Developing a risk communication and decision making modelling framework for system designers

Daniel John Nesbitt

School of Computing Science, Newcastle University

Thesis submitted for the degree of Doctor of Philosophy

November 2017

This thesis is dedicated to my friends and family.

Acknowledgements

I would like to thank all of the participants who took part in the studies described in this thesis.

I would like to give my gratitude to Dr Darren Flynn, Institute of Health and Society, Newcastle University for all of his help and support over the years.

I would also like to thank Professor Richard Thomson and Dr Peter McMeekin their help and support throughout my time working on this thesis.

I would also like to thank James Turland and Kanidia Sinami for their help in developing the scenarios used in this thesis.

Finally I would like to thank my supervisor, Professor Aad van Moorsel for his unwavering help, support and assistance throughout my PhD and beyond.

Disclaimer

This thesis presents independent research commissioned by the National Institute for Health Research (NIHR) under its Programme Grants for Applied Research scheme (RP-PG-0606-1241). The views expressed in this publication are those of the authors and are not necessarily those of the NHS, the NIHR, or the Department of Health.

Abstract

Risk communication defines the process of exploring the likelihood of an understandable event occurring. Shared decision making is the process where participants of different levels of expertise discuss and come to an agreed decision. A shared decision making scenario may involve risk communication. Depending on the scenario, a decision support tool may be desired to help visualise, communicate and potentially reduce the likelihood of an adverse event occurring.

As part of the development process for decision support tools, the developer should understand the problem domain that the support tool aims to resolve. One approach for understanding the problem domain is to develop a workflow model that visualises the activities and conditions that a decision making scenario has.

While decision making processes and user interactions can be well discovered by using data elicitation methods, subjective data from the elicitation process poses challenges when attempting to use such data in a workflow model. For example, the order activities a person does may be inconsistent, or subject to change depending on their subjective preferences.

This thesis discusses the development of tools and strategies for producing workflow models of decision making activities with qualitative and subjective data. Firstly, we discuss the development of BPMNdm (Business Process Modelling Notation – decision making), an extended workflow modelling notation to support the encapsulation of decision making activities. Secondly, we discuss how qualitative data produced using ethnographic method like participant observation can be transformed into a set of rules to encapsulate low level interaction activities between participants. Thirdly, we discuss how subjective data can be expressed as a triangular fuzzy number, which preserves the subjective nature of the data while allowing the data to be used to control the workflow model.

Contents

List of figures

List of tables

Chapter 1 - Introduction

In this thesis, we consider risk to be the probability of an undesirable event occurring, for example; financial loss, missed coursework deadlines and personal injury. How undesirability is measured is based upon a person's own preference and perceptions, which may be influenced by the preferences of other people. Risk is a concern for individuals, the wider public, business and governmental groups. Each of these groups will perceive risks based upon their own criteria, and what may be considered a risk for one group, may not be perceived as that by another. The optimal methods to inform people of the risk, as well as the impact of undesirable events are subjects of ongoing discussion [1], [2].

While the ultimate consequences of an understandable event occurring are relatively static, the types of possible risks that lead to the event are fluid. For example, when considering historical and current medical conditions that pose a significant risk of death or serious medical complications, in the medieval period, the plague was a potentially fatal and commonplace disease [3], [4], whereas in the 21^{st} Century cancers are growing increasingly commonplace [5], [6]. Likewise, the introduction of internet banking and Near Field Communication (NFC) have offered new ways for criminals to steal money. The fluidity of the types of risks means that there is a constant requirement to manage or mitigate these risks.

However, looking further into risk, an important consideration is a person's perception and attitude to risk [7]. Considering the increase in cancers, the actual reasons for the increased risk of a person getting a type of cancer are due to multiple indicators. One of the most significant risk factors is age, and as life expectancy increases, more people contract cancer [6]. Therefore, while likelihood of a person getting cancer has increased over time, it can be inferred that the risk is linked to a positive factor, the increase of life expectancy. This underlines the importance of sufficiently informing people about the actual risk factors – the attributes or characteristics that increase or decrease a risk.

Developing tools and strategies to reliably communicate with people about the probability of risk is a broad area drawing on the knowledge and expertise of psychologists, medical and safety professionals, system designers and non-professional people. These groups can be considered the stakeholders when developing solutions for informing, managing and reducing the risk.

Depending on the problem domain, a risk-communication tool may be used to assist the stakeholders to come to a decision on how to deal with the risk. When discussing the development of risk-communication principles, Fischhoff emphasised the importance of engaging with the stakeholders as part of the development strategy necessary for creating risk-communication tools [8]. It is considered good practice when developing these tools to engage with the potential users in order to elicit contextual and user requirements for such tools [9], [10].

In essence, a risk communication tool should be able to convey the risk it was designed to communicate in a manner that is understandable and accessible to all stakeholders and be based on a robust, accurate evidence base.

If the risk-commutation tool is computer based, the reliability, robustness and correctness of underlying prediction algorithms should be verified with the aid of testing and analysis. However, another important factor is the usability – to ensure the device is usable by the target user group. Usability of a device can be ascertained as part of the evaluation phase of the system development cycle with the intended stakeholders. Adopting the principles of the general software engineering process, which specifies how a software product should be developed, documented and evaluated, consideration of the ultimate design of the system should begin before development begins, and usually involves a range of ethnographic techniques, such as participant observations and surveys, to gather user and system requirements The generation of requirements in turn determines the principle requirements (functional) and technical (non-functional) requirements of the proposed system. There are many ethnographic methodologies for collecting qualitative or quantitative data, such as interviews, observations and workshops with the intended end users of the tool being developed or the intended user demographic [9].

The ultimate goal for a software or hardware development process is the production of a usable and appropriate tool for the problem domain. The usability of the tool is dependent on the knowledge of the user and system requirements as part of the requirements elicitation process, and ensuring the finished tool fulfils those requirements. For example, during the development of a risk communication tool to show the treatment risk of the drug Alteplase for treating ischemic strokes [11], as part of the requirements elicitation phase of development, stroke patients were asked to indicate their preferences for how the risk and benefits of the drug should be presented. The patients interviewed typically preferred seeing graphical presentations such as pictograms to show the potential risks and benefits of taking the drug. Clinicians indicated that the tool should be portable, to allow the clinician to explain what the graphics meant and it should be possible to rapidly enter information. During the course of development, clinicians also expressed a preference for a tool that could be used on their own devices, which would not be used as part of the patient consultation process, but rather as a method for quickly discovering the risks and benefits of the treatment before the clinician would see the patient. The two different versions of the risk commutation tool developed as part of this research are shown in Figures 1 and 2. These figurers show the salient differences between the two versions, with the differences being requested by the clinicians during the development phase.

Figure 1: Risk Communication Tool COMPASS® for Stroke Patients - Tablet PC version

Figure 2: Risk Communication Tool COMPASS® for Stroke Patients - Smartphone version

Figure 1 shows the version developed for communicating the risks and benefits of treatment to patients. The salient features are graphical risk presentations showing the risks and benefits of treatment and simplified textual phrases that do not include any medical terminology. Figure 2 shows the version developed for smartphones. As the smartphone version is intended for use by the clinicians only, the textual descriptions used the medical terms appropriate for clinicians to understand. As the interface was designed for smaller screens, linear navigation was used to allow the clinician to input the patient's details, view a table summary of the risks and benefits of treatment and then to see a risk presentation of the same data as separate screens.

During the course of eliciting requirements of the stroke risk communication tool, certain aspects of the problem domain remained static or deterministic, such as the location where the tool would be used, the need for rapid data entry as well as the types of user who would be using the tool. However, some aspects of the problem domain were less certain, such as characteristics of the individual patients such as if they were well enough to engage with the clinician, if the patient and/or family member wished to make the finial decision if they should have the drug, or delegate that decision to the clinician.

In order to clarify these issues, additional user interviews were undertaken, with the aim of clarifying the less certain aspects of the problem domain, particularly to ascertain peoples' attitudes to being involved in the decision making process, and if the risk communication tool should form part of that process. In this kind of situation, having methodologies to explore these issues with stakeholders, and to verify and analyse any assumptions made of the problem domain would be a useful asset in assisting in the development of a risk communication tool. Furthermore, such a methodology could prompt the tool developers on what tool design features could be appropriate, based on the assumptions made on the static requirements, as well as the less certain aspects, once the stakeholders had verified that the assumptions made were appropriate. Workflow models are one such methodology, where the tasks, people and any relevant attributes such as documents are represented as a set of graphical nodes with arcs that define the relationship of each node (such as the progression from one task node to a second task).

1.1 Aim

The aim of this research is to develop and describe a set of methodologies to develop workflow models tailored to describing decision making scenarios. The methodologies should indicate the necessary steps to tailor the workflow formalism to support decision making activities and encapsulate subjective data.

1.2 Objectives

1.2.1 Objective 1:

Appraise a number of decision making scenarios to allow a greater understanding of the required features the methodologies will need. This will be done in collaboration with other researchers.

1.2.2 Objective 2:

To identify and enhance a suitable modelling notation that can be used for developing workflow models based on decision making scenarios.

The modelling notation selected should be extendable in order to allow extensions and enhancements to be provided to address limitations in the modelling notation where identified.

1.2.3 Objective 3:

To identify a suitable workflow engine that support the execution of workflow models. The workflow engine should also be extendable to incorporate the components developed by the methodologies in this thesis.

1.2.4 Objective 4:

To devise a method that captures the interactions between participants in a decision making scenario, such as a person asking another person a question. The method should also allow these interactions to be expressed as part of the workflow model.

1.2.5 Objective 5:

To establish a methodology for encapsulating vague and subjective data that forms part of the decision-making scenario, for example, people's preferences to one type of surgical procedure that minimises scarring. Vagueness is the lack of clarity of the available information regarding a decision making problem, such as data that lacks detail, or is based upon uncertain classification or measurements, such as empirical data or informal observations. Subjectivity is considered the view of a state from a person's perspective, such as what temperatures are considered hot or cold [12]. Uncertainty can be considered the unknown likelihood of an event happening. In complex decision making contexts, it can be extremely difficult to capture crisp information that is vague, subjective or uncertain without diluting the meaning or significance of the information.

In order to accommodate this, the evaluation of two approaches to uncertainty will be considered; fuzzy logic and Bayesian theories. The most appropriate approach, based on examining the application areas for each theory, will be incorporated into the decision making model.

1.3 Contributions

This thesis presents an approach to produce workflow models that define the activities undertaken in shared decision making scenario involving the communication of risk and encapsulate subjective data that forms part of the given scenario.

This thesis provides the following contributions:

1. A set of extensions to the Business Process Model and Notation (BPMN) formalism to encapsulate specific activities and properties of decision making scenarios derived from case studies. These extensions are called BPMN decision making (BPMNdm). These extensions are defined with the aim to assist in developing design requirements for decision support tools that could be appropriate for the given scenario. BPMN was chosen due to its rich set of graphical notation objects, existing use for the construction of workflow patterns and a built-in extension mechanism.

2. A set of rules representing people interacting (such as a person asking a question) that can be used to control workflow models that model interactions between people in a scenario. These rules were derived from qualitative meeting observation notes of a series of decision making meetings.

3. A methodology of incorporating subjectivity into a BPMN workflow model. Ethnographic data can produce both quantitative (i.e. a person's age, travelling distance etc.), and qualitative (i.e. a person's religious preference, perceived economic status etc.) data which may be subjective, such as a person's preferences for treatment, for example. Using Triangular Fuzzy Numbers (TFNs) to encapsulate the subjective data into a tuple provides a method of retaining the data's subjectivity while allowing the data to be used in a manor similar to non-subjective data. This methodology is derived from a case study and the development of a shared decision making tool.

4. For each contribution, the research methods adopted or developed are documented, describing how initial data, such as qualitative meeting observation notes, was collected, analysed and transformed into the final results, such as a BPMN extension or rule.

1.4 List of publications

Nesbitt D, Turland J, Sinmai K, van Moorsel A, BPMNdm: Extending the BPMN Formalism to Aid the Decision Making Process. *Business Process Management Journal, in preparation.*

Flynn D, Nesbitt D, Ford GA, McMeekin P, Rodgers H, Price C, Kray C, Thomson RG. Development of a computerised decision aid for thrombolysis in acute stroke care. *BMC Medical Informatics and Decision Making* 2015, 15, 6.

Flynn D, Nesbitt D, De Brún A, Craggs C, Afolabi B, Keilty I, Lee C, Bradwel S, Dick K, Pointon S, Gerrard S, Williamson R. *A theory-based educational film about shared decision making in primary care for depression.* Newcastle upon Tyne, UK: Newcastle University, Digital Media Services, 2015.

De Brún A, Craggs C, Nesbitt D, Afolabi B, Thomson R, Flynn D. Co-production of a Theory-Based Film about Shared Decision Making in Primary Care for Depression. *In: International Shared Decision Making/ Evidence Based Health Care conference*. 2015, Sydney, Australia.

Flynn D, Nesbitt D, McMeekin P, Rodgers H, Price C, Ford GA, Thomson RG. A Computerised Decision Aid for Treatment of Acute Stroke with Thrombolysis (COMPASS). *In: 7th International Shared Decision Making (ISDM) Conference and the 4th International Society for Evidence-Based Health Care (ISEHC) Conference*. 2015, University of Sydney, Sydney, Australia.

1.5 Thesis structure

The remainder of this theses is structured as follows: Chapter 2 presents a background and related work to provide the context and underpinning theories for this thesis. Chapter 3 presents an extension of the BPMN formalism based on the analysis of the thrombolytic stroke treatment scenario, plus contributions from James Turland and Kanidia Sinami. Chapter 4 presents the development of a set of interaction rules produced by analysing the meetings of MAPPA (Multi-Agency Public Protection Arrangements). Chapter 5 presents how subjectivity can be encapsulated in workflow models by using Triangular Fuzzy Numbers (TFNs), with an example provided from questionnaire data collected as part of the thrombolytic stroke treatment work. Chapter 6 provides further technical details of how the BPMN extensions defined in Chapter 3 and the Drools Rules developed in Chapter 4 can be implemented and executed using the jBPM workflow engine. The conclusions and future work are presented in Chapter 7.

