busyness of employees when the trophic incoherence of the organisational network structure decreases.

Effect of network size

Network size affects information quality in different ways. As discussed in the corresponding results section and shown in Figure 21, both controllers and error percentages indicate no clear pattern due to network size. In the tested examples/scenarios a clear increase from N = 25 to N = 150 is visible, indicating that the number of mistakes goes up with network size. However, the results also show that the difference between N = 150 and N = 250 are minimal, the number of mistakes found by controllers and the median error percentage of information packages data even stay the same. The only differences observed are in the outliers. Where it can be seen that for the network of N = 150 the outliers have a higher value.

Along with the effect on data quality, network size also seems to influence the travel times of information. Figure 23 shows the mean travel time/step. These differences however can be explained due to the network size. The increase in travel time when going from N = 25 to N = 150 is probably due to increased path lengths in the network and the added chance of going past non-autonomous or collector/controller employees. The travel time however decreases when the network size is further increased to N = 250. This behaviour might be explained by the autonomous/non-autonomous employee differences between both scenarios. Appendix F shows that for scenario 1 the ratio autonomous/non-autonomous is (12/3), for scenario 10 (60/24) and for scenario 19 (112/36). Indicating that more non-autonomous employees increase travel times in an organisation since these are the only major differences between the networks that could theoretically explain the results since the other parameters are very similar. As the behaviour of the scenarios is the same in both normalised and total travel time, only the total travel time increase over time was plotted. These results show that travel times for scenarios 10 and 19 do increase over time. However, the incline for scenario 10 is steeper than the incline of scenario 19, giving another indication that non-autonomous employees delay information travel speeds in organisations.

Network size does seem to effect employee busyness in a similar way that is observed with information travel times. The workload increases drastically from scenario N25 to N150 and lowers between N150 and N250. This busyness differences can be explained by the increase in network size, but more likely are linked to the differences in non-autonomous employees between the networks.

In summary, network size does seem to affect information quality when looking at an increase from N = 25 to N = 150, but the differences between N = 150 and N = 250 are minimal and show even a decrease. The decrease in travel times might be explained by the higher number of non-autonomous employees in the scenario 10 networks.

Effect of priority messages

Figure 25 shows the effects priority messages have on the controller statistics. Scenarios 10, 11 and 12 are used to discuss the effects under a hierarchy value of 0.6, as this is the baseline of the research. The results from the figure show that there is an optimum for the minimum number of mistakes found with 10% priority messages. This is only an optimum however when looking at the outliers (maxima), the boxplots themselves do not differ among the scenarios. The differences can not be explained by the number of controllers present in the network since the network statistics of the scenarios in Appendix F show that for all networks, the mean number of controllers was approximately 10. The distributions in the Appendix G, show the mistakes and error percentages. The distribution of error percentages shows what the mean values already indicates, a minimum difference between the different scenarios. The first two scenarios shown in Figure 25 (right) do not differ. Only when increasing priority to 0.6 the outliers become one less. If this is due to randomness or the priority messages is unsure in this phase of the research.

Figure 26 and Figure 27 show the minimum effect of the percentage of priority messages on travel time in the network. When 10 percent of the messages is a priority message, one can see a slightly less normalised travel time and travel time/step, around 5 minutes compared with scenario 12 and 2 with scenario 10. Figure 28 shows the total travel time, where it can be observed that travel times increase over time for all scenarios.

The employee busyness is affected by the increase in priority messages percentage. Compared to the normal, when adding a percentage of priority messages, a big decrease in busyness can be observed in the boxplot in Figure 28. However, it can also be observed that this effect is negatively affected when the percentage of priority messages increases further. It indicates that 10% of the messages being a priority message does seem to work within the timeframe of the model.

Summarizing, priority messages do not seem to have a strong effect on data quality. With travel times an optimum is found with 10% priority which might be explainable due to the mean travel time of priority information being lower than regular information packages (Figure 20).

Case study

The effect of priority messages was also tested on the case study along with the random network scenarios. The network statistics of the network show a trophic incoherence of 1.78, which makes it an anarchistic network according to the paper by Pilgrim et al. (2020). The high value of q is due to the high number of cycles that are inside the network. Considering the small size, a high q value is easily reached.

Data quality does not seem to be affected by priority messages with the difference being 0.03% among the scenarios error percentages. The distribution shown in Figure 11 shows that most of the information packages arrive with 10% of their data corrupted for all three of the scenarios.

Mean (normalised) travel times of information packages do not seem to be strongly affected by priority messages. The differences between scenarios are minimal but show a travel time decrease when priority percentages increase. An effect observed in Figure 13 is that for scenario 0, no priority messages present has a regression line that decreases over time. However, as the error margins of the plot shows, data on travel time increase cannot be used to make any clear conclusions. Similar effects are observed with busyness. The scenario with 10% priority messages has the lowest mean busyness but the differences are not significant.

Effect of controller employees

Controller employees strongly affect data quality in an organisation. The results mentioned in the corresponding subsection of results show that the data correctness increases when the number of controllers increase. This is explained by the function of controllers, as they repair mistakes that they identify. The decrease in mistakes found by controllers can be explained by their numbers as well. When there are more controllers, they can be on more paths, and multiple times on one path, decreasing the number of mistakes before they reach other collectors. Which is confirmed by the distributions showed in the Appendix G. They show that the maximum number of mistakes found by a controller stays at 5 but the number of times this maximum is reached is clearly lower.

Travel time is affected by the number of controller employees in an organisation. Adding a slight number of controllers lowers the median travel time and the outer quarters of the boxplots. This can be explained with the model implementation and randomness, since with every mistake the total waiting time of an information package increases with 60 minutes. When there are controllers in the network and they find little to no mistakes, travel times will decrease. However, when there are more mistakes found, it will increase the total travel time. This is the likely behaviour observed in the figures. However, when more controllers are added, a 30% increase in normalised travel time can be observed. Figure 31 shows the increase of mean normalised travel time in which is shown the effect of 30% of an organisation being controllers versus none or 15% were the substantial difference of 4.5 minutes per step is visible.

Controller employees do seem to indicate that they affect employee busyness. Increasing the number of controllers from 0 to 15 percent decreases the busyness of all employees but increasing the controllers from 15 to 30 percent increases employee busyness. This effect might be explained due to the backlogs created by introducing controllers into the organisation. This backlog is then increased when more controllers are present.

6.2 Limitations

As with every study, this study has its limitations. Limitations that affect the interpretation and the weight of results of a study. The results, their analysis and how they are affected by limitations in this study are discussed in this subsection.

The assumption made in the model conceptualisation that there are 4 different types of employees do affect the implications of the study. It is an abstract representation of organisations that enables easier comparability between different organisational structures and reduced model complexity, improving run time and reusability in other scenarios. Nevertheless, with this abstraction the finer details and workings of organisations are lost. Organisations exists of employees with many different functions that sometimes fall into multiple categories of agent types in this model. An example can be found in the case study. Taking into consideration the BPMN diagram, we can observe that some employees have a collector task, controller task and non-autonomous tasks all combined. This is something that is more frequently observed in smaller companies. Unfortunately, this nuance is not present in the current model. The results of the model are still valid to study the effects of the different agent types and allows for the generalisation of an organisation and its employees but is not a direct copy of the workings an organisation. Thus, performance results cannot be directly generalized to specific employees' performance.

Another limitation of the study is that only formal information-flows are modelled. In practice, organisations also have informal information-flows. When co-workers know each other, requests can be passed around the chain of command, decreasing total travel times for information.

Another limitation of the study that affects the generalizability and direct usability of the results and conclusions of this study is the lack of validation. Unfortunately, validation by experts was not possible with the employees of the case study. The only validation comes in the form of the verification, where input values are compared to output values generated by model behaviour. Validation of the theoretical study results was not possible with experts, nor with literature validation since comparable studies or network data have not been published yet.

As with many fields of science, when dealing with models, simulations or other forms of analysis that require input data, a bias can be introduced due to the quality of the input data. In the case of this study there is a chance for the so called 'garbage in, is garbage out'. An example was seen with the network problems that generated a small artefact in the results. When networks are curated, just like the one from the case study, the model did not generate these artefacts.

Results from the case study model are limited. As shown in the results, patterns cannot easily be identified in small networks and the hierarchy values of these small teams seem to be indicative of an unstructured organisation.

Model limitations

The first limitation due to model implementation is related to non-autonomous employees. Non-autonomous employees only send information to the other employee 'above' them. If this employee itself is a non-autonomous employee, they will not reach further up but decide themselves. This is a simplification of organisational processes to reduce model complexity. Unfortunately, behaviour where sometimes a manager's manager signs off on requests made by an employee are lost. Thus, increasing workloads, or travel times due to non-autonomous employees are limited to waiting times of only one higher up manager, meaning that actual effects of non-autonomous employees might be higher.

The second limitation arises from the implementation of generating random networks. The current form of generating random networks does not allow for specific number of agent types to be present. Decreasing flexibility of the model and increasing randomness in the results (could be observed in the results of network size).

Unfortunately, the results showed that there are other limitations due to the implementation of random networks. For example, unexpected behaviour presented while visualising the results of the random network. Instead of travel time distributions that followed similar patterns like the verification of the model, enormous outliers of suspiciously low travel times were observed. As explained in verification, travel times should not go below certain numbers, since the minimum work size of information packages is set and the maximum speeds times are set as well. Analysing these outliers showed that information packages were having travel times of 0 instantly reaching a destination. The cause identified was that sources directly fed into sinks. This however did not explain the slightly higher travel times. The eventual reason for this behaviour was not identified. To ensure usability of the results, a correction for these peaks was performed. Removing two employees from the total employees visited for the first employee being skipped, and one for the counting the sink as an employee visited. Subsequently removing all information package data from travel time that was below 10 minutes. The network statistics show that the only differences between scenarios are the factors that were set in the experimental design. Combining this with the correction of the data does ensure that we can still use it for conclusions on relations between the tested factors. However, generalisation of the travel time results cannot be guaranteed as long as the problem is not identified making the results of this KPI more suggestions than conclusions.

7 Conclusion

This chapter presents the conclusions of the research findings in this study, the recommendations for the Ministry of Defence and recommendations for future research. The conclusions are explained through answering the sub research questions and afterwards the main research question.

7.1 Answering the sub research questions

To answer the main research question, first the five following sub-research questions are answered.

1. *What is information-flow quality, and can it be improved?*

Information-flow is the distribution of information in its widest sense through a system or organisation. It is defined by the relation between two separate agents that are connected and share information based structural and behavioural rules. Information-flow quality is a performance measure that is split into three different KPIs, data quality, timeliness and busyness, that are used to measure information-flow. The literature studies found in the background section, indicate that a flatter organisation (q = 1, is theorised) would improve information timeliness. Increasing the number of controllers in an organisation was theorized to positively affect information data quality.

2. *What are the organisational structures that determine information-flow at the Air Transport Unit?*

The Dutch Military Air Transport Unit is a small sub-unit of the DVVO which in itself is a sub-unit of DOSCO. The organisation has many responsibilities all surrounding planning and advice on air transport. The time that information takes inside this unit has a wide range and is used as base input for the model. Certain special characteristics of a military organisation like priority messages that have unrivalled precedence. This interesting behaviour was added to the scenarios of sub question 1.

3. *How can the information-flow within the organisations be conceptualized and implemented using ABM techniques?*

Organisational structures that allow for information-flow inside organisations were conceptualised in Chapter 3. Organisations should be represented as networks, where at every node an employee works. Information packages travel from employee to employee, only to be send to the next after processing, introducing the change of data corruption. Since organisations are made up of different employees with different tasks, four different employee types were conceptualised to abstractly represent an entire organisation. The employee types conceptualised are autonomous agents, non-autonomous, collectors and controllers. Each representing a generalised function inside organisations that might affect timeliness and or data accuracy of information.

The implementation was successfully implemented using ABM techniques in a python environment. Generating random networks to increase the test data for the different scenarios. Verification of the model on a smaller network, the case study, shows that the model and agents work as intended. However, a problem was later identified with the random networks. They introduce randomness in the results that is not always explainable.

4. *What are the effects of the different scenarios described in sub-question 1 on the information-flow quality?*

The different scenarios indicate that timeliness of information or the travel time of the information package is affected by hierarchy. The time that an information package takes to travel from employee-to-employee decreases when the trophic incoherence value increases. This same hierarchy does affect the data quality in an opposite manner. Increasing error percentages and mistakes the higher the value for q becomes. The effect of hierarchy on the network flow / busyness KPI is similar to that of data quality. Concluding that considering the effect of hierarchy an optimum for data quality and timeliness can be found in scenario where q = 1. Balancing timeliness with data quality.