Chapter 2 - Background and related work

2.1 Group and individual decision making

Hirokawa and Poole identify the decision making process as a defined sequence of events. A simplified interpretation of their observations can be considered a decision process that begins with a set of aims that a person or group would like to address. This is followed by a cyclic loop of communication and decisions to satisfy these aims [13].

Decisions can be defined as agreement or disagreement with a specific issue or point. However how people engage in verbal commutation to make a decision is considerably more complex as Hirokawa and Poole argue that the communication process incorporates such processes as analysis, information generation and exchange, an incorporation of contextual factors of the meeting environment and task, visibility and availability of information and the roles people play in the decision making space. People's roles have a significant impact on the type of decisions made [14], [15]. For example, if a group has a hierarchical social structure (i.e. a boss and subordinates) then this will result in different communication characteristics compared to a group of peers that would be typically free of hierarchical constraints. For example, considering a decision making scenario that involved a boss and subordinates, the hierarchical social structure would suggest that the bosses' decisions would carry more weight or importance than decisions made by the subordinates. The decision making process is summarised in Figure 3 [13]. In this thesis, where group decision making is use, it is assumed that decisions are made based on group consensus, i.e. the majority of the participants agree on a single solution.

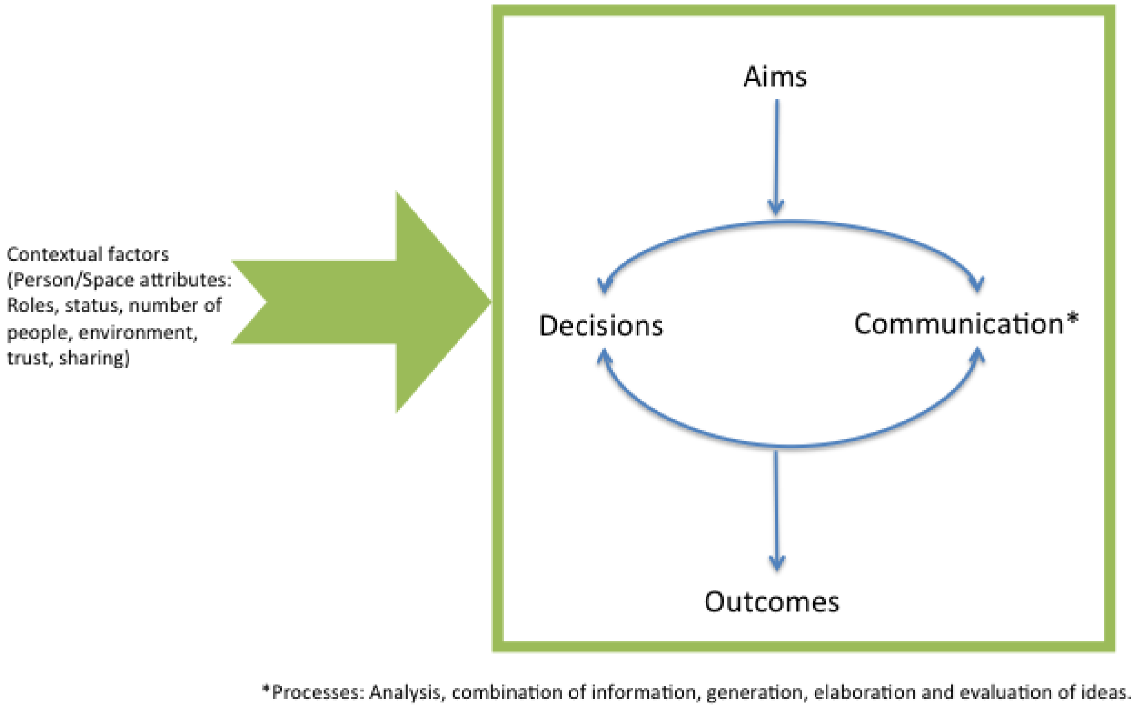

Figure 3: High level diagram of how groups perform decision making activities to satisfy a set of aims.

Considering how an individual makes decisions, as with group decision making, the contextual factors in the decision making environment will influence the individual decision maker [16]. In addition to contextual factors, the decision maker's own preferences, judgements and habits will influence what option they will choose [16]–[18].

There are two strategies for modelling and simulating decision making activities undertaken by human participants: normative decision making and descriptive decision making. Normative decision making theories describe options people should choose while descriptive decision making theories attempt to describe what options people actually choose [19]–[21].

Applying descriptive decision making theories relies on the use of empirical evidence based on the observation of decision makers [19] and is subject to change over time as peoples' preferences change, or can be unsuitable when trying to apply the same decision making model to different cultures [22]. Normative decision making theory is commonly used to describe a decision making process as by considering options that decision makers ought to make, it is easier to conceptualise how decisions are made based on choosing the optimum option [22]. However, the limitation with normative decision making theories is that people do not always make rational decisions [19], [23] but can be influenced by their own beliefs and preferences. Therefore, when conceiving the option that a person ought to make, it is suggested that a person's own preferences and beliefs are incorporated when considering the decisions a person ought to choose [22].

Normative decision making models typically use decision tables to determine which option is the most preferable [24]. The decision table has columns that indicate states which are events that are out of the control of the decision maker, with the rows indicating outcomes, which are dependent on the corresponding state and the option selected [25]. This is summarised in Table 1.

	State X	*State Y*
Option A	Outcome 1	Outcome 2
Option B	Outcome 3	Outcome 4

Table 1: Structure of a decision table

To illustrate how a decision table works in practice, an example is provided by Martin Peterson, where in a scenario, a house fire and no house fire are the two states. The two options are for the owner to take out insure and to not take out insurance. The outcomes indicate the financial cost: [22]

	Fire	*No Fire*
Take out insurance	No house and $100,000	House and $0
No insurance	No house and $100	House and $100

Table 2: Example of using a decision making table with a scenario of calculating the costs of having insurance in the event of having a house fire.

The decision table in Table 2 shows four outcomes. The next step would be to rank the outcome according to a preference scale. This is generated by consulting the decision makers to elicitate their preferences. In this example, if a decision maker valued retaining the value of the house regardless of the two possible states, they are more likely to choose to take out the insurance.

2.2 Workflow modelling languages

A workflow model is a representation of the sequence of activities that an entity such as a person or organisation, performs in order to achieve a goal. This goal could involve a decision making problem and encapsulate the decision making activities described in Section 2.1. Workflow models are typically used to analyse the characterisers of the process they represent, such as its resource requirements or the processes resilience [26]. A key concept of a workflow model is the *control flow* or *sequence flow*, which is defined as the order of the tasks (or nodes) that are executed in a workflow model [27], [28]. Combining tasks and sequence flows with logical gates allows for the creation of workflow patterns, which Wil van der Aalst et al. describe a series of design patterns that encapsulate a range of real-world activities [29], [30]. These design patterns are summarised as:

Basic Control patterns, such as a sequence of activities and exclusive choices and Advanced Branching which represent conditional and unconditional splits in the control flow, such as when a decision is made or when two tasks are performed in parallel.

Synchronization Patterns which combine two or more incoming control flows into a single control flow, such as the outputs from three multiple choices with the selected control flow being used.

Structural Patterns define workflow pattern restrictions, such as the prevention of arbitrary loops and ensuring that there is only one finish node. These structural patterns are enforced by the underlying workflow engine. For example, the workflow engine jBPM 5 validates the workflow model before execution and generates an error if a pattern that is not allowed by the formalism is detected [31].

Workflow models have two key characteristics: firstly the visual and spatial representation of the tasks and secondly control flows and an underlying formalism that provides a formal representation of each task, control flow, any associated constructs and the model itself [32]. The visual and spatial representation of the model is useful for describing and manipulating the model for both technical and non-technical users. The underlying formalism contains additional logic to define the behaviour of the entire model or specific nodes where necessary in order to allow the workflow model to be executed using a workflow engine. An example of this is the use of logical statements to evaluate variables at a specific decision point in the model [31], [33].

There are a plethora of workflow modelling notations available. Two examples described here are BPMN (Business Process Modelling Notation) and YAWL (Yet Another Workflow Language). These two notations were considered for use in this thesis due to their use of multiple graphical notations to represent different aspects of a workflow scenario [34], [35]. Both BPMN and YAWL are intended to model business processes such as the activities and sequences used to deliver products to customers as part of the operation of a business [36], [37]. Since these types of business activities naturally involve collaboration between people, discrete tasks and defined goals, we consider using such a workflow language designed for business processes appropriate for describing the activities undertaken as part of shared decision making scenarios that involve risk.

YAWL was developed by van der Aalst et al. to overcome deficiencies identified when using Petri nets to define control-flow based workflow models. The principle limitations identified were defining multiple instances of the same process and synchronisation between different parts of the workflow process [37].

To address these limitations, the authors proposed a new definition composed of *tasks* and *conditions.* A task could be defined as a single atomic task, composite tasks, multiple instances of atomic or compose tasks, and AND/OR/XOR split and join tasks. Split and join tasks define the control flow of the workflow model. Conditions are composed of input, output and generic conditions. Like Petri nets, YAWL uses tokens to describe the traversal of the model [37], [38].

BPMN is another type of workflow modelling language. BPMN was specifically designed to be a notation that is understandable to business analysts, system designers and business end users who would manage or monitor the processes [35].This intention resulted in the language having a large set of graphical notations. These notations are divided into two groups: flow objects and connecting objects.

While jBPM supports the BPMN 2.0 extension mechanism, jBPM only supports a subset of the BPMN 2.0 formalism [31], [39]. The supported elements are summarised as:

- Flow objects: tasks, gateways, start events, end events, intermediate throw, intermediate catch events and documents
- Connecting objects: sequence flows and document associations.

jBPM does not support the BPMN 2.0 notations and interaction patterns such as choreographies which are designed to show the interaction between two or more actors in a decision making process. An actor is analogous to participants in a real world scenario [40].

Flow Objects

The first group, *flow objects* define the tasks (termed *activities* in BPMN), conditions (termed *gateways*) and start/termination events in the workflow model.

Start Events symbolise the initiation of a process. Start events typically have a green background. These are the first elements in the workflow model to be triggered.

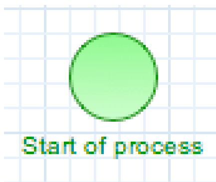

Figure 4: BPMN Start Event

The **End Event** symbolises the end of a process. The end event typically has a red background. These are the last elements in the workflow model to be triggered.

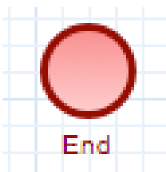

Figure 5: End Event

Intermediate events symbolise events that do not start or end a process, instead they symbolise an event that can result in the execution of a different set of tasks depending on the current state of the process. Intermediate events do not have a green background like Start events. The example in Figure 6 details a throw event with a signal definition, which indicates the event raises a signal intended for another part of the process which will handle this event.

When using throw events, the modeller must specify an accompanying *Catch event* to deal with the signal generated by the throw event.

Figure 6: Throw event with a signal definition

Catch events are intermediate events that listen for throw events and respond if appropriate. In the example in Figure 7, a generic catch event will capture all events and a conditional catch event will only catch pre-defined types of events:

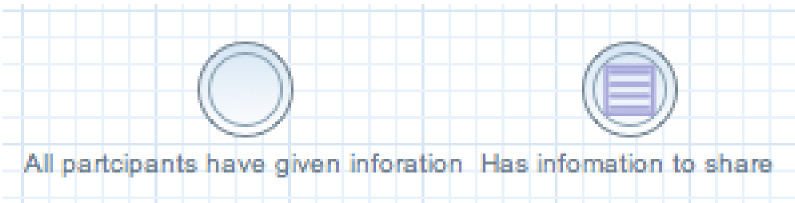

Figure 7: Generic catch event (left) and conditional catch event (right)

A **Task** represents a discret piece of work that a person or a group of people perform:

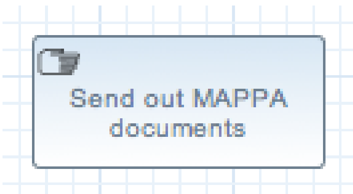

Figure 8: A BPMN task

Gateways represent the decision points in a model. Figure 9 depicts an exclusive *gateway* which allows only a single flow of the workflow model to be executed. The outgoing sequence flow is determined by the evaluation of a condition embedded in the gateway, and interpreted by the workflow engine at run-time. Other types of gateways include *parallel gateways,* representing concurrent sequence flows, *event gateways* which are triggered by events raised in the model rather than the evaluation of conditions, *inclusive gateways* for triggering 2 or more specific outgoing paths and complex gateways which signify a gateway with multiple conditions. Collectively, gateways that split the sequence flow are know as *diverging gateways.* [41].

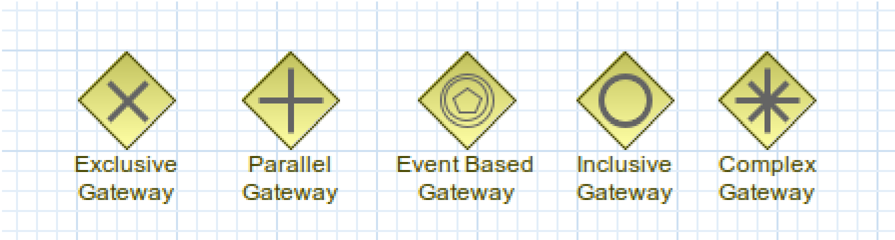

Figure 9: Gateways

Exclusive and parallel gateways are also used to converge multiple sequence flows into a single sequence flow. Exclusive converging gateways will trigger the outgoing sequence flow when one token is received from any incoming sequence flow. The Inclusive converging gateway triggers the outgoing sequence flow only when all tokens are received from all incoming sequence flows. The design pattern for diverging and converging gateways is shown in Figure 10 where an exclusive gateway "Choose task" mandates a single choice between Tasks B, C or D. This is followed by a converging exclusive gateway, which will trigger Task E when one token is received from one of the sequence flows emerging from Tasks B, C or D.

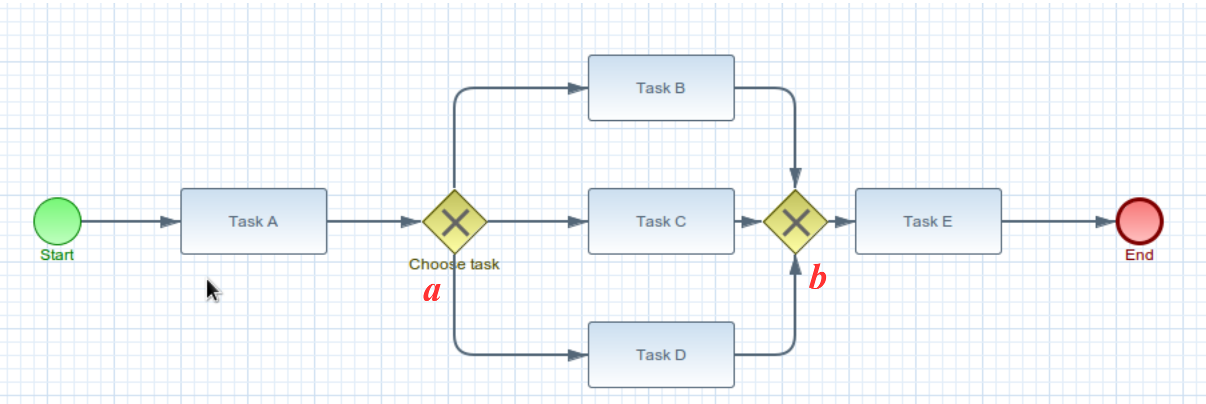

Figure 10: Sample BPMN workflow model showing a diverging exclusive gateway (a) and a converging exclusive gateway (b)

Assets represent documents and groups of documents or databases. An asset may represent a piece of paper, a document, binder, secured network drive etc.

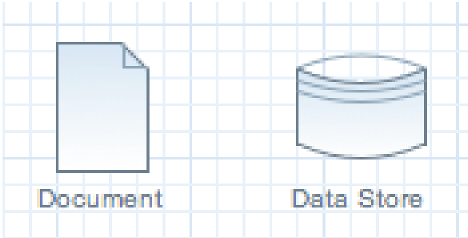

Figure 11: BPMN representations of assets: a document (left) and a data store (right)

Connecting objects

The second group, *connecting objects* define the connections between flow objects and other associated objects in the model. A **Sequence Flow** represents the logical flow of a process from a start or catch event, through to any activities and/or gateways and finally onto the end event. This is shown in Figure 12 where a Message event triggers a task, which in turn triggers an end event. An **Association** is the link between an asset and its associated task that uses the asset. This is shown in Figure 12 where the MAPPA Document is used with the task "Send out MAPPA document".

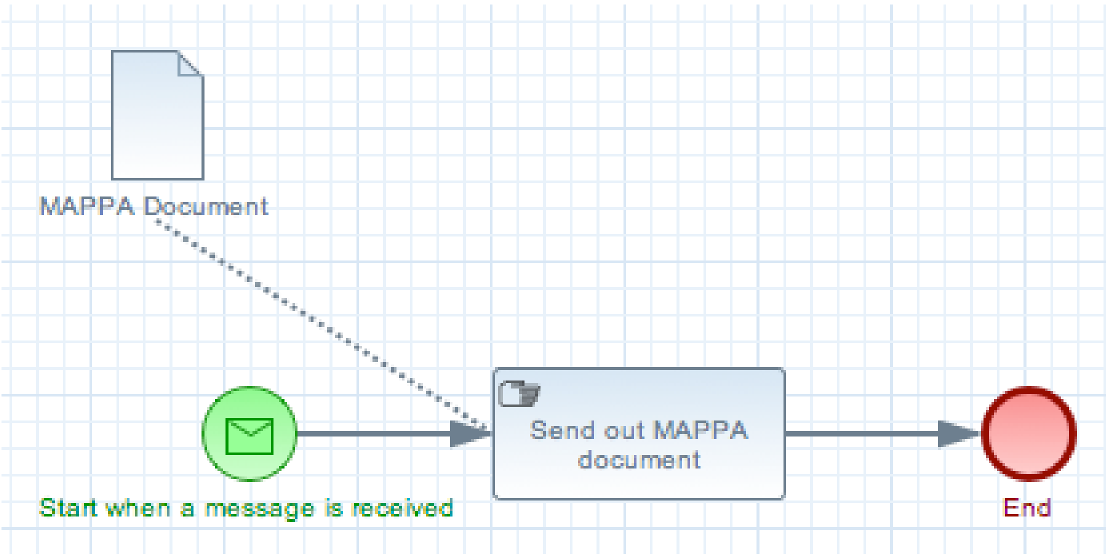

Figure 12: Simple BPMN process representing a person sending out a MAPPA document.

Swimlanes represent the partitioning of activities and other assets in the workflow model. A single swimlane can represent one or more individuals which represent a group of users (i.e. meeting attendees) who largely perform the same set of tasks. Any tasks or assets that are contained with in a swimlane are considered *private* tasks and assets, i.e., tasks that are conducted personally by one person, or assets which belong to one person (for example, a paper notebook) and that is not typically shared with the other participants. Tasks and assets that straddle swimlanes or fall outside of swimlanes are considered shared tasks and assets, i.e. tasks that are performed as a group or assets that can be shared (i.e. paper printouts).

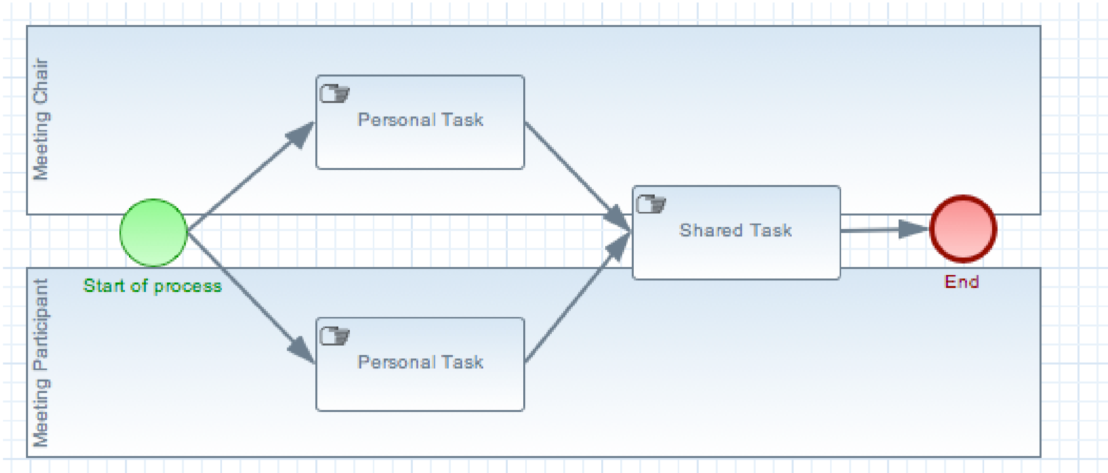

Figure 13: Simple BPMN model showing swimlanes, personal and shared tasks.

2.2.1 BPMN extension mechanism

Version 2 of the BPMN specification introduced the capability of extending the underlying formalism to tailor workflow models for specific design requirements for the given scenario [41], [42]. This has been used to provide additional definitions for existing elements in the BPMN formalism [43], or to define new elements tailored to a requirement for the real world problem being modelled [44]User-defined extensions are defined using an XML Schema Extension Definition Document (XSD), which contain sets of Extension Definitions, which in turn contain the individual attributes of each extension. The resulting XSD is then imported into the XML BPMN model definition document, where the extensions are used on the desired nodes of the BPMN workflow model [42].

Brambilla et al. described their extensions to the BPMN formalism to support socialisation interactions between human actors in workflow models [45]. These extensions consisted of lane attributes to identify specific actors, additional tasks and events to encapsulate specific types of social interactions, such as voting and commenting activities between actors. Saeedi et al. produced an extended BPMN formalism for incorporating quality requirements into BPMN workflow models [46]. The primary additions were the definition of a "QualityRequrements" task defining specific requirements to meet a quality threshold for a given task, incorporating the attributes Cost, Reliability and Response time. Gagné and Trudel describe modifying the BPMN specification for visualising time constraints of task nodes on the workflow model [44].

There are two principle limitations to the BPMN 2.0 extension mechanisms. Firstly, there is no specific methodology or guidance for defining extensions. Stroppi et al. describe a four-step process of transforming a conceptual extension to a formal BPMN extension XSD using UML diagrams as an intermediate step [42]. Secondly, the BPMN specification does not specify any graphical notation to represent the extensions defined by the user [41], [42]. Instead, the modeller must implement the graphical notation manually using a model editor, use the existing graphical notation that matches the extension as close as possible, or by modifying the workflow engine to visualise the extensions [47].

In the following example, a simple BPMN workflow model was devised, comprising of two tasks: where a user creates a report, and the second task where the user emails the report. This is shown in Figure 14.

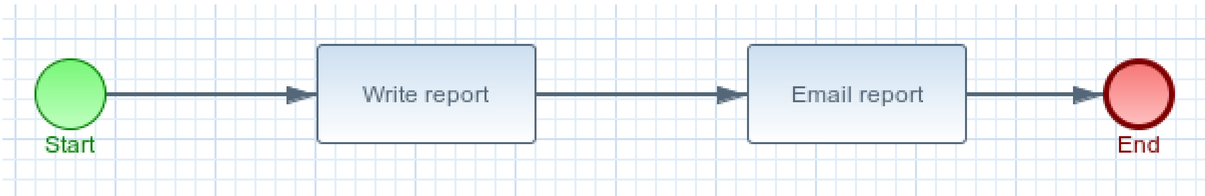

Figure 14: Simple BPMN model showing two tasks.

If the modeller wished to embed the file name of the report created for the “Write report” task, and the email address that would be used in the “Email report” task, the modeller would develop a XSD file that defines the extensions needed to express these extended attributes in the model. This is shown in Figure 15.

```
<?xml version="1.0" encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">


<group name="reportTask">

        <sequence>

                <element name="reportName" type="String" minOccurs="1" maxOccurs="1" />

        </sequence>

</group>


<group name="emailTask">

        <sequence>

                <element name="emailAddress" type="String" minOccurs="1" maxOccurs="1" />

         </sequence>

</group>


</schema>
```

Figure 15: XSD file showing the definition of a report task and email task.

In Figure 15, two groups are defined for each extension. Firstly, a `reportTask` group is defined that contains a single element, `reportTask`, that will store the file name of the document. Secondly, a `emailTask` group is defined that contains a single element, `emailAddress`, that will store the destination email address for the task. The XSD file is saved as `extensions.xsd`.

```
<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:java="http://www.java.com/javaTypes"
xmlns:tns="http://www.jboss.org/drools" xmlns:ext="http://www.ncl.ac.ukcom/myns"   1
xmlns="http://www.jboss.org/drools"
xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL BPMN20.xsd
http://www.jboss.org/drools drools.xsd http://www.bpsim.org/schemas/1.0 bpsim.xsd"
id="Definition" exporter="org.eclipse.bpmn2.modeler.core" exporterVersion="1.1.2.Final"
expressionLanguage="http://www.mvel.org/2.0" targetNamespace="http://www.jboss.org/drools"
typeLanguage="http://www.java.com/javaTypes">

  <bpmn2:import importType="http://www.w3.org/2001/XMLSchema" location="src/extensions.xsd"
namespace="http://www.ncl.ac.ukcom/myns" />   2


…

<bpmn2:task id="task1" name="Write report">

      <bpmn2:extensionElements>

        <ext:reportName>Report on Stage 1 student progress</ext:reportName>

      </bpmn2:extensionElements>                    3

      <bpmn2:incoming>SequenceFlow_1</bpmn2:incoming>

      <bpmn2:outgoing>SequenceFlow_6</bpmn2:outgoing>

    </bpmn2:task>

<bpmn2:task id="task2"name="Email report">

      <bpmn2:extensionElements>

        <ext:emailAddress>ug-dpd@newcastle.ac.uk</ext:emailAddress>
                                                              4
      </bpmn2:extensionElements>

      <bpmn2:incoming>SequenceFlow_6</bpmn2:incoming>

      <bpmn2:outgoing>SequenceFlow_9</bpmn2:outgoing>


    </bpmn2:task>

…
```

Figure 16: BPMN XML showing the extensions being imported and used.

Figure 16 shows the XML source of the BPMN diagram in Figure 14. To use the extension shown in Figure 15, four steps are necessary:

Step 1: Declare a new namespace type. This is to prevent the possibility of external XML tag names from conflicting. The URL should point to a page that describes the XML types. In this example, the tag names defined in Figure 15 will be prepended with `ext`.

Step 2: Import the XSD file. In this example, the XSD file is named `extensions.xsd` and is located in a folder named `src`.

Steps 3 and 4: Use the extensions in the desired tasks. To an extension inside an element, the extension must be surrounded by the `<bpmn2:extensionElements>` tag. In Figure 16, this is depicted for the "Write report" task, where the name of the report is provided, and for the "Email report" where the email address is provided.

2.3 Strategies for building flexible workflow models

All workflow models should be an accurate representation of the real world scenario that the model aims to represent. However, certain attributes of the scenario may be difficult to express using a workflow model such as: infrequent tasks, tasks that require data that is ill-defined or that are dependant on the individual scenario, such as the time of day when the scenario occurs or when certain staff are available. In this section, we summarise the existing principle of designing workflow models that cater for situations where the modeller may have to represent one or more activities where the scenario being modelled features uncertain or incomplete sections, as the information may be unknown and the time of developing the workflow model.