The experiments with network size are included to be able to generalise the results to other organisations or department than the ATU. As explained in the discussion, network size does affect information-flow quality. The connection however is not clear from the results. The results seem to indicate that the higher travel times might be due to the larger fraction of employees being non-autonomous in the larger networks, increasing travel times and busyness. The lower data quality is probably due to longer paths travelled by information packages in larger organisations. Meaning that more people process the same data.

The number of controllers affects information-flow quality by increasing information's data quality and increasing mean travel times of information packages, finding an optimum in 15% of the autonomous agents being controllers.

The percentage of messages being priority messages does not seem to affect information-flow quality in a significant manner. Data quality is not affected and timeliness of information packages in the network is only slightly lower with 10%, probably due to the faster travel speeds observed when comparing regular and priority messages.

Concluding that proposed solutions such as a more horizontal organisation, with a value of q = 1 and employees that are more autonomous, do speed up information timeliness in the model but might negatively affect data quality. Combining this with the tendency for busyness to be negatively affected by higher q values as well, solidifies this optimum.

5. *How can the results obtained in previous SQ be generalized for organisations and their departments to improve their information-flow quality?*

Introducing the random networks and testing on different sized networks has allowed for the representation of different types of organisations. Values for the trophic incoherence have ranged from 0.6 to 2, including many types of organisations this way. Allowing for the results of the experiments to be transferred to other organisations.

7.2 Answering the main research question

The goal of this thesis was to find the answer to the following question:

What are the effects of formal network structures within public organisations on their information-flow quality?

Using an ABM model that abstractly represents public organisations and an extensive literature background an attempt was made to answer this question. The results and answers to the sub-questions show connections between the hierarchy of an organisation and its information-flow quality, affecting data quality and timeliness in opposite manners. Suggesting an increase in data quality when an organisation becomes more hierarchical but also decreasing travel time in the same manner. Non-autonomous employees harm the timeliness of information-flow. Indicating that a public organisation with more autonomous employees will have faster communication.

7.3 Recommendations for organisations

Based on the model results and the case study, different recommendations can be given. During the case study, and while making up the BPMN graph few control points surfaced in the network. Back and forwards communication about errors in information happens all before the PN2 planner. This was not included into the model as employees are represented as one type of agent, but the model does suggest that having control points further down the line helps with data accuracy. Unfortunately, the case study did not allow for the testing of hierarchy scenarios on empirical networks. Therefore, the results from the random networks are used to recommend strategies.

For the Ministry of Defence, an organisation hierarchical in nature, data accuracy might be of greater importance than timeliness of information. This might be of great importance for the intelligence branches of the Ministry who's task it is to provide as accurate information as possible and where mistakes can have disastrous consequences.

However, if the goal of certain branches or units is to communicate as quickly as possible while keeping data quality in mind, the results of this study would suggest that a less hierarchical approach would improve the timeliness of information. Creating a more horizontal organisation, where there is less hierarchy, and more autonomy is granted to employees. The model results also suggest that there is a strong effect between hierarchy and busyness of employees. Thus, a hierarchical structure seems even more logical in this model.

Priority messages do seem to be harmless in the case of travel times for all information packages and for regular information packages under the same circumstances. This is of course assuming that part of the daily tasks is dropped until the crisis is dealt with.

Even though the data and parameters are based on the case study of the ATU, the generalizability of the results is high. Keeping the problem in the random networks in mind, the results are still based on abstract representation of public organisations. However, the type of agents, network structures and sizes make the data generalizable for organisations out of the public sector as well. These organisations have often already more horizontal organisations but the results considering the effectiveness of hierarchy on data quality could be useful for branches where accuracy goes over timeliness. Furthermore, the effects of controllers in an organisation and how network sizes affect information-flow quality can be generalized to the private sector, suggesting a balance between autonomous employees and controllers of 15% based on the model outcomes.

7.4 Recommendation for future research

Several interesting research topics and model changes or additions are recommended for further exploration based on the outcome of this research.

First, additional research and testing is necessary for the generation of random networks who represent an organisations information-flow. Networks should have the requirements of this research with the addition of realistic paths. Random networks that are generated where paths are a minimum length, allowing for more realistic representation of organisations would help to further aid in the generalizability and applicability of the model results. Eliminating errors that slipped into the results because of this.

Second, the effects of workload on data accuracy. The model in its current state does not take the relation between workload and data accuracy into account. However, testing models to see if there is a connection between work precision and being overworked and how this is interconnected with formal organisations would help to widen our knowledge of formal organisations and information-flow quality.

Third, the implementation of information package generation might be changed to test workload in organisations. The opportunity is here to test whether organisations can handle influxes in the number of information packages generated. Changing from 1 information packet per x minutes, to several information packages that arrive in a different time frame. Analysing how organisational structures handle influxes in information and how this affects the information-flow quality, would also be useful.

Fourth, mixing this up with influxes in priority information and comparing travel speeds of information sent before and after priority messages might provide insights in the effects of priority messages when not everything is dropped, but things try to continue as they would normally.

Fifth, there is an opportunity to expand the existing scenarios and widen the scope to bigger organizations, testing whether the model behaves differently for organisations magnitudes larger.

Sixth, there is an opportunity to expand on the existing model by adding more complex behaviour to the employee agents.

- Change the behaviour of the non-autonomous agent, so that if they ask permission to a non-autonomous agent, there is the chance that they ask their supervisor.
- Add the option to non-autonomous agents, to also provide priority to permission requests for priority information packages.

- Add the option to controllers to identify that mistakes are present without being able to repair it. Instead, the controller should send the information package back to the previous employee, who will check if they made the mistake. If not, they send it back. Until the information package is repaired. Removing a big part of the abstraction but gaining a more realistic representation of the effect of mistakes on the timeliness of information-flow.

Finally, an empirical study using the current state of the model with a larger case study should be performed. Ideally an organisation where the network size is closer to the one hundred than to zero nodes. The expert knowledge available and larger sized network will provide for an expert validation of the model. Giving more weight to future research with the model. Empirical research will provide the added functionality that using expert knowledge available in this case study, it will be possible to generate different strategies for an organisation, create hierarchical representations of those strategies and run them through the model.

Bibliography

Ahituv, Niv, Magid Igbaria, and Aviem Sella. 2015. “The Effects of Time Pressure and Completeness of Information on Decision Making.” *Http://Dx.Doi.Org.Tudelft.Idm.Oclc.Org/10.1080/07421222.1998.11518212* 15(2):153–72. doi: 10.1080/07421222.1998.11518212.

Albino, V., P. Pontrandolfo, and B. Scozzi. 2002. “Analysis of Information-flows to Enhance the Coordination of Production Processes.” *International Journal of Production Economics* 75(1–2):7–19. doi: 10.1016/S0925-5273(01)00177-3.

Arga Jafino, Bramka. 2017. “Measuring Freight Transport Network Criticality | TU Delft Repositories.” Master Theses, Delft University of Technology, Delft.

Ballou, Donald P., and Harold L. Pazer. n.d. “Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems | Management Science.” Retrieved August 19, 2022 (https://pubsonline-informs-org.tudelft.idm.oclc.org/doi/abs/10.1287/mnsc.31.2.150).

Bastian, Mathieu, Sebastien Heymann, and Mathieu Jacomy. 2009. “Gephi: An Open Source Software for Exploring and Manipulating Networks | Proceedings of the International AAAI Conference on Web and Social Media.”

Ben-Arieh, David, and Moshe A. Pollatscheck. 2002. “Analysis of Information-flow in Hierarchical Organizations.” *International Journal of Production Research* 40(15):3561–73. doi: 10.1080/00207540210137611.

Checkland, P. B. 1988. “Information Systems and Systems Thinking: Time to Unite?” *International Journal of Information Management* 8(4):239–48. doi: 10.1016/0268-4012(88)90031-X.

Citroen, Charles L. 2011. “The Role of Information in Strategic Decision-Making.” *International Journal of Information Management* 31(6):493–501. doi: 10.1016/J.IJINFOMGT.2011.02.005.

Creti, Anna. 2001. “Firms’ Organization and Efficient Communications Networks.” *Manchester School* 69(1):77–102. doi: 10.1111/1467-9957.00236.

Crooks, Andrew T., and Alison J. Heppenstall. 2012. “Introduction to Agent-Based Modelling.” Pp. 85–105 in *Agent-Based Models of Geographical Systems*, edited by A. J. Heppenstall, A. T. Crooks, L. M. See, and M. Batty. Dordrecht: Springer Netherlands.

Day, Jamison M., Iris Junglas, and Leiser Silva. 2009. “Information-flow Impediments in Disaster Relief Supply Chains.” *Journal of the Association for Information Systems* 10(8):1–1. doi: 10.17705/1jais.00205.

Defensie. 2021. “Militaire Steunverlening Breidt Zich Verder Uit over Limburg | Nieuwsbericht | Defensie.Nl.” *Denfensie*.

Defensie. n.d.-a. “Current Missions | Missions Abroad | Defensie.Nl.”

Defensie. n.d.-b. “Korte berichten [2/2] - 12 - Landmacht.” Retrieved September 28, 2022 (https://magazines.defensie.nl/landmacht/2015/06/10_korte-berichten-2-2).

Durugbo, Christopher, Ashutosh Tiwari, and Jeffrey R. Alcock. 2013. “Modelling Information-flow for Organisations: A Review of Approaches and Future Challenges.” *International Journal of Information Management* 33(3):597–610. doi: 10.1016/J.IJINFOMGT.2013.01.009.

DVVO. n.d. “Introductie Presentatie DVVO.”

Ehsani, Maryam, Ahmad Makui, and Soheil Sadi Nezhad. 2010. “A Methodology for Analyzing Decision Networks, Based on Information Theory.” *European Journal of Operational Research* 202(3):853–63. doi: 10.1016/J.EJOR.2009.06.028.

Fiala, P. 2005. “Information Sharing in Supply Chains.” *Omega* 33(5):419–23. doi: 10.1016/J.OMEGA.2004.07.006.

Gøtzsche-Astrup, Oluf, Morten Brænder, and Vilhelm Stefan Holsting. 2021. “Network or Hierarchy? Personality Profiles of Future Military Leaders.” *Nordic Psychology* 0(0):1–22. doi: 10.1080/19012276.2021.1945947.

Hagber, Aric A., Daniel A. Schult, and Pieter J. Swart. 2008. “Exploring Network Structure, Dynamics, and Function Using NetworkX.” *Proceedings of the 7th Python in Science Conference (SciPy2008),* 11–15.

Hill, Carolyn J., and Laurence E. Lynn Jr. 2005. “Is Hierarchical Governance in Decline? Evidence from Empirical Research.” *Journal of Public Administration Research and Theory* 15(2):173–95. doi: 10.1093/jopart/mui011.

Huntington, Samuel P. 1981. *The Soldier and the State: The Theory and Politics of Civil-Military Relations*. Harvard University Press.

Jannati, Sima, and Sarah Khalaf. 2022. “A Ruffled Mind Makes a Restless Forecast: The Effect of Rest on Analyst Forecast Accuracy.”

Johnson, Samuel, Virginia Domínguez-García, Luca Donetti, and Miguel A. Muñoz. 2014. “Trophic Coherence Determines Food-Web Stability.” *Proceedings of the National Academy of Sciences* 111(50):17923–28. doi: 10.1073/pnas.1409077111.

Jonkman, Bas. 2021. “High Water Limburg Summer 2021 More Drastic than River Floods in 1993 and 1995.” *TU Delft*.

K. H. Dam, I. Nikolic, and Z. Lukszo. 2013. *Agent-Based Modelling of Socio-Technical Systems*. Springer Dordrecht.

Kim, S., and H. Lee. 2006. “The Impact of Organizational Context and Information Technology on Employee Knowledge-Sharing Capabilities.” *Public Administration Review* 66(3):370–85. doi: 10.1111/j.1540-6210.2006.00595.x.

Klischewski, R., and H. J. Scholl. 2006. "Information Quality as a Common Ground for Key Players in E-Government Integration and Interoperability." Pp. 72–72 in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*. Vol. 4.

Krovi, Ravindra, Akhilesh Chandra, and Balaji Rajagopalan. 2003. "Information-flow Parameters for Managing Organizational Processes." *Communications of the ACM* 46(2):77–82. doi: 10.1145/606272.606277.

Li, Suhong, and Binshan Lin. 2006. "Accessing Information Sharing and Information Quality in Supply Chain Management." *Decision Support Systems* 42(3):1641–56. doi: 10.1016/j.dss.2006.02.011.

Liu, Rong, and Akhil Kumar. 2011. "Leveraging Information Sharing to Configure Supply Chains." *Information Systems Frontiers* 13(1):139–51. doi: 10.1007/S10796-009-9222-8/FIGURES/6.