The work around design patterns by van der Aalst et al. is complimented by the concept of *flexibility patterns* described by Schonenberg et al. [48]. Models incorporating flexibility patterns are desirable as otherwise a workflow model may represent an idealised form of a process, whereas the real-world process may include variances that would lead to a significant deviation from the expected control flow of the model, thus reducing the correctness of the model. In order for a model to remain viable, Schonenberg et al. develop a "taxonomy of flexibility" that allows for the incorporation of flexibility design patterns into workflow models. These taxonomies allow the modeller to design workflow models that provide flexibility to allow each instance of the model to be customised to suit individual scenarios or situations where the predefined model may not be an accurate representation of the tasks, data or participants that are part of a specific real-world scenario. The taxonomies are summarised as follows:

Flexibility by design

A flexibility by design pattern allows a model to incorporate alternative execution paths, parallelism and repeated iteration of tasks in the model. Flexibility by design is interoperated at the *design time* of the model and the *execution path* is determined at run-time.

Flexibility by deviation

This flexibility pattern allows the run-time instance of the model to deviate from the process definition. The deviations are restricted to the run-time instance of the model and do not change the process definition.

Flexibility by underspecification

This flexibility pattern purposefully excludes the development parts of the model, while allowing the run-time execution of the resulting incomplete model. The rationale for this flexibility pattern is that the model cannot determine a future definition in a given part of the model. Therefore, the modeller will provide this definition at run-time.

Flexibility by change

This is defined as the ability to modify a process model at run time, so that some or all of the executing process instances are migrated to a new process model. The rationale for flexibility by changes is to account for instances where certain events occur that were not foreseen during the design of the process model.

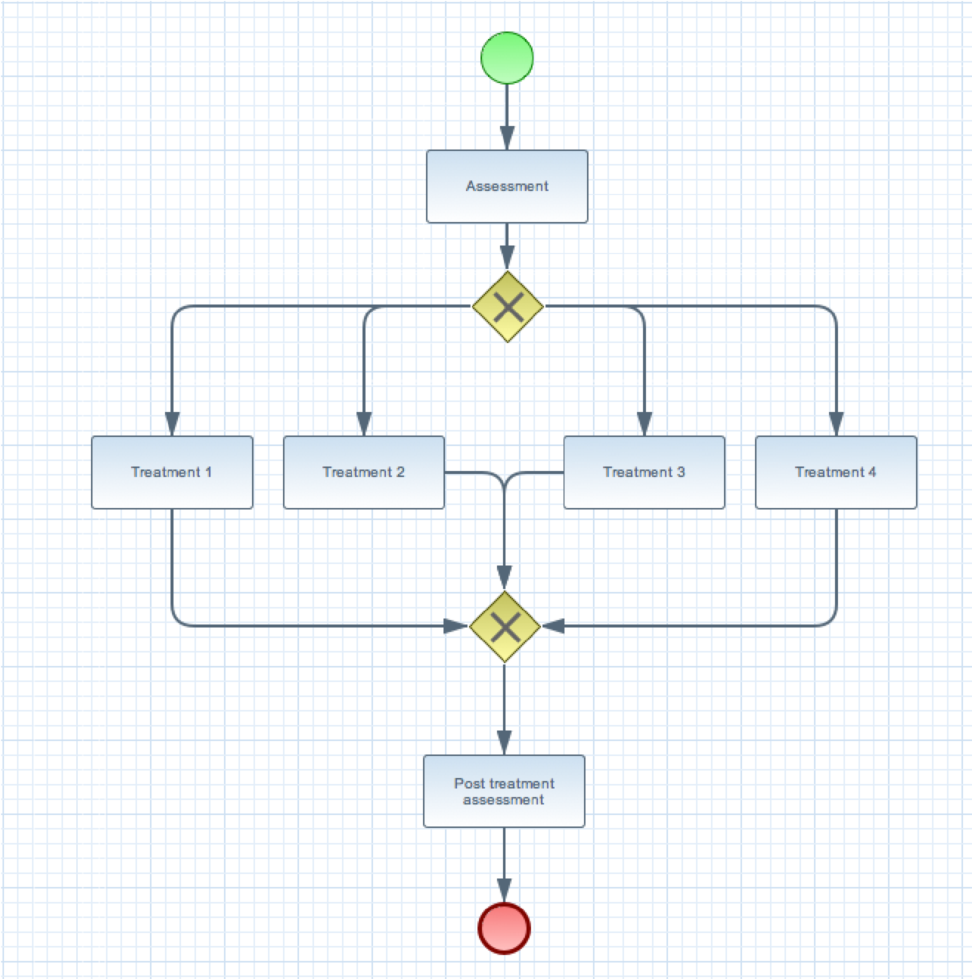

Figure 17: Sample workflow model demonstrating flexibility by design design pattern.

Flexibility patterns provide the modeller with the ability to incorporate a degree of flexibility into their workflow models by adhering to certain design strategies or by using specific features of the workflow model formalism. Examples of how sample process models would implement flexibility by design is provided in Figure 17 and flexibility by underspecification in Figure 18.

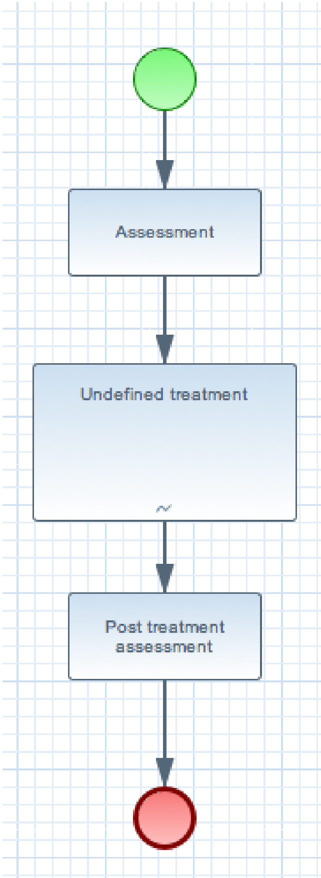

Figure 18: Sample workflow model demonstrating flexibility by underspecification design pattern

Figures 17 and 18 represent the same fictitious scenarios regarding a medical protocol; an initial assessment task, treatment phase and a post treatment assessment at the end of the process. Figure 17 adopted the principle of flexibility by design, where every possible treatment option is defined as an individual sequence flow at the design-time of the model. This type of model does not require additional implementation details to be provided at the run-time of the model by the modeller or end-user. The specific treatment and associated sequence flow would be determined at the run-time of the model. Figure 18 adopted the principle of flexibility by underspecification, where the desired treatment option would be defined and run as part of the run-time of the model. This would be supplied by the modeller or end-user who should be able to provide all of the necessary data and an additional sub-process model tailored to the scenario being evaluated.

Reijgers et al. [49] provided a case study of using Schonenberg et al. Flexibility patterns for modelling the treatment pathways for skin cancer and pre-malignant skin lesions. In this context, the researcher's work identified that "complex patients", i.e. patients whose condition required a treatment protocol that deviated from the standard protocol, was the application area for flexibility patterns. The researchers conducted additional ethnographic work with the clinicians and came to the conclusion that a "Flexibility by design" approach was most appropriate, due to the ability of the clinicians and researchers to map out every treatment pattern for this group of patients.

The usage of flexibility taxonomies to deal with specific features of the decision making scenario indicate that a visual representation does not necessarily provide a complete encapsulation of the activities or data found in the scenario. Therefore, the modeller may have to provide additional information at design or runtime of the model, or adopt specific flexibility taxonomies to provide an aggregation of the activities and data used in the scenario. The flexibility taxonomies in this section define design strategies for constructing the workflow model to deal or aggregate parts of the workflow model that are not as well defined, however further consideration is needed to how less well defined, or subjective data is handled when it defines the control-flow of the model, such as when subjective or less defined data is used in a gateway.

2.4 Handling subjective data in workflow models

In the previous section, we discussed how flexibility patterns for workflow models are used to deal with parts of a workflow model that represent real world activities that are less certain or where the information used to define the control flow of the model is incomplete or difficult to conceptualise, for example conditions that are based on the decisions made by people based on their own preferences, which we consider to be subjective data.

Subjective data can be considered a type of uncertain information. Uncertain information has a deficiency such as being incomplete, inconstant, vague or contradictory [50]. In the context of human cognition, subjective data can be considered data or information that is based on an individual's self-appraisal of their thoughts or feelings on a specific matter [51], [52]. The use of subjective information is useful in a number of contexts, an example being for assessing criteria such as life satisfaction and happiness [53].

In the context of Human Computer Interaction, examples of subjective information are a person's appraisal of their technical competence (analogous to their self-efficacy [54]) or appraisal of a computing device or similar. In the context of workflow modelling, encapsulating subjectivity is useful when stimulating human decision makers making choices. Software agents can be used for this purpose, where the agent is assumed to be an analogue for a human actor, where the agent is assumed to perform the same activities as the human actor would do in the real-world scenario [55]. In context of workflow modelling, the agent is assigned specific roles, such as people-flow, market and organisational simulation, or for dealing with possible exceptions in the workflow model when erroneous states occur [56].

In circumstances where the modeller may wish to separate the logic of controlling the outgoing sequence flows of the gateways of a BPMN model, such as in cases where of the outgoing sequence flow is determined using large data-sets that are processed in real-time, or if the requirements of the model specify that it operates on behalf of a user, software agents can encapsulate the decision-making activities in a workflow model. Eric Bonabeau points out that human agents need to take into account human traits; such as irrational behaviour, subjective choices and other "complex psychology" [55]. Coping strategies are seen as having a central role in human behaviour [57]. Marsella and Gratch present an agent based framework based on the principle of the agent appraising their environment with the use of preference attributes, which produces a coping response to the event [57] The resulting coping strategies are triggered based on specific conditions and then have a specific effect on the agent and any other associated entities in the model. Gratch and Marsella provided a detailed example of agents using appraisal and coping strategies [58]. In their example, a doctor and patient scenario was provided where the doctor was deciding to administer Morphine to a terminally ill child. In the scenario, the doctor's desire was to reduce suffering without hastening death. Morphine increases the risk of death in patients [59]. The agent operated using the appraisal of past, present and future events and activities. This type of agent design pattern is analogous to Belief-Desire-Intention (BDI) agents. A BDI agent provides functionality that is analogous to the appraisal and coping strategies defined by Marsella and Gratch. The basic architecture of a BDI agent is set of *beliefs* or assumptions about the world, *desires* which are tasks or states the agent would like to achieve and *intentions* which are the sequence of events/tasks the agent will undertake [60], [61].

The user of software agents to encapsulate subjective data is not always necessary. If the model requirements do not justify the use of an agent, for example if the actors in the model are a direct representation of the real-life human actors and the autonomous functionality of agents is not necessary to represent the activities these actors do, the salient functionality of agents, such as decision making and commutation can be directly represented in the BPMN model and functionality provided by the workflow engine [62], [63].

When collating subjective data for analysis, statistical analysis can be performed on the data set. Furthermore, there are strategies for representing and using subjective data. We consider two strategies for encapsulating and using subjective data sets: Bayesian sets or fuzzy sets.

Bayesian probabilities work on the theory of the probability of an uncertain event occurring [64]. In simpler terms, Pan and McMichael summarise Bayesian probabilities as encapsulating the potential undecidability of decision making [65]. Bayesian probabilities form part of the wider Bayes' Theorem, a conditional probability to determine probability of event A occurring assuming event B has occurred [66].

Fuzzy numbers and sets are an alternative to encapsulating subjectivity. In contrast to Bayesian probabilities, Klir and Yuan define fuzzy sets as being analogous to *possibility theories* while Bayesian probabilities are analogous to *probability theories.* The salient differences between the theories is that probability theories are based on single measurements and possibility theories are based on two bounded measurements of probability: possibility measures (the upper probability of an event occurring) and necessity measures (the lower probability of the event occurring) [50], [67]. Necessity measures are also useful for determine how robust the probability is by ensuring the probability is within the bounds of the fuzzy number [68].

In essence, a fuzzy number can be considered an interval of confidence [69] in the range of 0 (low confidence) to 1 (high confidence). For given value x, a *membership function* $\mu_a(x)$ indicates the degree of membership the value x has for a given interval of confidence. The theory of fuzziness and its applications in decision making has been controversial [70], [71] with the principal concern being the most appropriate strategy for ranking fuzzy preference scales that would represent a ranking axiom (a ranking based on knowledge assumed to be valid). The primary concern regarding ranking is that most ranking approaches require the comparison to be done with crisp values (non-fuzzy values derived from the fuzzy set), therefore these comparisons lose the uncertain and subjective properties that fuzzy numbers intended to represent [70], [72]. Yuan counter argues that the ranking operation forms the final part of the decision making process, where a discrete set of choices must be made by the decision makers [73].

In the context of business process modelling, the use of fuzzy numbers is useful when the use of subjective and/or uncertain data is intrinsic to the scenario being modelled. Cotofrei and Stoffel describe the incorporation of fuzziness into BPMN models for modelling crime analysis [43] Their rationale for incorporating fuzziness that certain aspects of the data used in crime scene analysis will be subjective or uncertain, such as the time of day when the crime occurred. The principle extensions to the BPMN formalism, were *fuzzy attributes* which represented linguistic values (such as “evening”) associated with a fuzzy set and *fuzzy constraints* which were used as part of a defuzzification function, where a non-fuzzy, or crisp value was produced. The outgoing sequence flows from the gateway that used the fuzzy number were assigned a value corresponding to a set of crisp values, and the outgoing sequence flow was chosen based on the crisp value generated by the defuzzification function.

2.5 Risk communication techniques

Determining how risks should be communicated is dependent on the context of the scenario [74] which in turn is dependant on the quality of the work undertaken to collect the user and system requirements (data produced using ethnographic methods) as well as the interpretation of the data by the modeller when building the resulting workflow model. The overarching aim of risk communication is to ensure the recipient of the information regarding the risk can sufficiently comprehend the magnitude of the risk and likelihood of the risk occurring with minimal bias, regardless of their age, cognitive ability, emotions or prior beliefs [75].

Risk is typically conveyed using verbal, graphical or textual methods. The context of a given risk will impact on how the risk is communicated and understood. However, regardless of the context, the recipient of the information should be able to accurately understand the risk. Another consideration is the type of risk; whether the risk can be quantified numerically or non-numerically. Regardless of the type of risk, a key challenge is how to overcome a person’s uncertainty and potential aversion to loss (such as financial, or impact on their long term health). Sobolev and Harvey that in the cases of appraising risk in financial trading, these factors impose higher attention demands on traders, therefore could impair their capability of performing risk assessments. The use of virtualisations of risk can ease these attention demands by providing extra information regarding the fluctuations in prices [76].

Other challenges to risk communication is the potential that the division maker under estimates their mental capability of comprehending the risks and making an informed decision. Petrova and García-Retamero observe studies relating to “risky decision making” in older adults, found those who lived independently were less risk adverse than those living in a retirement community due to higher initial perceptions of risk [77]. These findings are consistent when researching self-efficacy (the belief that a person has in completing an activity) [78] in older adults, they typically underestimate their actual compliance in completing the given task [79]–[81].

The next two sections will discuss strategies of visualising quantifiable and non-numerical.

2.5.1 Quantifiable risks

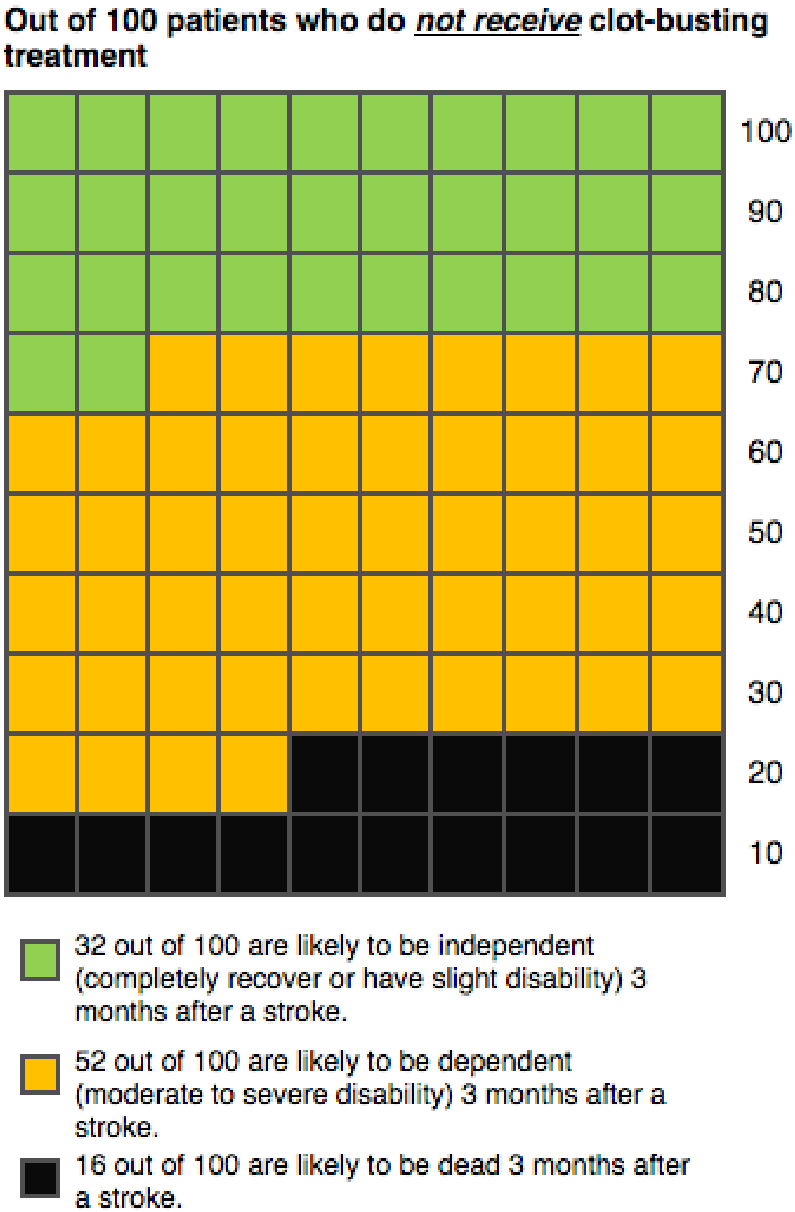

Figure 19: Pictograph of patient outcomes for stroke patients who do not receive recombinant tissue plasminogen activator drug Alteplase

Quantifiable risks can be conveyed using textual descriptions, for example, when framing a risk of an adverse event occurring, the risk could stated as "*5 more women get...*" [82].When such risks are framed in such a manner, it is vital that the meaning of the number is provided. For example, if there is a baseline for a risk (such as a given set of women in a defined age group who take a particular drug, their risk of developing cancer due to taking the drug, compared to another set of women in the same age group who do not take the drug).

Furthermore, quantifiable risks can be expressed using a variety of graphical techniques such as pictographs, bar charts and other types of graphs as an adjunct to textual descriptions [83]–[85]. Stone et al. observed that when participants were presented with graphical risk presentations without supporting text, that there was evidence of decreased understanding of the magnitude of the risks presented [86]. Research has also shown that presenting risks numerically gives patients a more accurate understanding of the given risk [87]. Figure 19 presents a quantifiable likelihood of a patient making a full or partial recovery, or dying from an ischemic stroke. These probabilities are calculated using multiple risk factors, such as age and body weight. A pictograph is used to visually quantify the likelihood of patient independence, dependence and death 3 months following an ischemic stroke. Textural descriptions with numbers are also provided as a key for the information in the pictograph.

2.5.2 Non-numerical risks

Visualisation options for non-numerical risks are comparatively limited compared to numerical risks due to the non-ordinal properties of the risks which prevents meaningful categorisation [88]. Depending on the data used, it may not be possible to categorise or quantify textual data. An example of this is when considering the risk of a sex offender re-offending. Decision makers can base their predicting on the likelihood of re-offending based on existing case studies. However, each individual case is dependant on individual factors, such as an offenders adherence to rehabilitation programs, and support and monitoring in the community [89], [90]. Quantifying these factors in predicting the risk of re-offending is therefore largely dependent on the expertise of the decision makers and the understanding of what resources are available to deal with the risk the offender poses. When using textual descriptions of non-numerical risks, terms such as "*Increased risk*" and "*probable cause of disease*" have been suggested to convey a non-numerical risk [91]. Visualising non-numerical data can be done using shape, colour and sizing [92] or by using tables and diagrams to show relationships and associations between different risk factors [93].

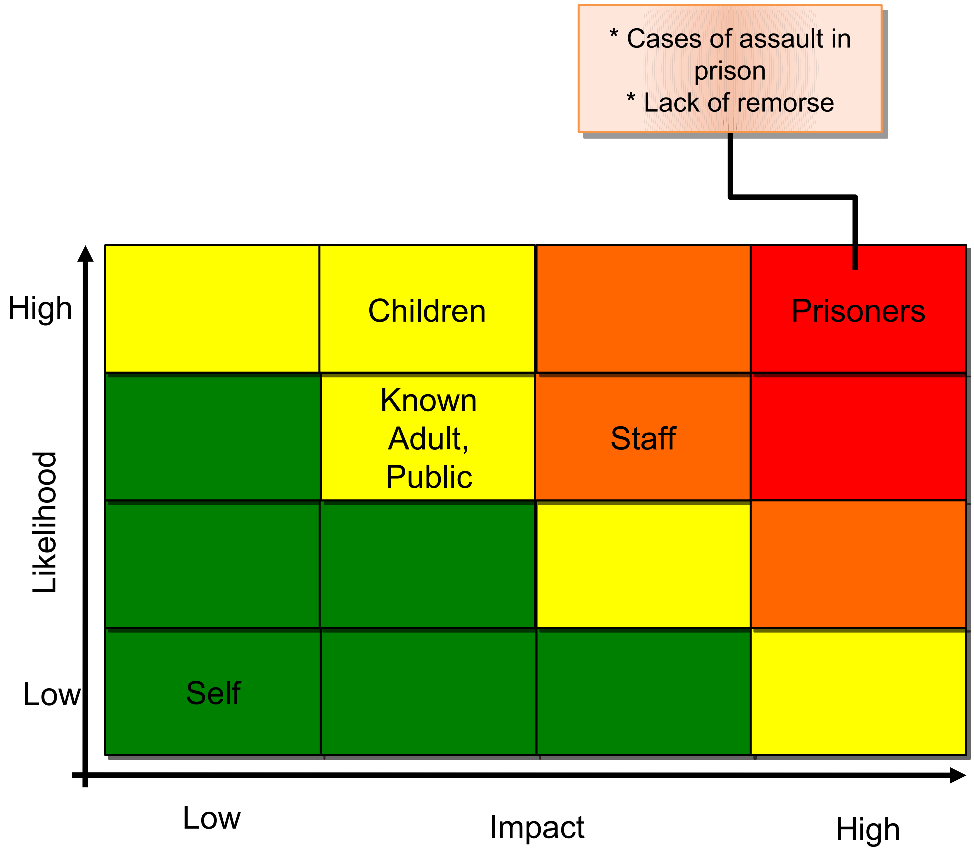

Figure 20: A heat chart of a fictitious scenario of the risk a prisoner may pose to certain members of the community.

The chart is one method of visualising qualitative data [94], [95]. A heat chart can be interpreted as a scalar table with the probabilities of the impact on the x-axis and likelihood on the y-axis on the scale of low to high. A sample heat graph is provided in Figure 20. How the data is presented on a heat chart is dependant on how the data is interpreted by the user [95], therefore it is important to have consensus on how the data is interpreted by the relevant stakeholders. A second consideration is how the data is summarised on a heat graph. The fictitious data presented in Figure 20 has additional information for each individual category placed on the heat chart. In this example, for each member of the community (Self, Children, Know adult, public, staff and prisoners), there is additional information on why a given prisoner would pose a risk to others, in the form of risk factors. The heat chart in Figure 20 shows this information in the form of a clickable annotation, however an alternative method to show this data would be to use a table [93], as depicted in Figure 21. This has the advantage of making each risk factor clearly visible, and negates the need for clickable elements if this data was to be presented on a digital device.

Risk to:	Level of Risk:	Risk factor(s):		
Prisoners	Very high !!!	Cases of assault in prison	Lack of remorse	
Staff	High !!	Historic cases of assaulting social worker	Frequently makes threats of violence towards prison staff	Does not attend behavioural improvement classes
The Public	Medium !	Cases of Anti-social behaviour	Frequently intoxicated	
A known adult	Medium !	Frequently intoxicated	Cases of verbal abuse to partner	
Children	Medium !	Frequently intoxicated	Cases of verbal abuse to their child	
Self	Low =	No documented cases of self-harm		

Figure 21: Table view of information displayed in the heat graph detailed in Figure 20.

2.6 Discussion

This chapter provides and overview of the relevant background and related work for the development of a workflow modelling based framework for decision making scenarios involving risk. The rationale for developing this framework is that no existing approaches were identified that could fully represent decision making and risk communication activities in a workflow model. For example, workflow modelling notations are used to provide a graphical representation of the individual tasks and decision points of real-world scenarios, which may include elements of decision making and risk commutation. However, these modelling notations do not directly deal with scenarios that have a degree of subjectivity or uncertainty. However, by adapting existing workflow modelling tools and existing approaches for dealing with subjectivity and uncertainty, this thesis aims to develop and describe a set of methodologies to develop workflow models tailored to describing decision making scenarios.

In section 2.1 we discuss the salient theories around group decision making, considering how decisions are made on a group level, based on the contextual factors, such as social orders. We also discuss normative decision making theories as a common method for describing how individual make decisions, based on the idea of decision tables and ranking to determine an optimal solution.

In section 2.2 we discuss how workflow modelling languages are used to conceptualise a given process, such as a group decision making process. We describe the salient features of BPMN, a workflow modelling notation that includes an extendible formalism to allow for the development of extensions to suit a specific scenario.

Section 2.3 discussed four strategies devised by van der Aalst et al. for incorporating flexibility into a workflow model. By their structured nature, a workflow model may not be able to fully express all of the activities that are undertaken in a real world scenario, possibly due to incomplete or uncertain information about the decision making scenario, such as the order of tasks or if additional tasks are needed in specific circumstances. The flexibility patterns are: flexibility by design, where the workflow model should encapsulate all possible execution paths, flexibility by deviation, where the run-time instance of the model can deviate from the process definition, flexibility by underspecification, where certain parts of the workflow model are unimplemented until the run-time of the model and flexibility by change, where the process is modified at runtime forming a new process model.

Section 2.4 discussed how subjective data can be conceptualised. There are two primary theories and methods for dealing with subjectivity; Bayesian probabilities and fuzzy probabilities. In essence, Bayesian theories focus on the possibility of an event occurring, whereas fuzzy theories focus on the probability of an event occurring.

Finally, Section 2.5 discussed how the probability of an event occurring in a risk commutation scenario can be visualised. How the likelihood of an event occurring is visualised is dependant on the type of data being displayed (such as numerical or non-numerical data). A salient consideration for the development of risk communication tools is to understand the wider context of type of person who will be interpenetrating the risk, as well as the impact the risk has on the person. Therefore, the principle aim of any risk communication tool is to support the user as much as possible, in understanding the magnitude of the given risk, without providing unnecessary bias or further uncertainly about the likelihood and magnitude of the given risk.

Choosing the optimal method for visualising the risk can determined by constructing a workflow model that captures and interprets the contextual information of the decision making scenario.

Chapter 3 - BPMNdm: BPMN extensions for Decision-making

3.1 Introduction

A strategy for analysing a decision making scenario is to use a workflow model to aggregate the activities into a set of linked nodes. The resulting workflow model can then be used to evaluate the properties of the scenario. When developing tools to support decision making, workflow models can be used as evaluation tools to validate assumptions the system designer has with potential end users, or domain experts. However, depending on the notation chosen to represent the workflow, there may be deficiencies in expressing the activities undertaken as part of the decision making process.

BPMN (Business Process Modelling Notation) was chosen as the preferred workflow language due to its large set of graphical notations for tasks, conditions and user collaborations, its intentional design for allowing non-professionals to understand its notation and its extendable formalism, allowing developers to define an extended BPMN formalism, then incorporate their extended formalism into an existing BPMN workflow model.