Lurie, Nicholas H., and Jayashankar M. Swaminathan. 2009. "Is Timely Information Always Better? The Effect of Feedback Frequency on Decision Making." *Organizational Behavior and Human Decision Processes* 108(2):315–29. doi: 10.1016/J.OBHDP.2008.05.005.

McEvily, Bill, Giuseppe Soda, and Marco Tortoriello. 2014. "More Formally: Rediscovering the Missing Link between Formal Organization and Informal Social Structure." *Academy of Management Annals* 8(1):299–345. doi: 10.1080/19416520.2014.885252.

Mintzberg, H. 1980. "Structure in 5's - a Synthesis of the Research on Organization Design." *Management Science* 26(3):322–41. doi: 10.1287/mnsc.26.3.322.

Moutsinas, Giannis, Choudhry Shuaib, Weisi Guo, and Stephen Jarvis. 2021. "Graph Hierarchy: A Novel Framework to Analyse Hierarchical Structures in Complex Networks." *Scientific Reports* 11(1):13943. doi: 10.1038/s41598-021-93161-4.

NATO. 2022. "Relations with Russia."

Niagara Institute. n.d. "'I Keep Making Mistakes At Work': What To Do To Break the Pattern." Retrieved October 5, 2022 (https://www.niagarainstitute.com/blog/i-keep-making-mistakes-at-work).

NL Times. 2021. "Limburg Flood Damage Estimated at Hundreds of Millions of Euros; Dike Breaks in Gelderland | NL Times." *NL Times*, June.

Oliver, Gillian. 2008. "Information Culture: Exploration of Differing Values and Attitudes to Information in Organisations." *Journal of Documentation* 64(3):363–85. doi: 10.1108/00220410810867588.

Oxford English Dictionary. n.d. "Information, n. : Oxford English Dictionary." *Oxford English Dictionary*. Retrieved July 22, 2022 (https://www.oed.com/viewdictionaryentry/Entry/95568).

Paramount Business Jets. n.d. “Boeing 737-300 Private Jet Charter, Hire Costs, and Rental Rates.” *Paramount Business Jets*. Retrieved September 30, 2022 (https://www.paramountbusinessjets.com/private-jet-charter/aircraft/boeing-737).

Pedroso, Marcelo Caldeira, and Davi Nakano. 2009. “Knowledge and Information-flows in Supply Chains: A Study on Pharmaceutical Companies.” *International Journal of Production Economics* 122(1):376–84. doi: 10.1016/J.IJPE.2009.06.012.

Pilgrim, Charlie, Weisi Guo, and Samuel Johnson. 2020. “Organisational Social Influence on Directed Hierarchical Graphs, from Tyranny to Anarchy.” *Scientific Reports* 10(1):1–13. doi: 10.1038/s41598-020-61196-8.

Rainey, Hal G., Robert W. Backoff, and Charles H. Levine. 1976. “Comparing Public and Private Organizations.” *Public Administration Review* 36(2):233–44. doi: 10.2307/975145.

Rainey, Hal G., and Barry Bozeman. 2000. “Comparing Public and Private Organizations: Empirical Research and the Power of the A Priori.” *Journal of Public Administration Research and Theory* 10(2):447–70. doi: 10.1093/oxfordjournals.jpart.a024276.

Rowley, Jennifer. 1998. “What Is Information?” *Information Services & Use* 18(4):243–54. doi: 10.3233/ISU-1998-18402.

Seal, Alan. 1988. “Value-Added Processes in Information Systems, by Robert S. Taylor. Norwood, NJ : Ablex Publishing Corp., 1986. Xiii, 257 Pp. 24 Cm. Illus, Tabs, Bibliog, Indexes. ISBN 0-89391-273-5 (Cased): £37.95.” *Learned Publishing* 1(2):101–2. doi: 10.1002/leap/10034br3.

Soeters, Joseph L., Donna J. Winslow, and Alise Weibull. 2006. “Military Culture.” Pp. 237–54 in *Handbook of the Sociology of the Military*, *Handbooks of Sociology and Social Research*, edited by G. Caforio. New York, NY: Springer US.

de Waard, Erik, Peter de Bock, and Robert Beeres. 2019. “The Transaction Costs of Intra-Organizational Captive Buying and Selling Relationships.” *Journal of Accounting and Organizational Change* 15(2):257–77. doi: 10.1108/JAOC-07-2018-0058.

Westrum, Ron. 2014. “The Study of Information-flow: A Personal Journey.” *Safety Science* 67:58–63. doi: 10.1016/j.ssci.2014.01.009.

Whetsell, Travis A., Alexander Kroll, and Leisha DeHart-Davis. 2020. “Formal Hierarchies and Informal Networks: How Organizational Structure Shapes Information Search in Local Government.”

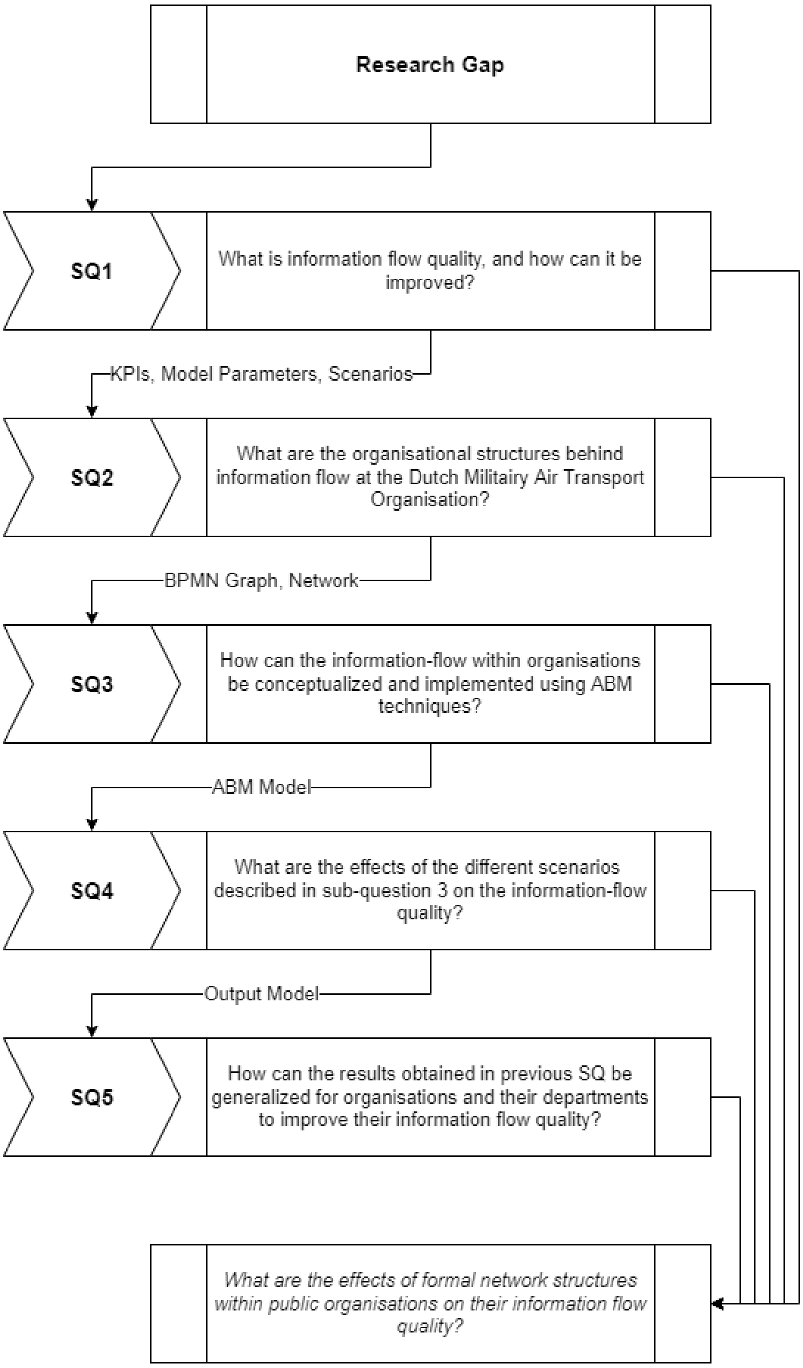
Willem, A., and M. Buelens. 2007. “Knowledge Sharing in Public Sector Organizations: The Effect of Organizational Characteristics on Interdepartmental Knowledge Sharing.” *Journal of Public Administration Research and Theory* 17(4):581–606. doi: 10.1093/jopart/mul021.

Yang, Tung Mou, and Terrence A. Maxwell. 2011. "Information-Sharing in Public Organizations: A Literature Review of Interpersonal, Intra-Organizational and Inter-Organizational Success Factors." *Government Information Quarterly* 28(2):164–75. doi: 10.1016/J.GIQ.2010.06.008.

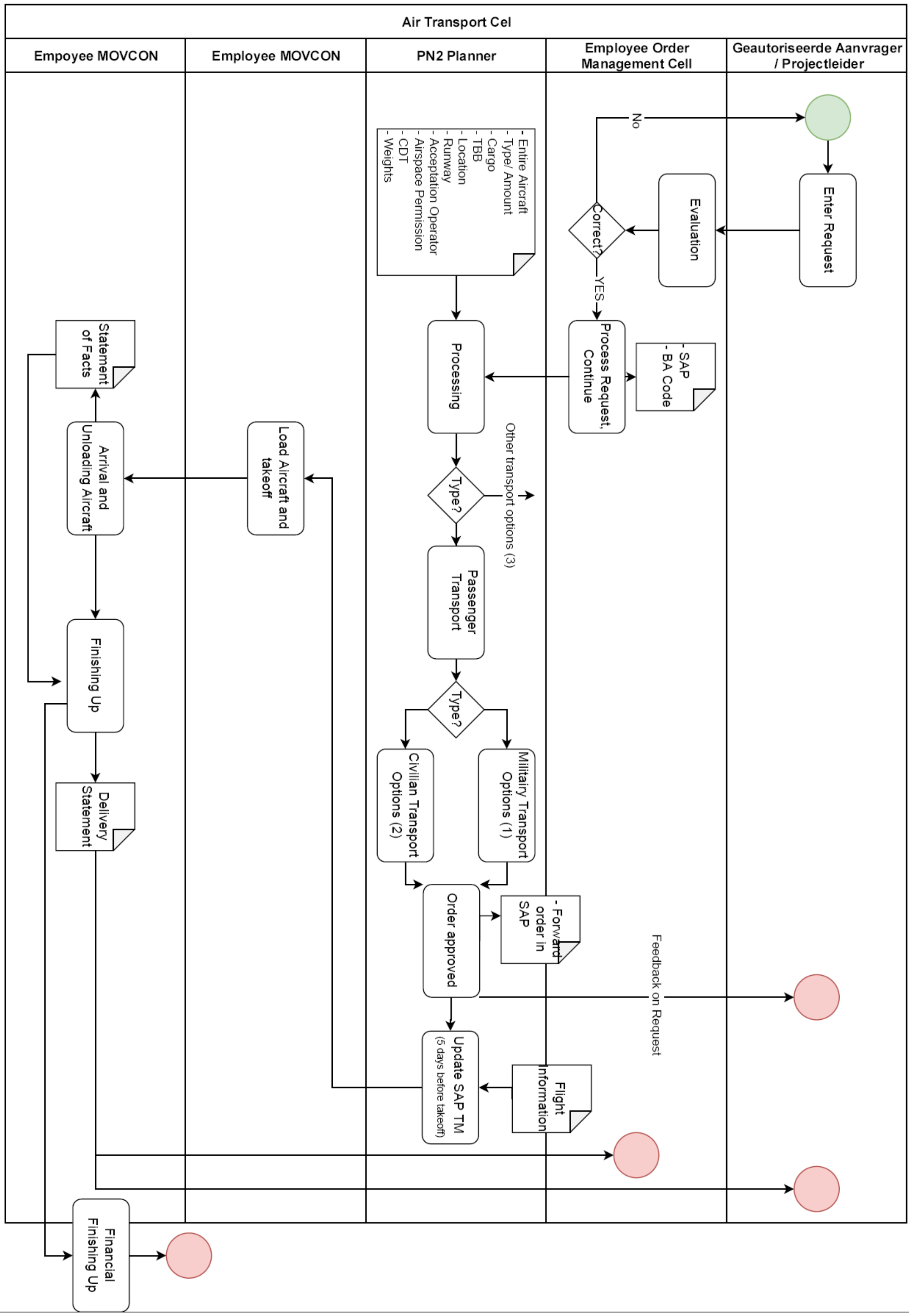
Yazici, Hulya Julie. 2002. "The Role of Communication in Organizational Change: An Empirical Investigation." *Information & Management* 39(7):539–52. doi: 10.1016/S0378-7206(01)00110-0.