In order to build a workflow framework to describe decision-making activities, we evaluate a decision making case study, thrombolytic stroke treatment, specifically at the point where the decision to treat is made, identifying the salient decision-making activities. We also identify the salient decision making activities from three high-leval scenarios: developing a security policy, MSc project dissertation selection and Cardiovascular risk reduction. Once all scenarios have been described, their salient decision making features are summarised. The salient features identified are cross-referenced with the existing BPMN 2.0 formalism. In the cases where the formalism does not cover a specific activity, an extension is defined to encapsulate the given activity. We have defined these extensions as BPMN for Decision Making, or BPMNdm.

For each extension of BPMN, a rationale is given for its inclusion in the extended formalism. Next, we use a methodology derived from Stroppi et al. [96] to transform each extension concept into a BPMN extension that can be used in a BPMN workflow model. For each concept, a UML class diagram is constructed showing the relationship between the new extension and the related parts of the BPMN formalism. The UML visualisation is termed the BPMN metamodel. The UML metamodel is then used to produce an XML Schema Definition (XSD) of the extension, then an example is provided of using the extension in an existing BPMN workflow model. In order to run the resulting enhanced formalism, implementation details are provided to allow the jBPM 5.4 [31] workflow engine to interpret and utilise these extensions. As discussed in Chapter 2, jBPM supports a subject of the BPMN 2.0 formalism, therefore all diagrams are designed to use the BPMN notations supported by the jBPM workflow engine.

3.2 Defining BPMNdm extensions

3.2.1 Methodology for developing BPMN extension schemas

For each BPMNdm extension described in Table 4, a description of the construct is provided, followed by a UML class diagram of the construct, finally an implementation is provided. The proposed constructs are converted into the BPMN 2.0 XML schema using a modified methodology based upon the methodology outlined by Stroppi et. al. [96].

Stroppi et al.'s original methodology contains the following following steps:

Step 1: Definition of a Conceptual Domain Model of the Extension by using UML.

The first stage of defining an extension is to construct a UML Conceptual Domain Model of the Extension which describes the desired attributes to be represented by the extended BPMN models and their relationship with the existing BPMN formalism.

Step 2: Definition of a BPMN plus extensions model

For the second stage, a type of UML profile termed *BPMN+X* was defined by Stroppi et. al. to indicate which elements generated in Step 1 are part of the BPMN formalism (such as the existing BPMN elements), and their relation to the extensions.

Step 3: Transformation of the BPMN+X Model into an XML Schema Extension Definition Model

The third stage maps the UML elements generated in the second stage and maps the extensions to the specific types of definitions:

- `ComplexTypeDefinitions` which represent complex extension attributes, such as arrays and non-BPMN objects.
- `SimpleTypeDefinitions` each represent enumerations and variables.

Step 4: Transformation of the XML Schema Extension Definition Model into an XML Schema Document (XSD)

The last step transforms the XML Schema Extension definition model to a valid XML XSD by using the examples given in the BPMN 2.0 specification document [41]. This XSD file is imported into the BPMN XML file and each given BPMN extension element follows the format specified by the XSD.

During the course of developing the BPMNdm extensions, Stroppi et. al.'s approach was simplified in order to reduce the number of UML models required to define the extensions. The principle motivation for this was that Stroppi et. al. identified that it was possible to develop multiple Conceptual Domain Model of the Extension models as part of Step 1, with each model having different sets of attributes due to the modeller's subjective interpretation of the requirements of the desired extension. The disadvantage of this approach is that the modeller's interpretation of the requirements may be incorrect. Therefore, it wiould be necessary for the extensions to be evaluated by the experts in the domain.

The XML for BPMNdm extensions can be generalised into the following XML schema pattern:

```
<schema>


       <group name="...">

          <sequence>

             <element ... />

             ...

          </sequence>

       </group>


</schema>
```

Figure 22: Outline of an XML schema pattern used for extending the BPMNdm formalism

The XML defined in Figure 22 begins with the element Schema which contains one or more Group elements, which define one or more extension. Each Group has one Sequence element which has one or more element, which represents the additional attributes that the BPMN extension defines. The attributes for the XML extension tags are defined as `ComplexTypeDefinitions` for complex objects and arrays and `SimpleTypeDefinitions` for enumerations and variables. The final attributes to define values for are the `minOccurs` and `maxOccurs`, which specify the minimum and maximum number of instances for the given element the extension can hold.

3.2.2 Modified methodology for development of BPMN extensions

By modifying Stroppi et. al.'s methodology and the XML schema pattern in Figure 22, the following methodology was devised for implementing the BPMNdm extensions:

Step 1: Construct a BPMN model using the pre-defined formalism based on the desired scenario or activity of interest and identify any constraints/missing features during development. In order to illustrate this, we have documented four case studies and discussed the desired features for BPMNdm. This is provided in Sections 3.3 and 3.4.

Step 2: For each constraint identified, identify a BPMN element that fulfils as many of requirements of the missing feature as possible using the BPMN 2.0 specification document.

Next, modify the visual representation of the closest matching BPMN 2.0 element to represent the BPMNdm extension. This represents a visual prototype of the proposed extension.

Step 3: Once an appropriate BPMN element has been identified and visualised, construct a UML class diagram of the BPMN formalism of the existing elements used, along with the new element, showing its relationship to the extension and the attributes of the new extension. This visualisation of the BPMN formalism using a UML class diagram is considered the metamodel. In our UML class diagrams, the existing metamodel is denoted by grey objects, while our extensions are shown as white objects.

Step 4: Using the XML schema pattern in Figure 22, implement the extension as an XML Schema Document (XSD)

Compared to the methodology outlined by Stroppi et al., our methodology produces one UML class model in order to show the relationship between a given extension at the related parts of the BPMN formalism. Stroppi et al.'s methodology adopts a more incremental approach, with the development of three intermediate models before the XSD is produced.

3.3 Step 1: Case studies

To derive requirements for modelling decision making scenarios in BPMN, four case studies in involving decision making are evaluated: Thrombolytic Stroke Treatment, the development of a Security Policy for transferring data, the selection of MSc dissertation project, and Cardiovascular disease risk reduction. The Thrombolytic Stroke Treatment is the primary case study for this chapter with BPMN workflow models being constructed to show the decision making process for treatment.

3.3.1 CS1. Thrombolytic Stroke treatment

Thrombolysis is the process of the break-down of blood clots in the bloodstream by pharmaceutical agents [11], [97]. These pharmaceutical agents (also called “clot-busting drugs” or *recombinant tissue plasminogen activators* [rt-PA]) can be used to improve the outcome of an acute ischemic stroke (caused by a blockage of an artery supplying blood to, or within the brain.) [98]. However, thrombolytic treatment can cause bleeding complications, with the most serious being symptomatic intracranial haemorrhage (SICH) that typically occurs 24-36 hours after treatment and leads to deterioration of the patient or death [99], [100]. Treatment efficacy is time dependant, with earlier treatment being associated with the increased likelihood of functional independence of the patient. The maximum time following the stroke event for thrombolytic treatment is determined to be 4.5 hours, after which the risk associated with bleeding complications outweigh the benefits of treatment [101], [102].

In addition to the time the treatment is administered, the potential risks and benefits of thrombolysis are also dependant on the patient's characteristics, such as blood pressure, weight and blood glucose levels. These variables are part of the larger licensing criteria for thrombolytic treatment which is used to determine the patient's overall suitability for treatment [103]. Deciding if a patient is suitable for thrombolytic treatment is dependant on the patient being within the licensing criteria as well as the clinicians' preferences towards thrombolytic treatment [11]. However, thrombolysis is under-utilised due to clinician concerns about SICH events [104], uncertainty of effectiveness of treatment [105] and the overall benefits of treatment in practice as a function of patient characteristics [106] despite research suggesting that the benefit of treatment will often outweigh the risks for patients at risk of these adverse events.

A further consideration is if the clinician should make the finial decision to treat the patient based on the clinician's own judgement (paternalistic care model) or engage in shared decision-making (SDM) with the patient and/or family member [107]. This is dependent on the context of each individual case, as the patient may be unable to communicate or there may be insufficient time to delay treatment. However when these factors are not an issue, patents and family members appreciate decisions over treatment to be made when considering their own preferences and desires [2], [11], [108]–[110]. By using involving the patient and/or family member, SDM has been shown to improve patient engagement as well as improve long term outcomes [2], [111]. The benefits of SDM have also been identified for other health conditions, such as Schizophrenia [112] and Depression [113], [114].

3.3.2 BPMN workflow models of Thrombolytic stroke treatment

In order to show how a decision making scenario can be represented using the existing BPMN formalism, this section contains a series of BPMN workflow models that encapsulate a generalised scenario showing the decision making process for thrombosis with the overarching aim of encouraging the practice of SDM when appropriate. We assume that SDM would be used where appropriate, i.e. where the patient is well enough to make the decision to administer the drug, or if a family member is present. Furthermore, it is assumed that the type of clinician involved in the scenario wishes to engage in SDM if it is possible and would not seek consent if the patient was judged to be unsuitable for the treatment. For each model, a description is provided to describe the activities undertaken as well as any assumptions and comments and any limitations of using the BPMN formalism to represent the scenario.

Iteration 1: Initial patient contact:

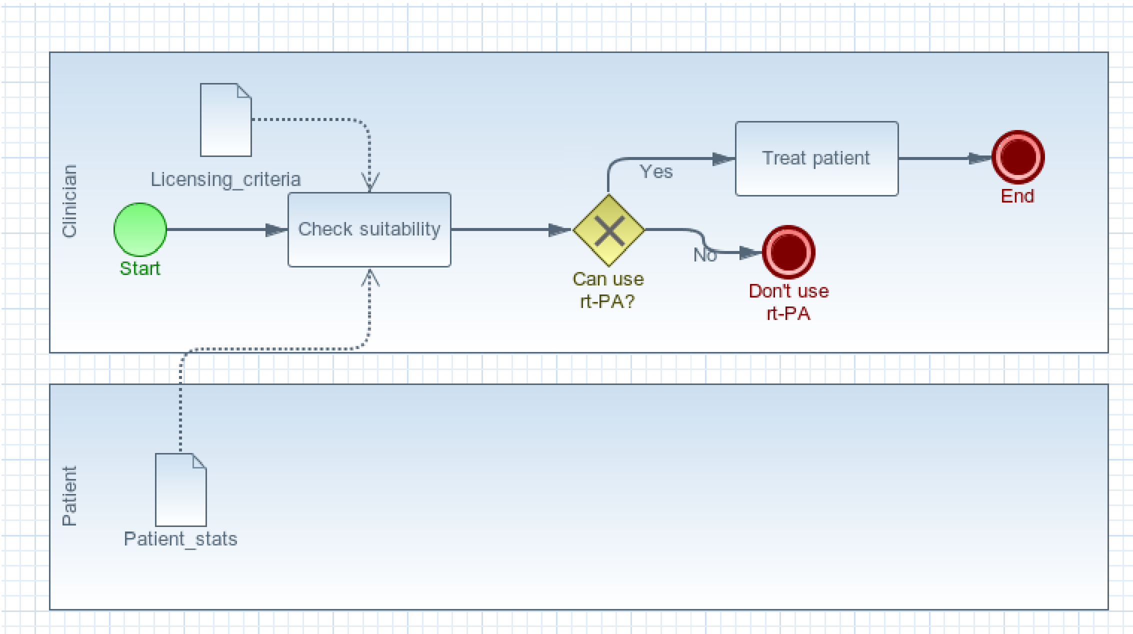

Figure 23: BPMN workflow model showing initial assessment of a stroke patient's suitability for thrombolysis

Iteration 1 shows the first version of the model, where it is assumed that the clinician makes the choice regarding treatment. The BPMN workflow model in Figure 23 depicts this initial development, which is an assessment of patient who has had an ischemic stroke. The process starts with the clinician checking the suitability of the patient. The clinician and patient are represented by two swimlanes, which indicate which person undertakes each activity and decision, as well as which documents are associated with each person. The assessment will be done by looking at the patients notes, or by using additional tests (such as measuring blood pressure). These measurements are represented by the data item "`Patient_stats`". The clinician checks these against the "`Licensing_criteria`" data object. If the patient's measurements are within the licensing criteria, the clinician will treat the patient with the thrombolytic drug. If the patient does not, then the patient will not be treated with the thrombolytic drug.

The workflow model in Figure 23 assumes that the decision is solely made by the clinician (as the “Check suitability” task and resulting sequence flows are contained by the Clinician swimlane). The clinician then decides if the patient can be administered the drug, as shown by the “Can use rt-PA?” gateway. If the clinician chooses to proceed with treatment, the patient is treated with the drug, otherwise the workflow model ends without the patient receiving the drug. The decision making activity depicted is analogous to a paternalistic decision making model, where the clinician makes the critical decisions for the patient [107]. This model is incomplete, as consent should be sought before treatment is started. However, if the modeller intends to include shared decision making activities between the patient and clinician, then additional changes are needed.

Iteration 2: Describing Shared Decision Making in the model

The second iteration of the model is to include elements of shared decision making, where the patient shares the final decision to administer the drug, assuming they meet the licensing criteria. Figure 24 depicts the changes made to the BPMN model in Figure 23. In this version of the model, it is assumed that shared decision making can be used as part of the consultation process. The scenario shows that if the patient meets the licensing criteria for treatment, the clinician will then assess if the patient is capable of engaging with the clinician. If the patient can not engage, for example if the patient is unconscious or is unable to communicate due to the side effects of the stroke, the clinician will proceed with the treatment. If the patient can engage with the clinician, the clinician will discuss the treatment with the patient, symbolised by the throw event (marked “*” on the figure) and the catch events (marked “+” on the figure) on the Patient and Family Member swimlanes. The patient will then indicate their preference. If the patient wishes to proceed with the treatment, the treatment will commence, otherwise the drug is not administered. To determine if both Patient and Family Members participate in the scenario, the modeller needs to provide the necessary logic to be implemented by the modeller in the throw event to indicate which participants should be included.