Appendixes

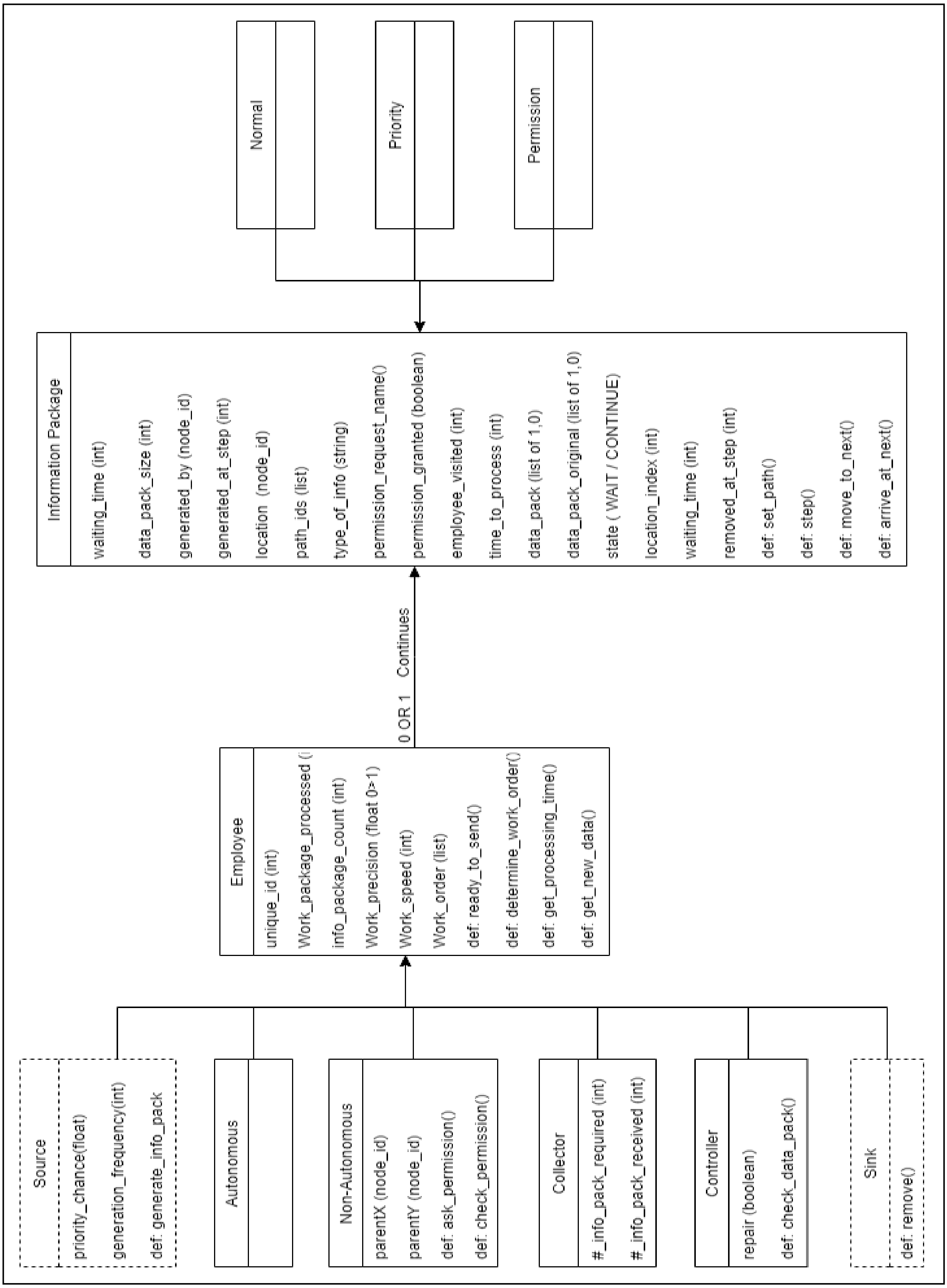
A: Research flow diagram



B: BPMN Diagram



C: UML Class Diagram



D: Verification

This Appendix chapter will describe the process of model verification that went on both during model developing and afterwards. Three different steps were used as model verification, a thorough code walkthrough, tracking agent behaviour and analysing and running a small sample model.

Code walkthrough

During the thorough code walkthrough, the structure in between classes the functions created and the implementation inside functions is checked whether they are at the right place in the code, perform the correct operations, change data that they have to change, and return the correct values of their function. Using print functions variables and variable types were analysed while the model was running. In the code itself, explanations in code and above classes and functions are given to why certain choices are made and what functions do. Verifying that results stored during the modelling process are correct.

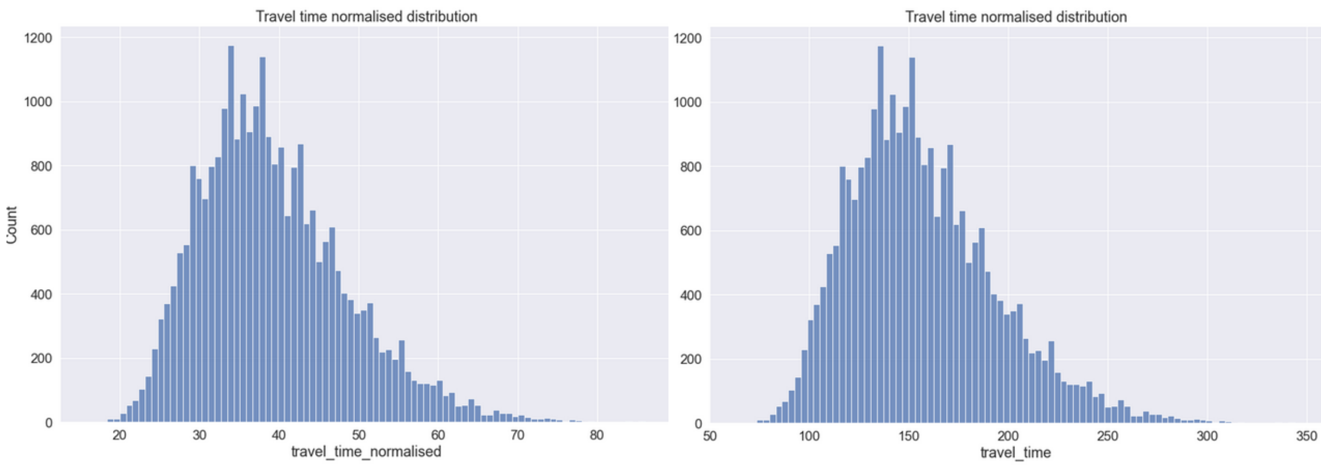
Tracking agent behaviour

The implementation of the non-autonomous employee agent asked for a lot of iterations to get it to function properly. One of the major ways of verification for this agent was using print functions to analyse agent behaviour. Whether the agent was correctly generating permission requests and if the path of these information packages was set correctly and in the correct format.

The next step was to track every agents behaviour while the model was running. Printing their location, waiting times and more, which allowed the modeller to track the interaction and movements of information packages. The results of tracking the agent behaviour showed that every information package moved from employee to employee and waits a set amount of time at that location before continuing. Printing the results of employee functions that are called while the model is running. Testing for each independent agent of they did what they were supposed to do.

Running small sample network

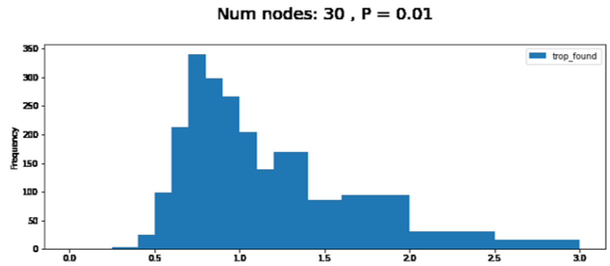
To test whether the model functions, the choice was made to use the case study network as input for the verification of the model. During this test run every agent is tested and if the model functions properly the results for travel time distribution should come close to the process time distribution from the case study. Showing similar shapes and minimum and maximum ranges that are within that reason. The figure below shows the travel time distributions of the test run of the small sample network. Comparing this to figure Figure 9 we see that the range of the steps is very similar. The minimum value in figure 9 is 30 minutes whereas the minimum value for a step in this network is around 20 minutes. This difference can be explained due to the slightly faster working employees that are possible. Since travel times are normalised the maximum range is probably smaller than could have been generated since the probability to generate the high values is small. These results, compared with tracking agent behaviour while this small sample network was running supported to conclusion that the model works as intended.

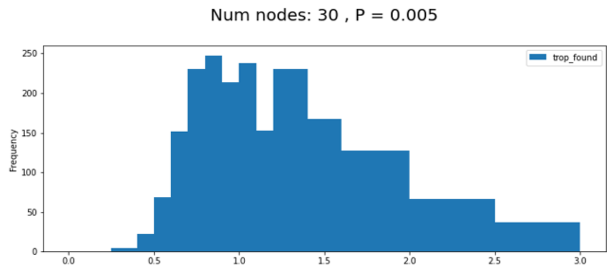


E: Network Generation Probability

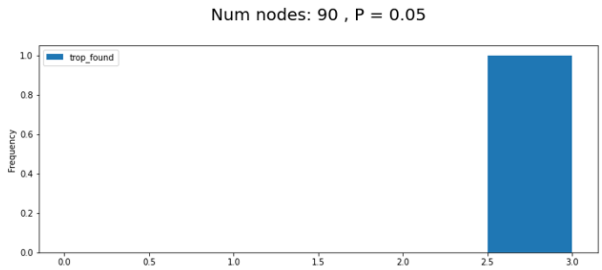
This Appendix chapter visualises the effect of the probability parameter for the network generation algorithm of NetworkX. Shown in the figures below are the tested network size and the probability used as input. The X axis shows the trophic incoherence values that can be generated using this q value and the network generation algorithm developed in this thesis. For network sizes of around 30 and 90 the value of p = 0.005 generates the widest range in q values. For networks around 270 nodes the value of p = 0.001 generates trophic incoherence values of 0.6. Lowering the probability even more had no actual effects as too many loose nodes were generated. These than had to be manually connected to others.

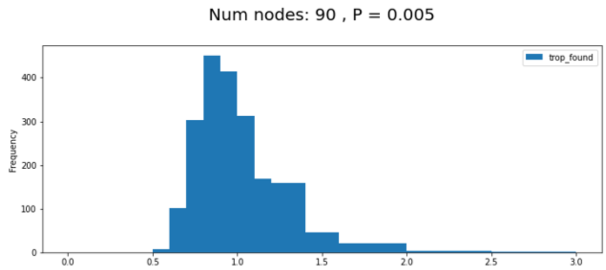
Network size 30



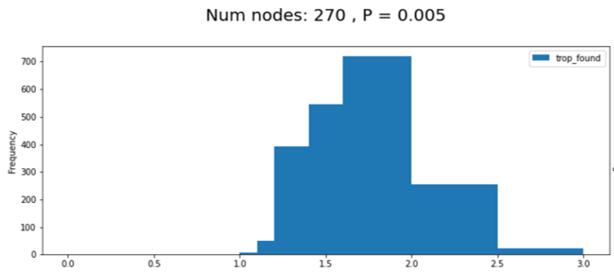


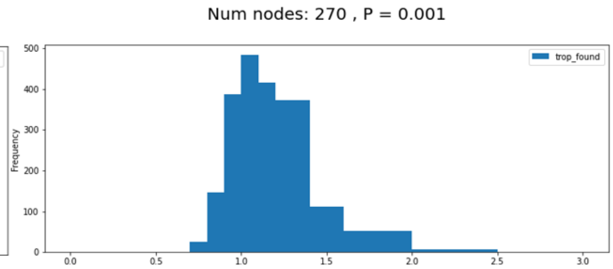
Network size 90





Network size 270





F: Network statistics

Scenario	Num. nodes	density	trophic incoherence	Avg. Shortest path	auto	non. auto	controller	collector
1	25	0.05	0.63	0.33	12.7	3	2.9	0.7
2	25	0.05	0.63	0.40	12.5	3.7	2.4	0.6
3	25	0.05	0.62	0.48	12.7	3.2	2.5	0.9
4	25	0.05	0.98	0.50	12.6	3.3	2.8	0.6
5	25	0.05	1.00	0.55	13.2	3.5	1.9	0.6
6	25	0.05	1.01	0.68	13.2	3.9	2	0.5
7	25	0.05	2.01	0.95	14.2	3.2	1.8	0.7
8	25	0.05	2.00	0.98	13.5	3.3	2.3	0.4
9	25	0.05	2.00	0.88	12.6	3.4	2.6	1
10	150	0.01	0.68	0.10	60.6	24.1	11.5	9.8
11	150	0.01	0.67	0.09	61.9	24.7	10.6	8.8
12	150	0.01	0.67	0.09	61.6	24.1	9.9	10.4
13	150	0.01	0.97	0.19	63	21.7	10	11.3
14	150	0.01	1.00	0.15	61.5	24.6	12.2	7.7
15	150	0.01	0.98	0.16	59.5	24.8	11.8	9.9
16	150	0.01	2.00	0.38	61	25.3	10.7	9
17	150	0.01	2.00	0.30	62.5	23.7	9.8	10
18	150	0.01	1.99	0.33	63.5	23.7	11.4	7.4
19	250	0.00	0.67	0.05	112.6	36.7	18.7	8.3
20	250	0.00	0.67	0.05	112.6	36.7	18.7	8.3
21	250	0.00	1.02	0.07	119.1	30.3	19.8	9
22	250	0.00	1.02	0.07	119.1	30.3	19.8	9
23	250	0.00	1.96	0.15	121.3	30.8	20.8	6.8
24	250	0.00	1.96	0.15	121.3	30.8	20.8	6.8
25	25	0.05	0.63	0.35	14.5	3.6	0	1.1
26	25	0.05	0.62	0.36	12.2	3.9	2.6	0.6
27	25	0.05	0.61	0.39	10	4.2	4.4	0.6
28	150	0.01	0.68	0.10	72.4	24.6	0	9
29	150	0.01	0.68	0.10	61.1	23.4	12	9.5
30	150	0.01	0.67	0.09	49.8	24.2	23.8	8.2

G: Distributions

All distributions are made using the pandas and seaborn library. Every effect of… on… is visualised in these graphs. They are all named and labelled with the scenarios that they show and what the effect of … on what is.