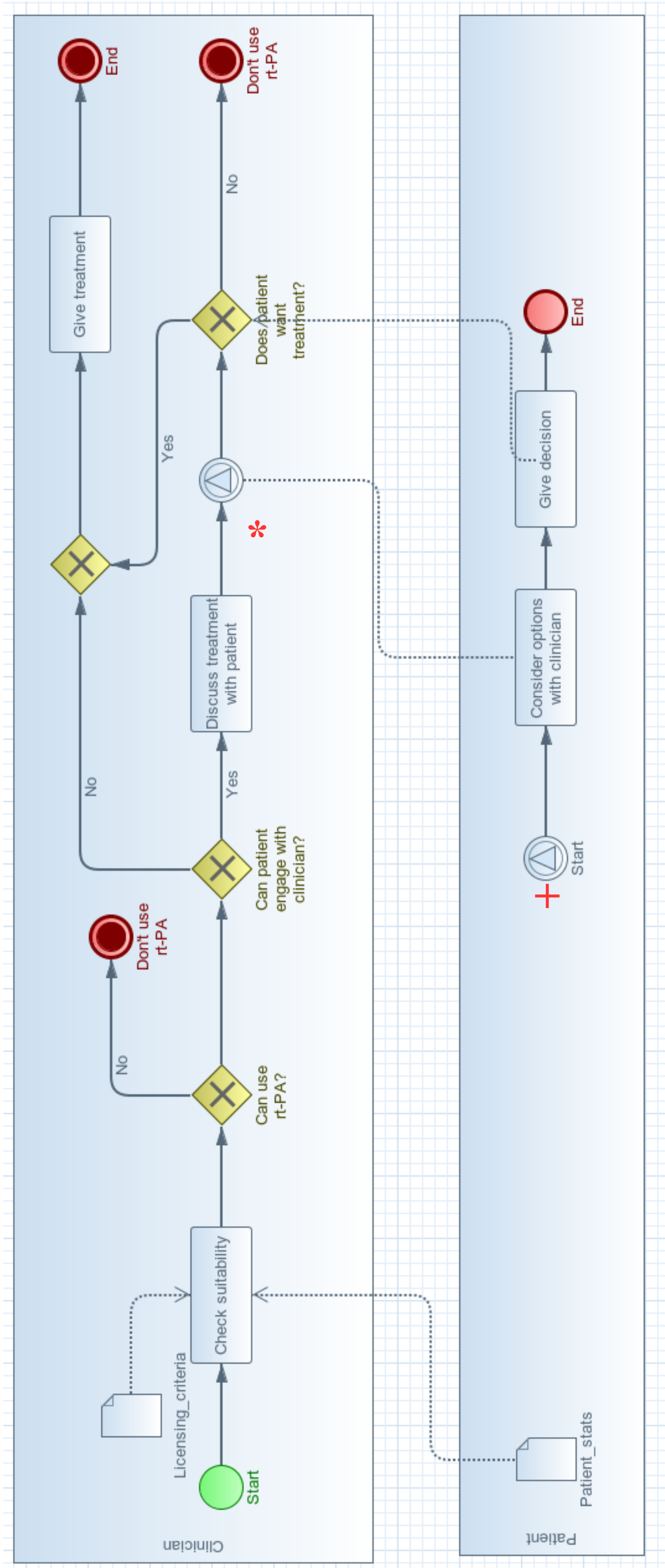

Figure 24: BPMN workflow model showing Shared decision making between clinician and patient.

Iteration 3: Family members and time constraints

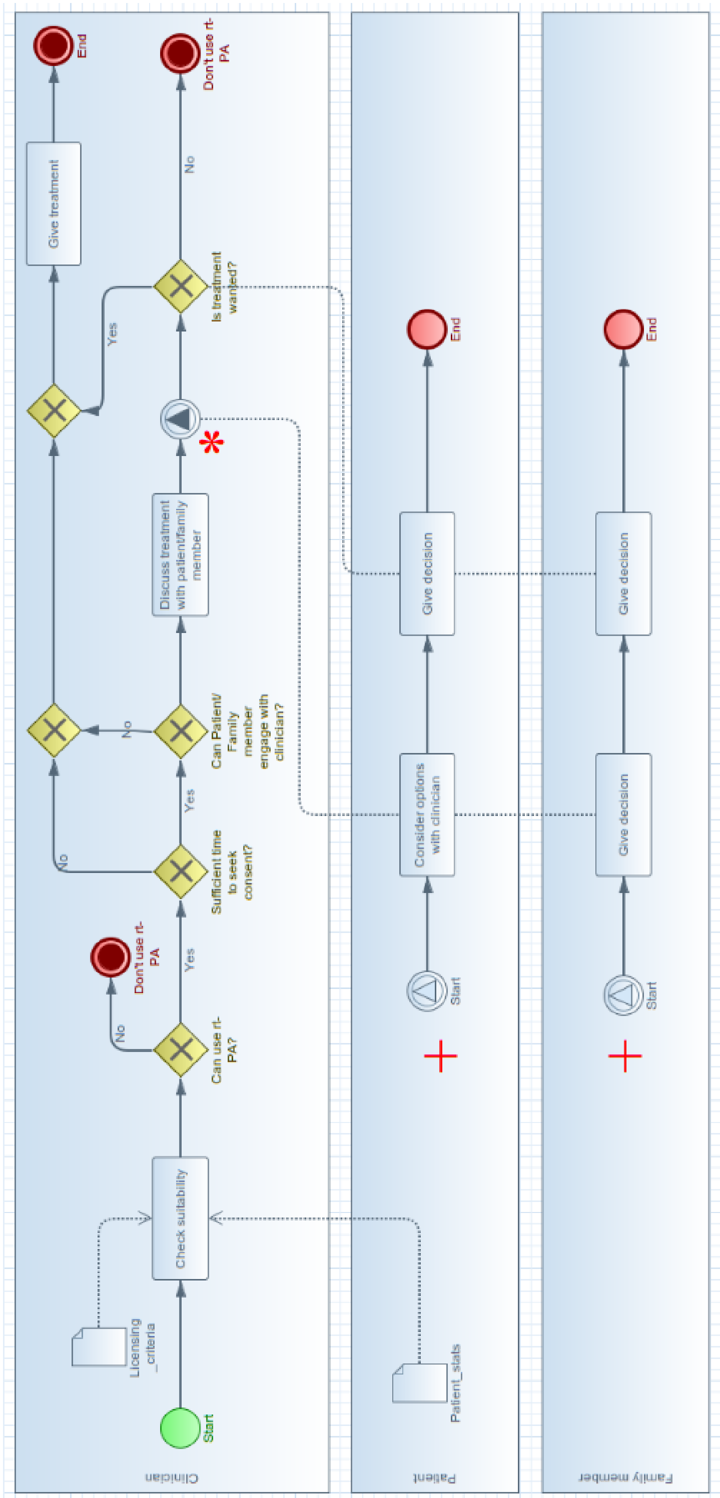

Figure 25: BPMN model showing the addition of a Family member swimlane

Iteration 3 represents the inclusion of time constraints and an additional swimlane representing a family member. As thrombolytic treatment must be given within 4.5 hours of stroke onset, if the clinician determines that there is insufficient time to consult with the patient/family member, then the clinician will proceed immediately with the treatment. This is determined by two new gateways, firstly "Sufficient time to seek consent?" which is necessary due to the 4.5 hour treatment window for the drug, and secondly "Can Patient/Family member engage with clinician?" as shown in Figure 25. As there may be other persons involved in this scenario (such as a family member of the patient), an additional swimlane is provided to represent the decision-making activities this additional person will do. As this person may not be present in the scenario, the conditional throw start event for the person can be disabled by the modeller, indicating that the person is not present. This change requires additional variables embedded in the workflow model, and the use of a workflow engine to determine the correct sequence flow. As with Figure 24, the throw event is marked with a "*" symbol and the catch event is marked with a "+" symbol. An alternative strategy for developing this model is to remove the throw/catch events and include the tasks undertaken by the Patient and Family member as part of the normal sequence flow. This was not chosen as the additional graphical artefacts on the model could reduce the model's visual clarity.

3.3.3 CS2. Developing a security policy for data exchange

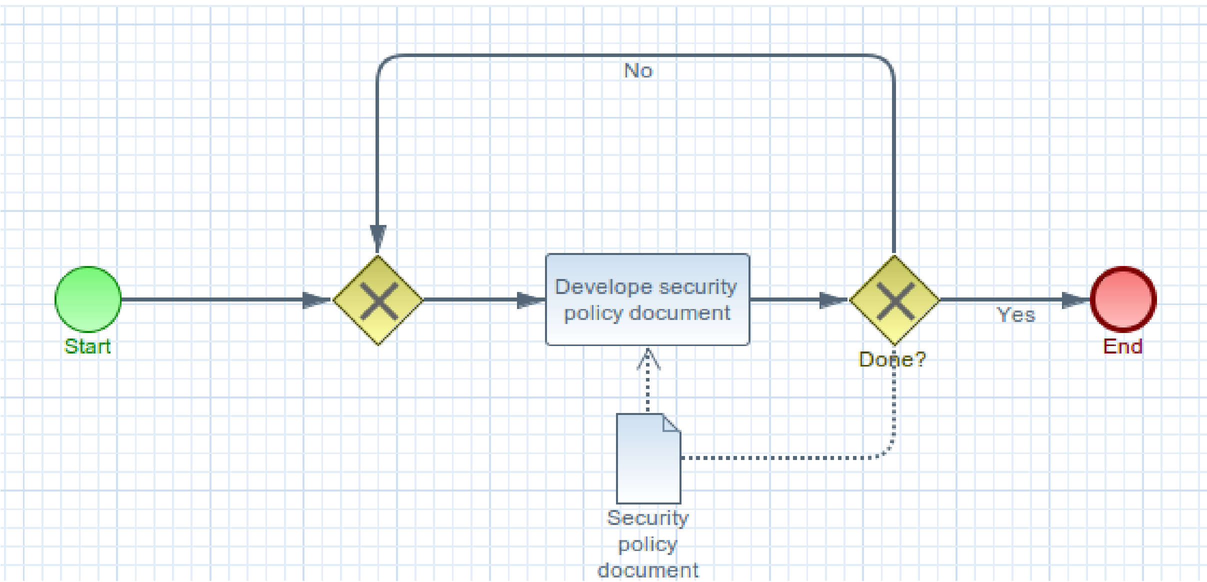

Figure 26: BPMN model showing the development of a security policy

This scenario deals with the management of data transferral (for example transferring confidential data from one server to another within or outside of the University network). This scenario does not deal directly risk communication, but rather with the generation of documentation that will be used to communicate risk and how it should be dealt with. In any decision making process relating to security policy there are finite values (typically cost and time) that must be accurately defined and adhered to. Figure 26 shows a basic BPMN representation of this process, where the security policy document is developed in a cyclic loop until the document is considered complete, as indicated by the "Done?" gateway. With these variables known, it would be desirable to indicate cost and time increments per node as you traverse the model. As budget and time scale are pre-defined, tool support would allow for the next process to be automatically determined based on confines and bounds set initially.

3.3.4 CS3. Cardiovascular disease risk reduction

In this scenario, a general practitioner is using their practice's EMIS (Egton Medical Information Systems) computer system (a database that stores a practice's patient records) [115] to check on their patient records. EMIS will flag patients who are at high risk of a cardiovascular disease (such as a stroke or heart attack). The GP will invite each patient at risk for a discussion on how to reduce their risk. This is achieved using a tool called CVdecide that displays risk presentations in the form of a Pictograph, showing their risk of disease based on such factors as the patient's weight and whether they smoke or not. Based on these factors, the GP and the Patient will discuss a plan for reducing the patient's cardiovascular risk, such as reducing their weight and quitting smoking. In this scenario, the aim of the scenario is tow fold, firstly the ultimate goal for the individual patient is to reduce their risk of having a cardiovascular event. Secondly, a further aim is for the GP and the surgery to reduce the number of patients who have a cardiovascular event, in order to reduce emergency referrals to hospital and likelihood of early death.

Based on the above scenario, a BPMN model has been constructed incorporating a consent point, indicating the goals that have been agreed by both users. This is shown in Figure 27. The process starts with the doctor assessing the patient's risk of a cardiovascular event with the aid of the patient's medical record. If they are at risk, the patient will be asked if they would like to discuss the issue. If the patient wishes to discuss the issue, both the clinician and patient will discuss how the patient can reduce their risk of a cardiovascular event. A Throw/Catch event could be use to symbolise the patient making the decision, but this was not done as there is only one specific decision the patient needs to make, therefore negating the use of the Throw/Catch event. If both patient and doctor agree on a plan to reduce their risk, this plan will be added to the patient's medical record to indicate what was agreed.