Travel Time

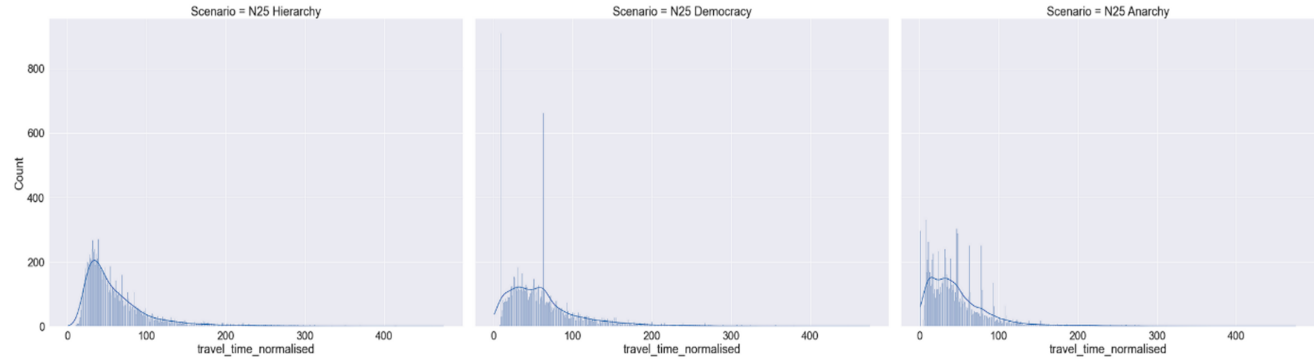


Figure 33: Normalised travel time distribution. Effect: Hierarchy. N = 25

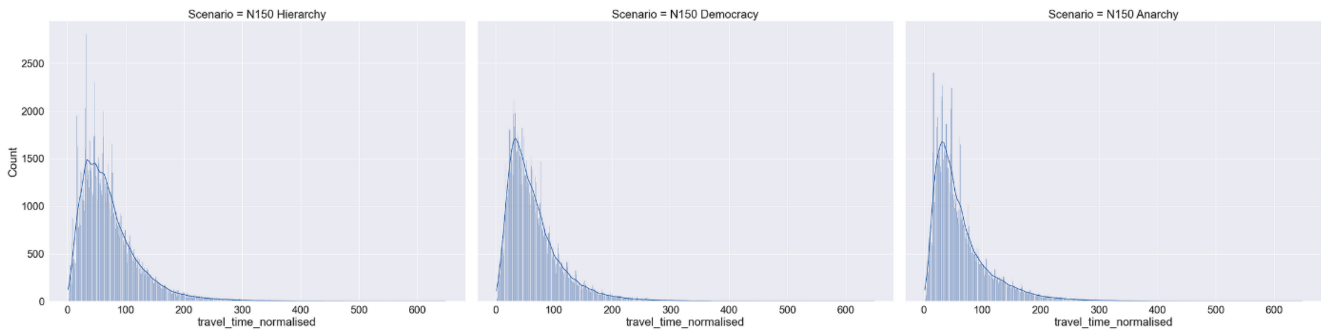


Figure 34: Normalised travel time distribution. Effect: Hierarchy. N = 150

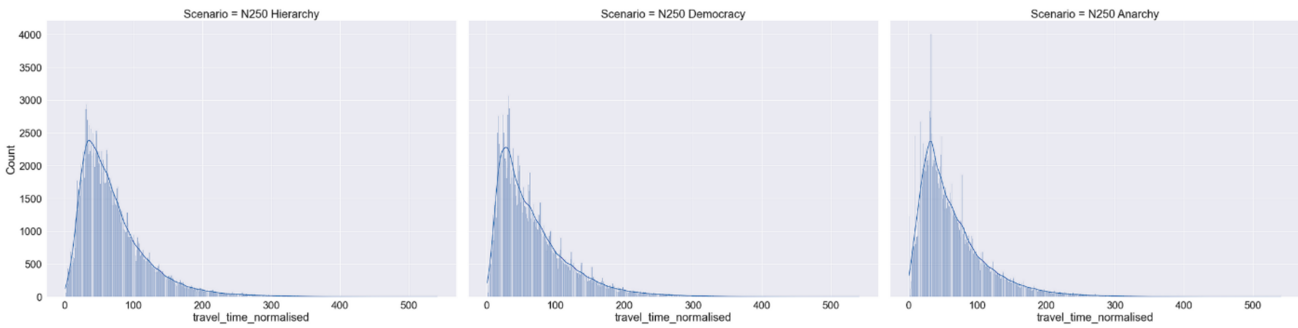


Figure 35: Normalised travel time distribution. Effect: Hierarchy. N = 250

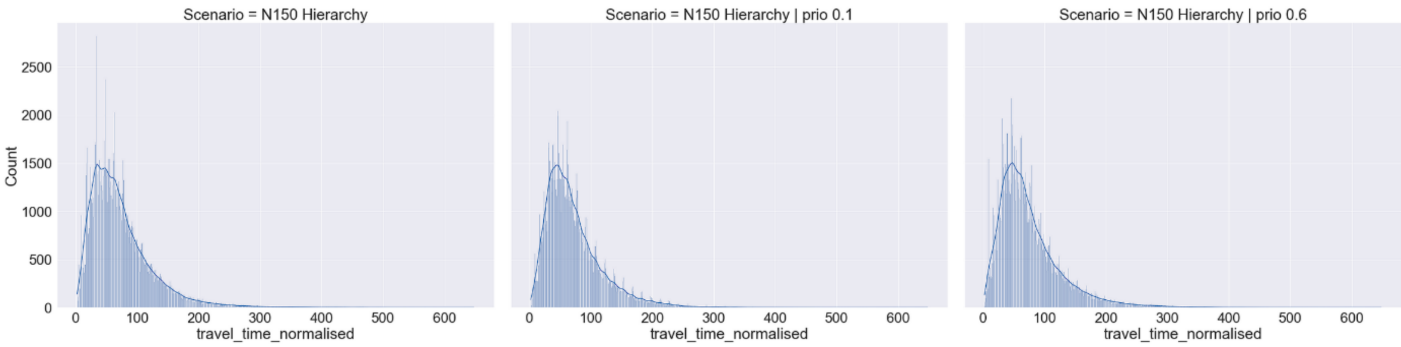


Figure 36: Normalised travel time distribution. Effect: Network size.

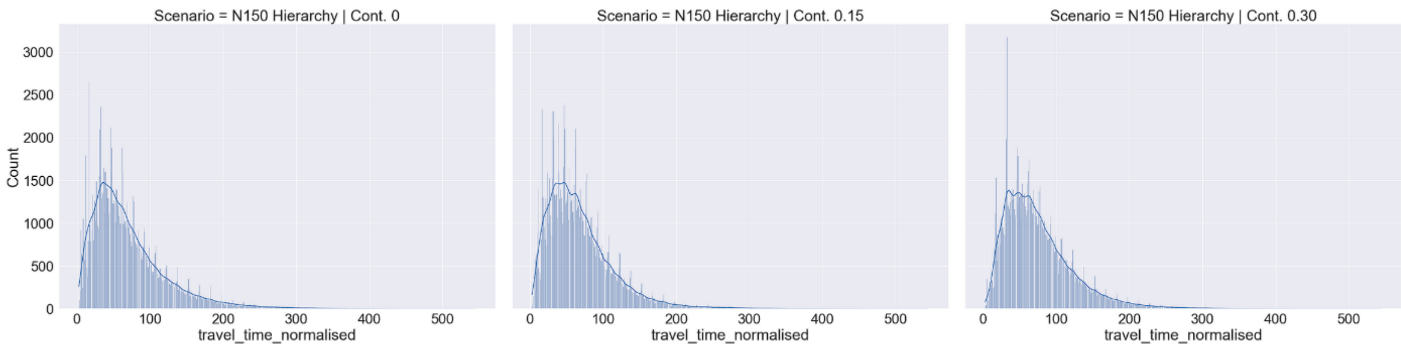


Figure 37: Normalised travel time distribution. Effect: Priority.

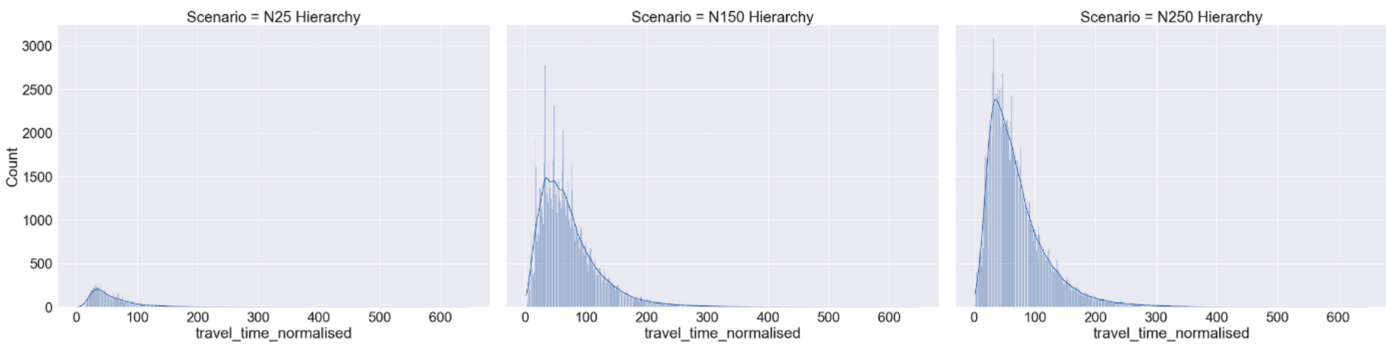


Figure 38: Normalised travel time distribution. Effect: Controller.

Controller

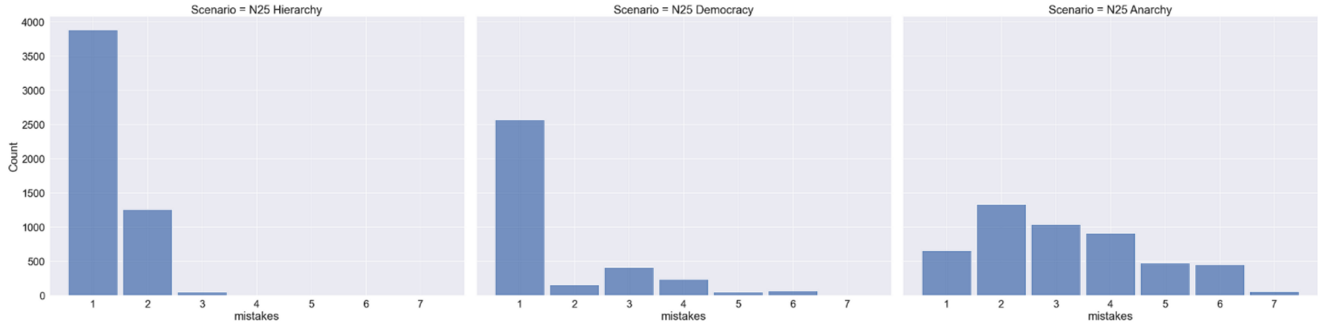


Figure 42: Mistakes caught by controllers. Effect: Hierarchy. N = 25

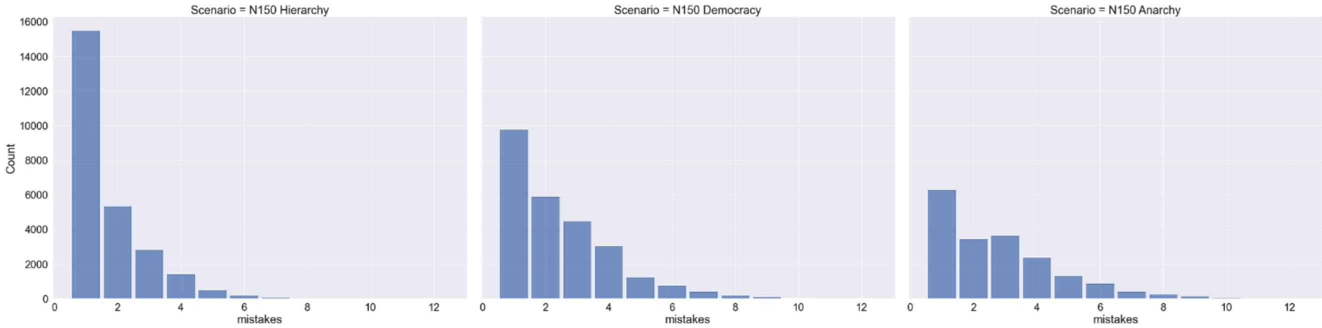


Figure 41: Mistakes caught by controllers. Effect: Hierarchy. N = 150

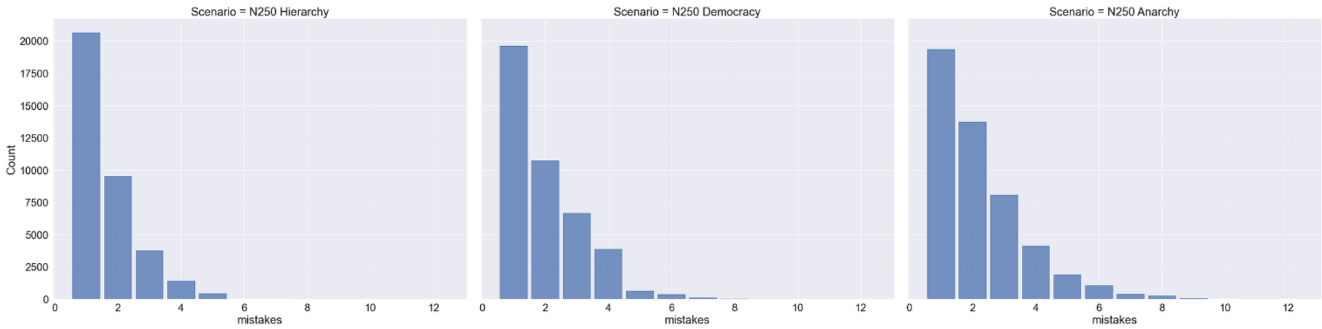


Figure 40: Mistakes caught by controllers. Effect: Hierarchy. N = 250

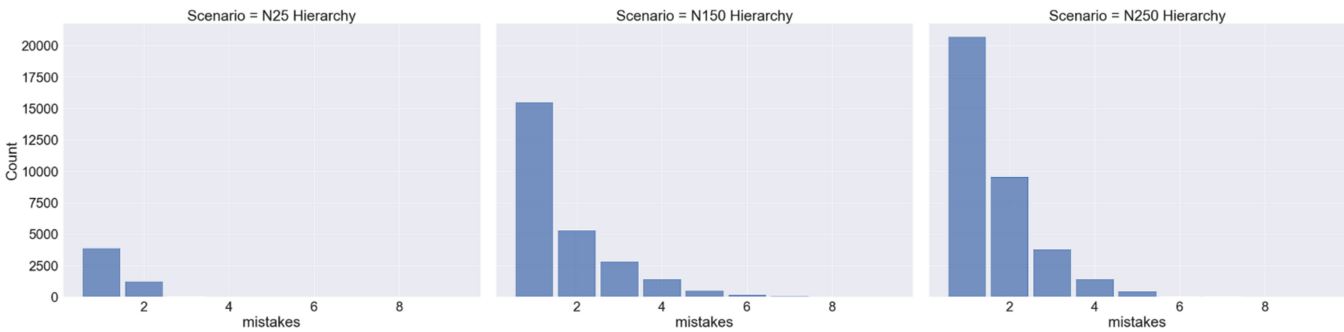


Figure 39: : Mistakes caught by controllers. Effect: Network size

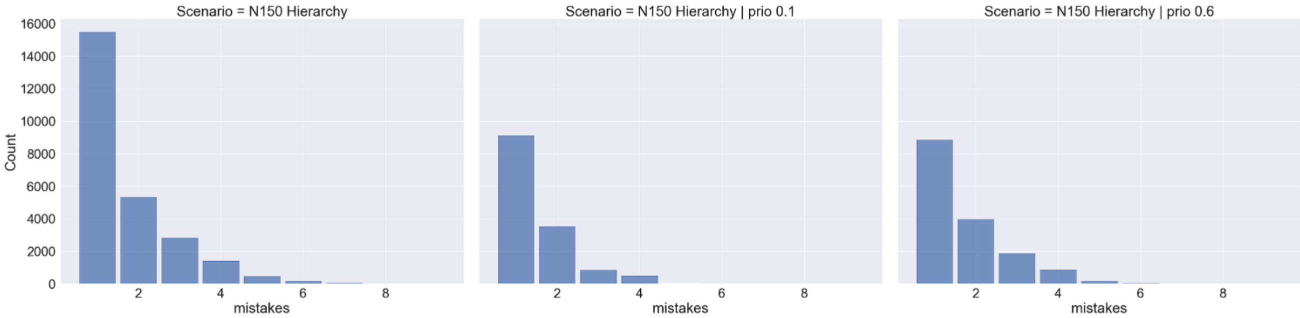


Figure 43: Mistakes caught by controllers. Effect: Priority

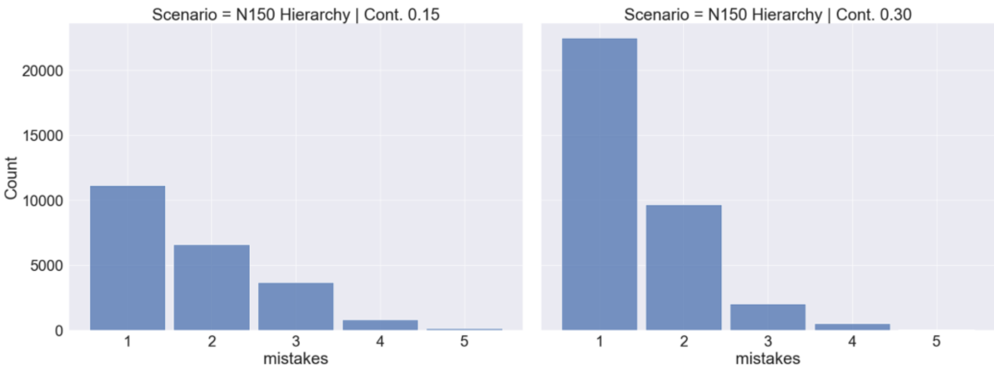


Figure 44: Mistakes caught by controllers. Effect: Controller.

Information pack data

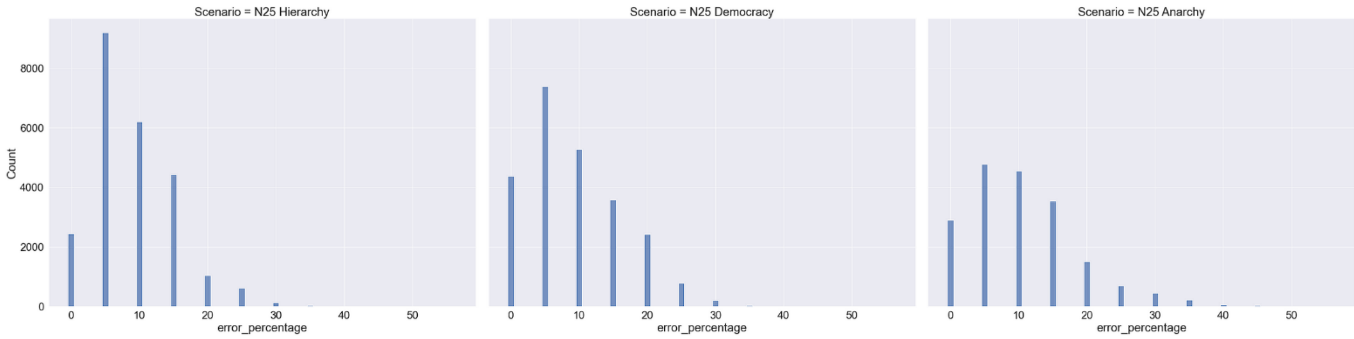


Figure 46: Error percentage of information package. Effect: Hierarchy. N = 25

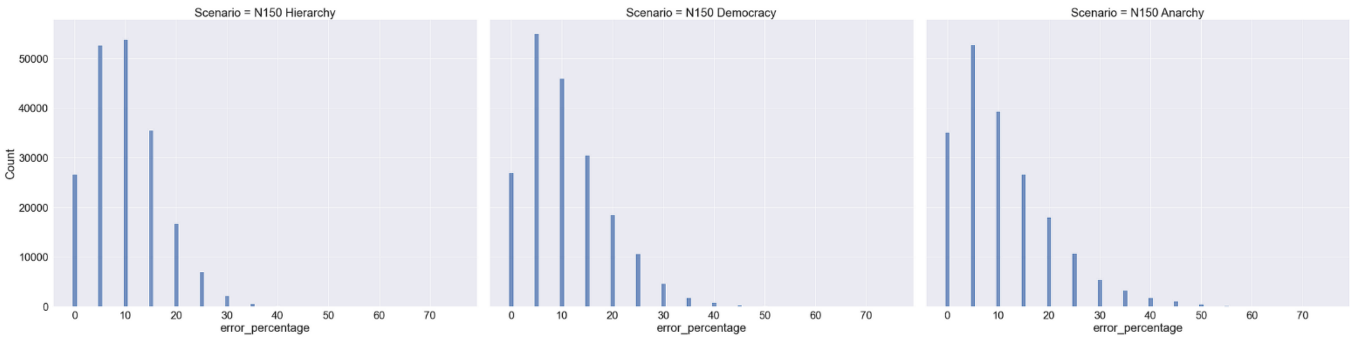


Figure 45: Error percentage of information package. Effect: Hierarchy. N = 150

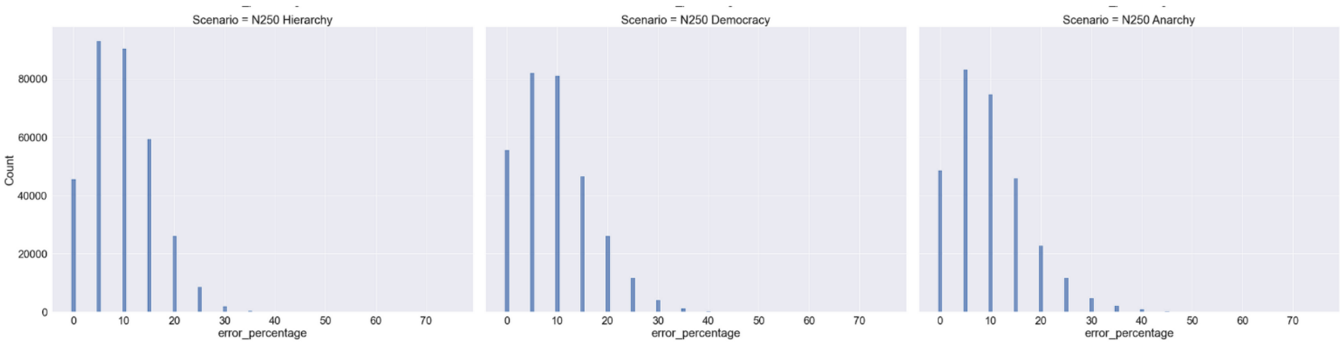


Figure 47: Error percentage of information package. Effect: Hierarchy. N = 250

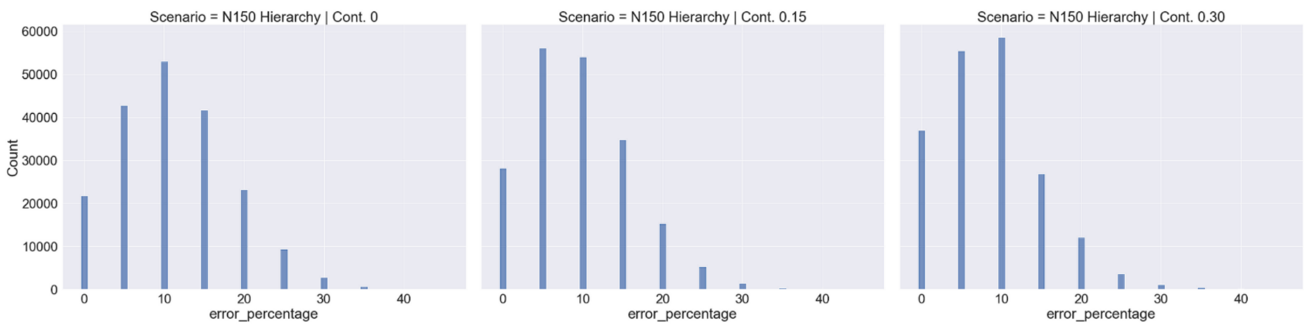


Figure 48: Error percentage of information package. Effect: Controller

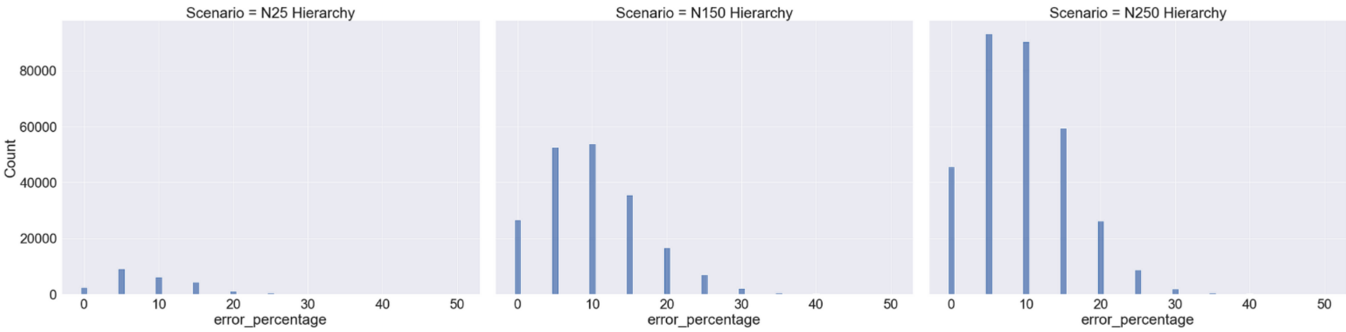


Figure 50: Error percentage of information package. Effect: Networksize

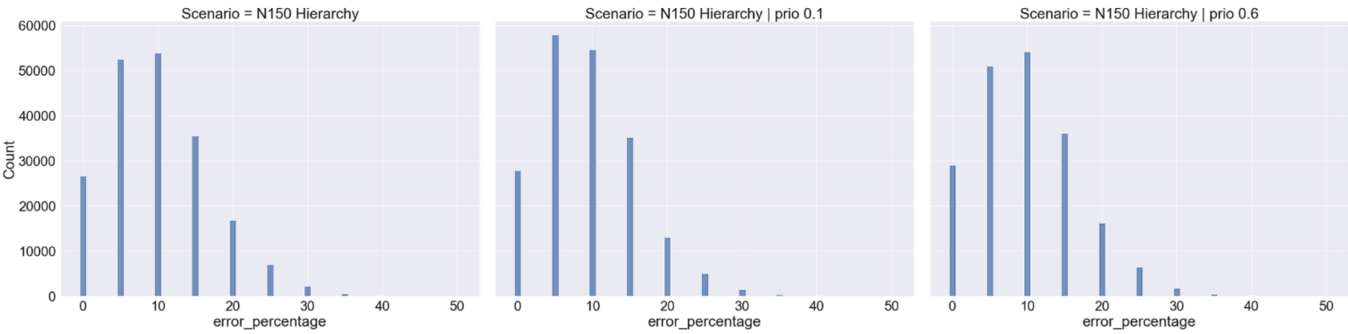


Figure 49: Error percentage of information package. Effect: Priority

Busyness

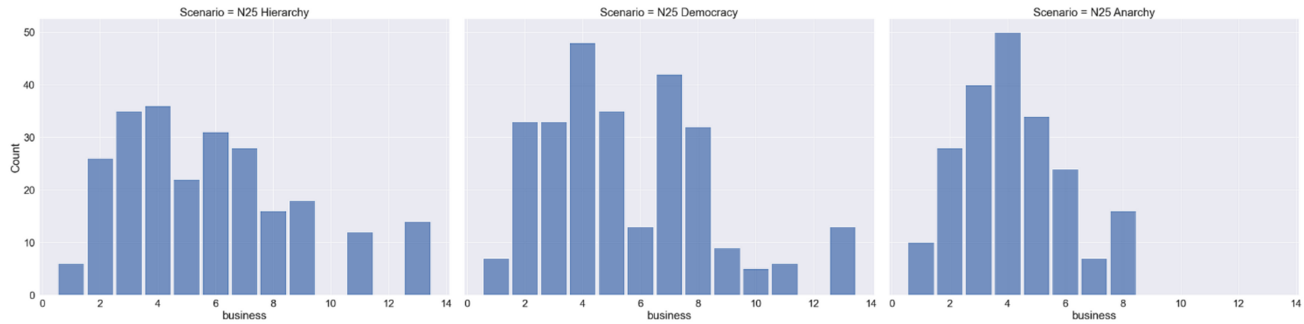


Figure 53: Busyness of employees at final step. Effect: Hierarchy. N = 25

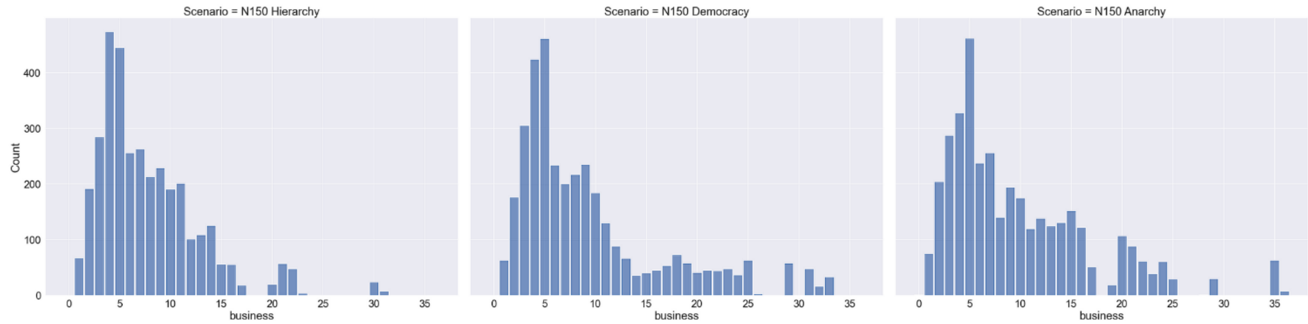


Figure 52: Busyness of employees at final step. Effect: Hierarchy. N = 150

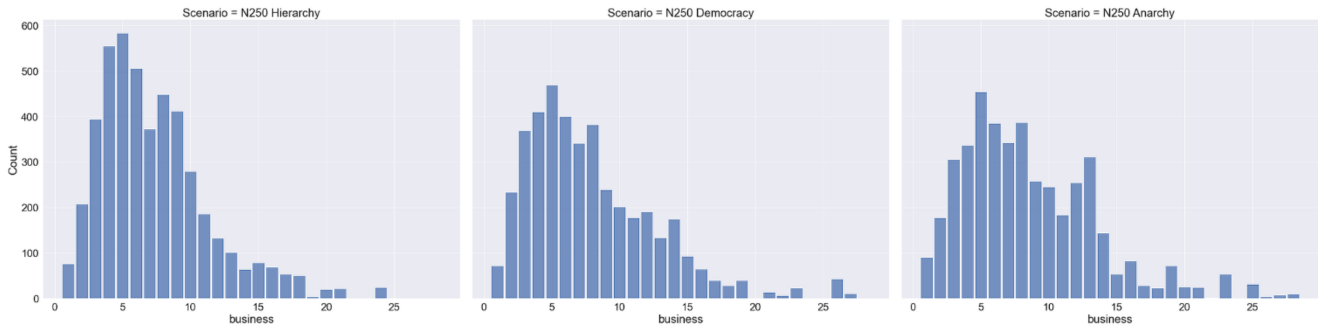


Figure 51: Busyness of employees at final step. Effect: Hierarchy. N = 250

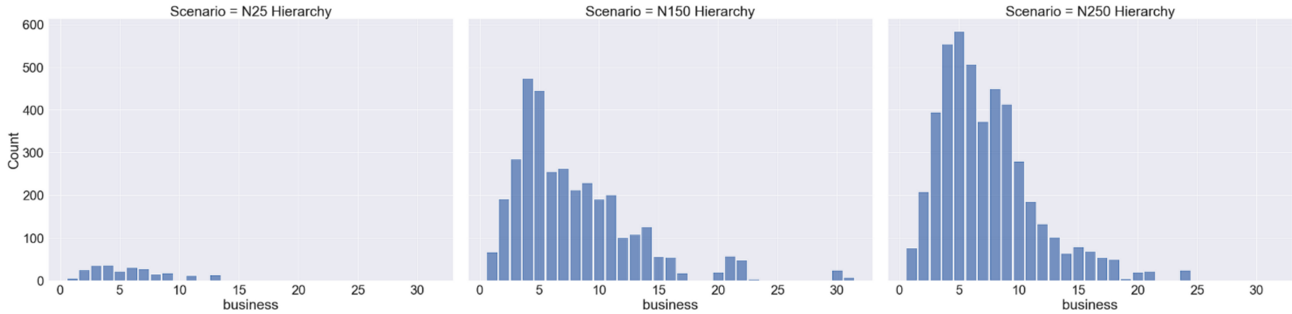


Figure 56: Busyness of employees at final step. Effect: Networksize

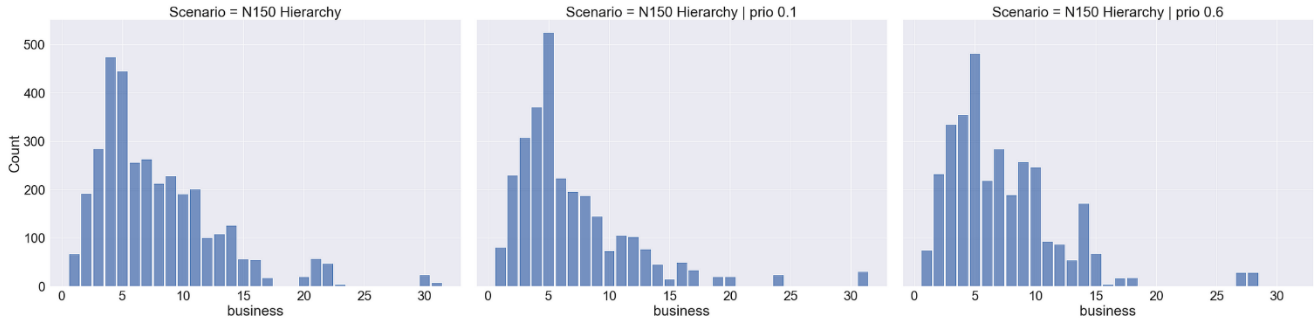


Figure 55: Busyness of employees at final step. Effect: Priority

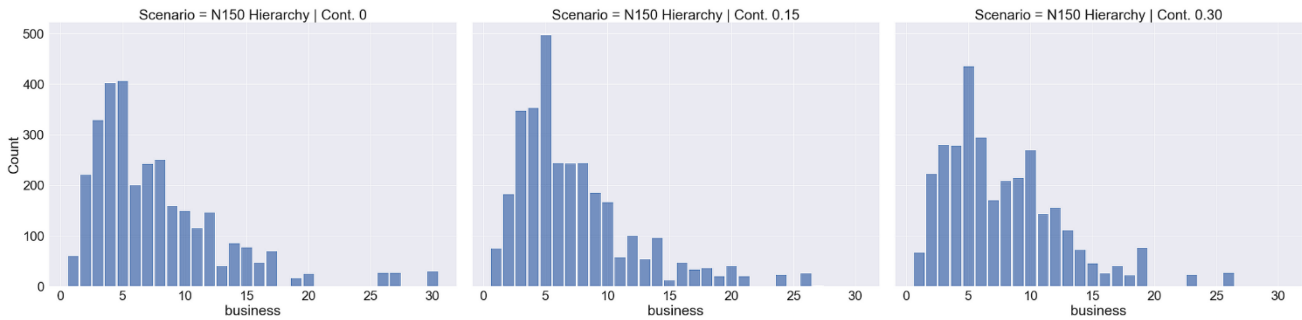


Figure 54: Busyness of employees at final step. Effect: Controller

H: Network correction

A peak in very low normalised travel time is visible and is orders of magnitude more than the peak of the expected travel times. For clarity and transparency, this is shown before the results are discussed. The choice was made to leave out these peaks since the model verification showed that the model behaviour was correct, and they are an outlier effect from the random networks. Figure 57 and Figure 58 show the peak with low normalised travel times and after it is removed. The peak was present between travel time 0 and travel time 10. After removing these outliers, Figure 58 still shows that there are many outliers in the data. With visual inspection an outlier can be seen around 25 minutes travel time. Furthermore, the distribution of travel time over the model runs from as low as 0, without correction with a maximum travel time of 2234 minutes. Since these high values are outliers, the plot is cut-off at a point where the counts are no longer visible. Another correction in the raw collected data is for the number of employees visited. In the collection method there was a problem where one employee was counted too many.

Every scenario that was run collected data on the theoretical networks generated. The results are shown in Appendix F. The results are not used for determining information-flow quality, but when discussing the results, help compare outcomes from different scenarios.

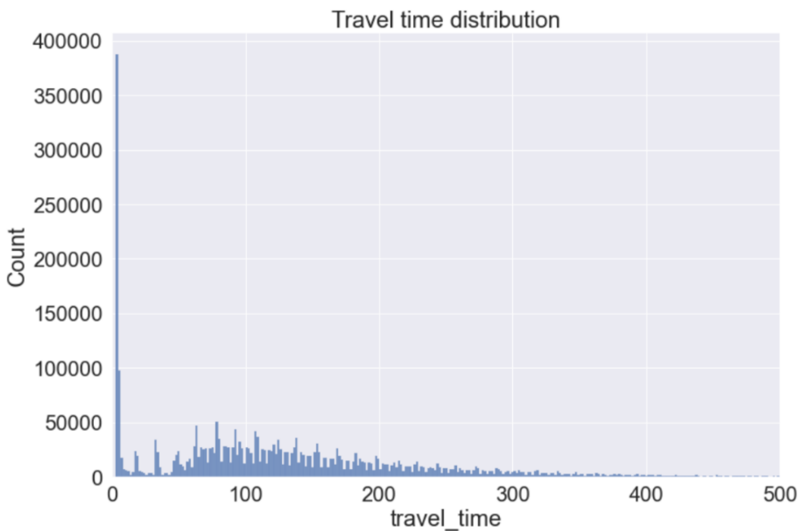


Figure 57: Case study | Distribution of travel time

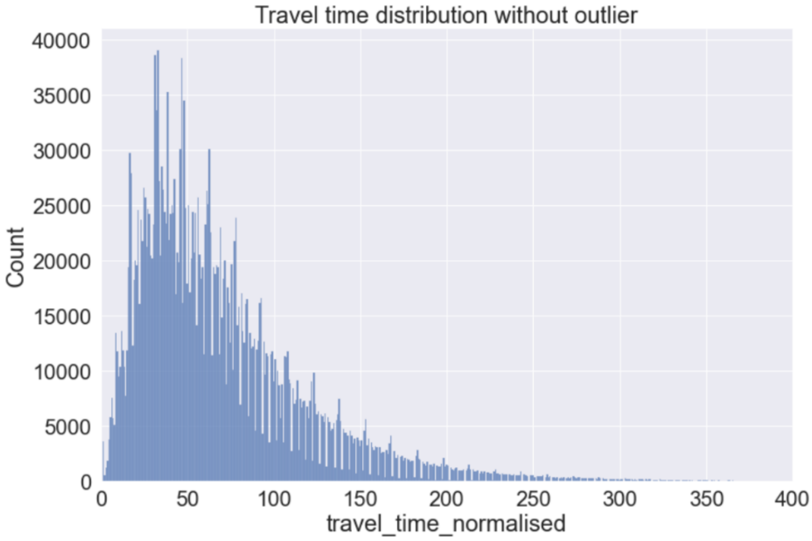


Figure 58: Distribution of normalised travel time without the visible outliers below travel time 10

I: KS Tables of distributions

In order to statistically verify the results of the experiments, check whether the results, differences between distributions, is not due to randomness, Two Sample Kolmogorov-Smirnov Tables were made for each of the datasets. Where every scenario was compared with each of the other scenarios. Values higher than 0.5 where rejected (coloured in red). Values of p=0 are visible in the tables, this is due to the minimum python notation, and could not be changed.

	N25 Hierarchy	N25 Democracy	N25 Anarchy	N150 Hierarchy	N150 Democracy	N150 Anarchy	N250 Hierarchy	N250 Democracy	N250 Anarchy	N150 Hierarchy | prio 0.1	N150 Hierarchy | prio 0.6	N150 Hierarchy | Cont. 0.15	N150 Hierarchy | Cont. 0.30
temp													
N25 Hierarchy	1	2.56E-81	1	2.41E-133	0	0	9.66E-123	0	0	1.84E-34	3.58E-125	4.83E-233	5.01E-41
N25 Democracy	2.56E-81	1	0.338562	4.32E-52	0	0	1.28E-74	1.13E-211	0	1.24E-34	2.86E-80	4.85E-151	1.72E-58
N25 Anarchy	1	0.338562	1	0	4.83E-215	8.22E-138	0	0	1.75E-281	0	0	0	0
N150 Hierarchy	2.41E-133	4.32E-52	0	1	0	0	3.85E-16	1.43E-250	0	3.69E-73	5.22E-14	1.11E-103	5.03E-193
N150 Democracy	0	0	4.83E-215	0	1	2.67E-72	0	1.80E-189	4.21E-70	0	0	0	0
N150 Anarchy	0	0	8.22E-138	0	2.67E-72	1	0	0	6.44E-289	0	0	0	0
N250 Hierarchy	9.66E-123	1.28E-74	0	3.85E-16	0	0	1	3.59E-244	0	7.36E-49	2.44E-09	1.16E-67	4.04E-118
N250 Democracy	0	1.13E-211	0	1.43E-250	1.80E-189	0	3.59E-244	1	2.05E-102	7.76258e-311	6.87E-89	2.10E-83	0
N250 Anarchy	0	0	1.75E-281	0	4.21E-70	6.44E-289	0	2.05E-102	1	0	2.42E-288	5.44E-201	0
N150 Hierarchy | prio 0.1	1.84E-34	1.24E-34	0	3.69E-73	0	0	7.36E-49	7.76258e-3	0	1	6.77E-55	7.45E-168	1.66E-06
N150 Hierarchy | prio 0.6	3.58E-125	2.86E-80	0	5.22E-14	0	0	2.44E-09	6.87E-89	2.42E-288	6.77E-55	1	4.14E-29	1.66E-132
N150 Hierarchy | Cont. 0.15	4.83E-233	4.85E-151	0	1.11E-103	0	0	1.16E-67	2.10E-83	5.44E-201	7.45E-168	4.14E-29	1	3.50E-263
N150 Hierarchy | Cont. 0.30	5.01E-41	1.72E-58	0	5.03E-193	0	0	4.04E-118	0	0	1.66E-06	1.66E-132	3.50E-263	1

Figure 59: KS diagram controller data

	N25 Hierarchy	N25 Democracy	N25 Anarchy	N150 Hierarchy	N150 Democracy	N150 Anarchy	N250 Hierarchy	N250 Democracy	N250 Anarchy	N150 Hierarchy | prio 0.1	N150 Hierarchy | prio 0.6	N150 Hierarchy | Cont. 0.15	N150 Hierarchy | Cont. 0.30
N25 Hierarchy	1	1.29E-68	6.20E-72	1.20E-112	1.11E-240	0	1.19E-66	7.73E-121	2.89E-101	3.21E-37	9.53E-101	2.74E-52	7.59E-145
N25 Democracy	1.29E-68	1	2.73E-57	9.02E-129	7.08E-89	2.09E-109	1.83E-79	4.41E-37	7.07E-37	1.07E-46	5.00E-116	1.97E-63	5.61E-77
N25 Anarchy	6.20E-72	2.73E-57	1	6.16E-13	9.23E-14	4.58E-42	6.96E-38	6.28E-46	1.38E-31	3.17E-64	2.57E-18	5.62E-44	9.32E-215
N150 Hierarchy	1.20E-112	9.02E-129	6.16E-13	1	4.71E-230	0	9.99E-47	2.20E-197	1.09E-160	1.13E-113	5.74E-13	4.78E-54	0
N150 Democracy	1.11E-240	7.08E-89	9.23E-14	4.71E-230	1	4.72E-152	0	3.16E-282	6.90E-187	0	3.47107e-313	0	0
N150 Anarchy	0	2.09E-109	4.58E-42	0	4.72E-152	1	0	0	0	0	0	0	0
N250 Hierarchy	1.19E-66	1.83E-79	6.96E-38	9.99E-47	0	0	1	1.22E-224	2.17E-162	2.55E-28	1.48E-24	0.0011193	0
N250 Democracy	7.73E-121	4.41E-37	6.28E-46	2.20E-197	3.16E-282	0	1.22E-224	1	2.23E-32	2.72E-168	3.26E-130	9.64E-136	0
N250 Anarchy	2.89E-101	7.07E-37	1.38E-31	1.09E-160	6.90E-187	0	2.17E-162	2.23E-32	1	1.92E-206	2.51E-128	5.90E-111	0
N150 Hierarchy | prio 0.1	3.21E-37	1.07E-46	3.17E-64	1.13E-113	0	0	2.55E-28	2.72E-168	1.92E-206	1	3.61E-81	6.49E-15	1.34E-252
N150 Hierarchy | prio 0.6	9.53E-101	5.00E-116	2.57E-18	5.74E-13	.47107e-313	0	1.48E-24	3.26E-130	2.51E-128	3.61E-81	1	1.77E-36	0
N150 Hierarchy | Cont. 0.15	2.74E-52	1.97E-63	5.62E-44	4.78E-54	0	0	0.001119	9.64E-136	5.90E-111	6.49E-15	1.77E-36	1	0
N150 Hierarchy | Cont. 0.30	7.59E-145	5.61E-77	9.32E-215	0	0	0	0	0	0	1.34E-252	0	0	1

Figure 60: KS of info. error distributions

<table>
<tr><th></th><th>N25 Hierarchy</th><th>N25 Democracy</th><th>N25 Anarchy</th><th>N150 Hierarchy</th><th>N150 Democracy</th><th>N150 Anarchy</th><th>N250 Hierarchy</th><th>N250 Democracy</th><th>N250 Anarchy</th><th>N150 Hierarchy | prio 0.1</th><th>N150 Hierarchy | prio 0.6</th><th>N150 Hierarchy | Cont. 0.15</th><th>N150 Hierarchy | Cont. 0.30</th></tr>
<tr><td>N25 Hierarchy</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N25 Democracy</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N25 Anarchy</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N150 Hierarchy</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N150 Democracy</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N150 Anarchy</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N250 Hierarchy</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N250 Democracy</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N250 Anarchy</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N150 Hierarchy | prio 0.1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>N150 Hierarchy | prio 0.6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>N150 Hierarchy | Cont. 0.15</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>N150 Hierarchy | Cont. 0.30</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>

Figure 61: KS of timeliness

<table>
<tr><th></th><th>N25 Hierarchy</th><th>N25 Democracy</th><th>N25 Anarchy</th><th>N150 Hierarchy</th><th>N150 Democracy</th><th>N150 Anarchy</th><th>N250 Hierarchy</th><th>N250 Democracy</th><th>N250 Anarchy</th><th>N150 Hierarchy | prio 0.1</th><th>N150 Hierarchy | prio 0.6</th><th>N150 Hierarchy | Cont. 0.15</th><th>N150 Hierarchy | Cont. 0.30</th></tr>
<tr><td>N25 Hierarchy</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>N25 Democracy</td><td>0.694454</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>N25 Anarchy</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>N150 Hierarchy</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>N150 Democracy</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>N150 Anarchy</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>N250 Hierarchy</td><td>0.000001</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>N250 Democracy</td><td>0</td><td>0</td><td>0</td><td>0.000095</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>N250 Anarchy</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>N150 Hierarchy | prio 0.1</td><td>0.016481</td><td>0.00009</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>N150 Hierarchy | prio 0.6</td><td>0.000057</td><td>0</td><td>0</td><td>0.000002</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td></tr>
<tr><td>N150 Hierarchy | Cont. 0.15</td><td>0.000777</td><td>0</td><td>0</td><td>0.000002</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0.000028</td><td>0.000494</td><td>1</td><td></td></tr>
<tr><td>N150 Hierarchy | Cont. 0.30</td><td>0</td><td>0</td><td>0</td><td>0.084031</td><td>0</td><td>0</td><td>0</td><td>0.079912</td><td>0</td><td>0</td><td>0.000001</td><td>0</td><td>1</td></tr>
</table>

Figure 62: Effect on busyness