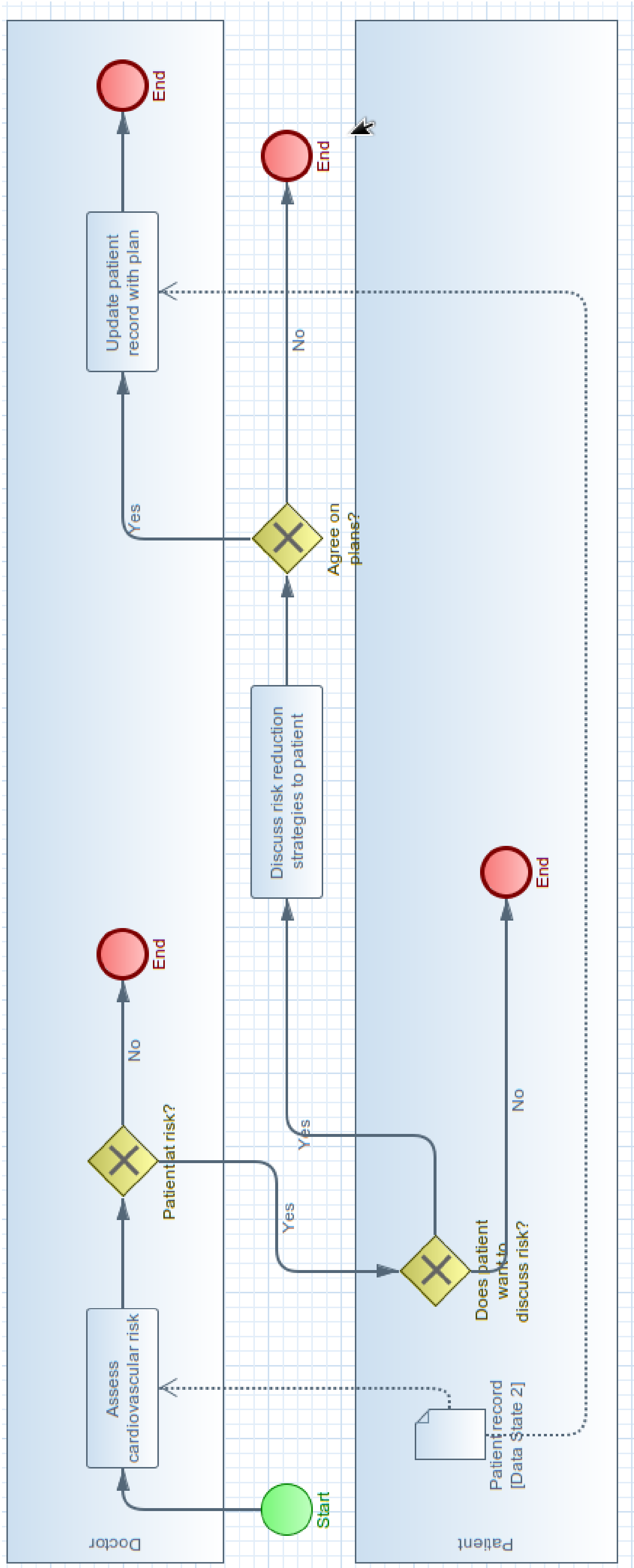

Figure 27: BPMN model showing a doctor and patient discussing cardiovascular risk reduction

3.3.5 CS4. Masters project selection

The research study is one of the most important parts of Master degree which the students will need to submit in partial fulfilment of their requirement if they are to pass the degree. For reaching the goal, multiple projects are provided to the students. They are placed into groups of size between two and four. Each group of students has a single project assigned. Each student has to do an individual dissertation of their assigned project.

In this context, a BPMN model is used to capture the decision making process in terms of the aims/objectives and what characteristics contribute to failure of the meeting. The scenario does not deal with how failures should be mitigated and focusses on failures based around immediate issues the students may have, i.e. scheduling probables without considering external factors, such as problems with their supervisor not being proficient in the subject area.

The BPMN model is shown in Figure 28. The model has two swimlanes, one representing the individual student and the second representing the whole group of students in the scenario. The process starts with the group producing the aims/objectives for the project. This is a cyclic loop where it is assumed that the group will decide if the aim and objective is complete. If it is not, then it is assessed to see if the students can complete the task during the meeting (based on constraints such as the amount of time remaining for the meeting). If the participants cannot complete the task, this will terminate the model with an error. Once the aim and objectives are complete, each individual student begins to work on their individual dissertation.

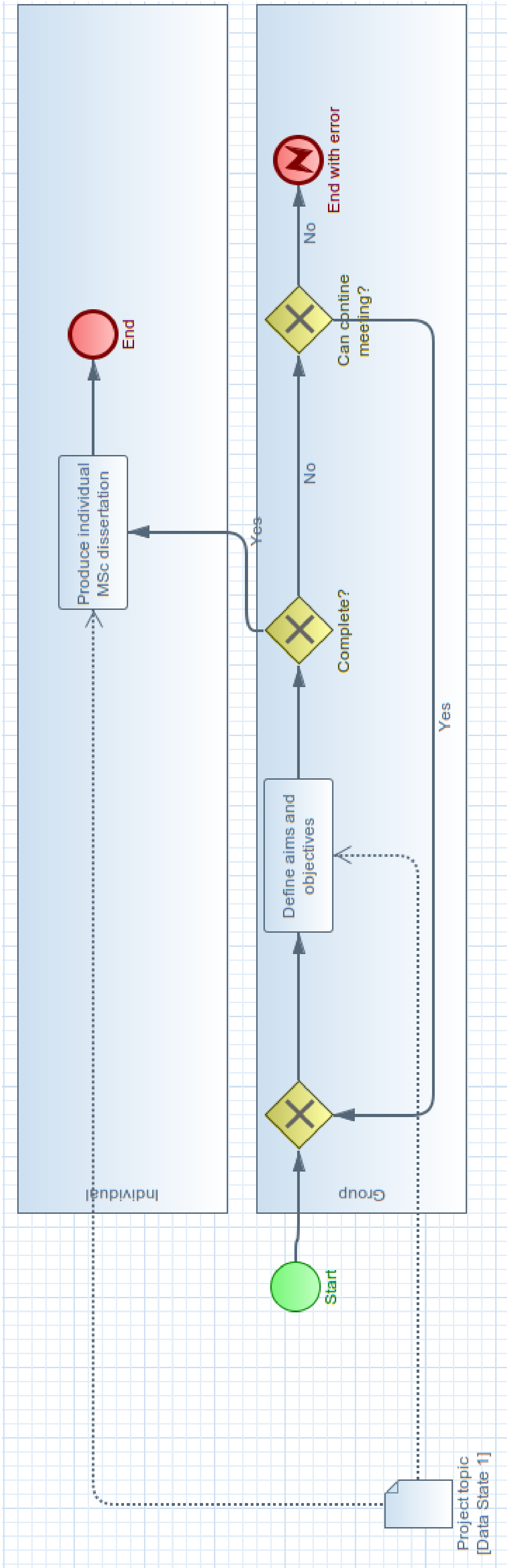

Figure 28: BPMN workflow model of masters project selection

3.4 Step 1 continued: Identification of Decision Making Features

For each decision making example described, we have appraised the salient features each case study has, and where applicable, the comparable BPMN construct as defined in the BPMN specification document [116].

Cost

Cost is identified as a requirement in CS2. Cost is a scalar value representing an incremental value associated with one or more tasks in a sequence flow. The cost associated with a task may include the price of hiring an expert, buying new equipment and travel costs. In the example of CS2, cost is assigned to each tasks that is performed when defining a security policy, and the final cost of all of the tasks should be less than or equal to the allocated budget for defining the policy. Cost does not necessary have to relate to monetary value. If a cost has a scalar value, such as the decrease of free resources as time advances, such constrains could be visualised using the sample principle as showing the scalar increase or decrease of monetary resources.

A BPMN Swimlane can be used to group tasks that incur a cost together, though this intended to show the sequence of activities undertaken and not for the increase of any associated costs with each task. Furthermore, the semantics of left to right traversal of the sequence flow of the model can be assumed to indicate the scalar property of an incrementing cost.

Time

Time is a requirement identified in CS1, CS2 and CS4. For CS1, time is a constraint that defines the maximum time the thrombolytic drug can be administered to a stroke patient (4.5 hours), measured from the onset of their stroke. This could be considered the treatment deadline. In this case, the time value would need to be evaluated at each task, in order to ensure there was still sufficient time to administer the drug. As the maximum time is fixed, this value could be visually identified on the workflow model, indicating that treatment should finish within the time constraint. This time constraint can also be used to indicate the duration each individual task should take, either as a static value defined by the modeller, or dynamically calculated based on additional runtime variables supplied by the modeller.

CS2 and CS4 have similar time constraints, with each potential scenario having a fixed time constrains (or deadline) for the overall process, with the possibility of individual time intervals for each of the individual tasks.

The BPMN 2.0 specification does not support the definition of time constraints. The closest approximation to time as identified in the case studies is the timer start and end events, which specify when a process should start or end based on a time variable.

Multi-person

All case studies describe one or more person participating in the decision making problem. The BPMN 2.0 formalism supports the representation of one or more person or actor participating in a workflow model with the swimlane construct. A swimlane visually encapsulates the tasks performed by a delegated actor. This is further extended with the collaboration type of interaction. In a collaboration, a pool type swimlane is used to represent actors, with collaboration tasks uses to represent the tasks shared by the two actors. However, collaboration tasks and Pools are not supported in jBPM 5.

Confidentiality

Case studies CS1, CS2 and CS3 describe activities that generate or require confidential information. In the context of the case studies, confidential information could be in the form of data that contains personal information, such as a patient record for CS2 and CS3. In the case of CS2, confidential information could be in the form of trade secrets, or information that could lead to compromise of security.

While the BPMN document element is designed to represent the use of documents in a workflow model, the specification has no definition of confidentiality. Therefore, to indicate if a document is confidential would require the modeller to include this as an attribute that would be evaluated outside of the workflow model.

Information capture and output

All case studies describe activities that involve the use of, or production of documents. For example, in CS1, the patient's stats and licensing criteria are examples of documents that are necessary for deciding if the patient should be treated with thrombolysis. CS2 describes the generation of a security policy document. CS3 requires the patient stats in order to help with the identification of cardiovascular risk factors. CS4 describes the requirement of indicating which project an MSc student chooses.

The BPMN document element is an analogous element that can represent the items generated as part of a task, as well as items required for a given task.

Public/private documents

As with Information capture and output activities, all the scenarios describe the use of documents. For CS1 and CS3, it is assumed that these documents (such as a patient record) are private documents that are only available to authorised people, whereas for CS2, the security policy generated may be private initially, then public once it is approved. For CS4, the resulting decision is assumed to be private to the participants, such as the student and supervisor.

While the BPMN specification provides a definition for documents, the specification does not provide for the indication if the documents are private to one or more persons or actors, or are publicly accessible to all persons. Using the definitions of scoped data to represent private data, and workflow to define private data [117] specific workflow patterns are proposed to visually indicate if documents are public or private. A limitation of visualising scoped documents is that the visualisation may lack fine-grained control of the document that may occur in the scenario. A solution for this is to provide additional logic for the runtime engine to determine how scoped data is handled at a given time during the execution of the model. This approach is discussed in Chapter 6. A disadvantage of this is that it does not provide a visual affordance of the finer grained control.

Defined goals

All case studies imply a preference for specific events to occur. In CS1, the treatment should be administered before 4.5 hours after the onset of the stroke, and ideally the clinician should discuss the treatment risks and benefits with the patient and/or family member. In CS2, the end goal of the scenario is the development of a security policy that satisfies the security requirements for the University. In CS3, if the patient is identified as high risk for a cardiovascular event, the intended goal is for the general practitioner to discuss strategies for reducing this likelihood. For CS4, the ultimate goal is for a student to select a MSc project.

In order to satisfy a policy, such as ensuring shared decision making takes place between patients and clinicians, or to ensure that a security policy conforms to a set of specifications, a specific set of tasks may need to be undertaken. Therefore, a preference value may be allocated to specific sequence flows of tasks in order to indicate that a given subset of tasks should be perform to satisfy a given policy.

The BPMN specification defines the group visual element as a method of grouping elements that are part of the same category, which could be defined as a policy. The BPMN specification also defines a text annotation element as a method for the modeller to provide additional textual descriptions on the visualisation of the model. However, for the case studies these constructs are less than optimal. For example, if a policy defines a preferred course of action based on a single activity or event, multiple instance of the group annotation would be required, and potentially grouping single tasks and or gateways. Furthermore, it a policy applies to multiple parts of the workflow model, this would lead to multiple groups with may potentially increase the visual complexity of the model. Text annotations would be appropriate in this circumstance, however if the sequence flow that satisfied the policy was assigned a weighted value, this would not be visualised by a textual annotation or a group element.

Share documents

All case studies imply the sharing of documents, such as Patient records for CS1 and CS3, security policy documents for CS2 and notes generated for a project in CS4. The BPMN specification defines message flows as a method of visually and symbolically passing messages between participants. The specification also defines associations as a method of visualising the link between documents and their associated tasks.

A concern regarding message flows and associations is where a document is used multiple times across more than one task, the resulting workflow diagram may become visually cluttered. For example, in CS2, if a working draft of the policy document is edited in 5 different tasks, this would require 5 outgoing associations from the policy document to each individual task.

Feature	Stroke (CS1)	Security (CS2)	CVD (CS3)	MBA (CS4)	Comparable BPMN construct
1. Cost		*			*No comparable construct identified*
2. Time	*	*		*	Time start event only
3. Multi-person	*	*	*	*	Swimlanes
4. Confidentiality	*	*	*		*No comparable construct identified*
5. Information capture/ Information output	*	*	*	*	Data object
6. Public/Private documents	*	*	*	*	*No comparable construct identified*
7. Defined goals	*	*	*	*	*No comparable construct identified*
8. Share documents	*	*	*	*	Associations and sequence flows

Table 3: Summary of salient decision making features identified from scenarios

Table 3 summarises the derived salient features of each given scenario and forms the basis for the research described in this thesis. By using our domain knowledge of each scenario, a list of features or decision-making activities that take place in at least one scenario have been identified.

Based upon Table 3, each feature was cross referenced with the BPMN 2.0 specification [116] to identify a comparable BPMN construct. The three similarities that all scenarios share is the concept of information output, that is, data that is produced over the course of the scenario, such as a risk presentation, policy document or a thesis. These concepts are comparable to the existing BPMN data object construct. Likewise, the multiple actors that appear in all of the scenarios are neatly defined in BPMN using the existing swimlane constructs. While sharing documents is symbolised with data associations, where documents are referenced multiple times, this requires multiple data associations which can be perceived as being visually unappealing.

For all other features, a set of extended BPMN constructs has been proposed in Table 4:

Proposed BPMN construct *(full definitions given in Sectioin 3.5)*	**Description**	**Implements feature(s) identified in Table 3**
Cost Swimlane	Modified Swimlane that represents increasing or decreasing cost of a scale with child nodes	1. Cost
Time Swimlane	Modified Swimlane that represents increasing time on a scale with child nodes	2. Time
Scoped and Workflow Data objects	Scope Data objects are owned by an actor and not accessible to other actors unless shared. Workflow data objects are accessible to all users.	4. Confidentiality, 6. Private data, 8. Sharing document
Weighted Paths	Weighted paths represent the predetermined preference of an outgoing edge	7. Defined goals

Table 4: Summary of the extended BPMN formalism constructs derived from the decision-making scenarios

Combining the extensions described in Table 4 with the existing BPMN 2.0 formalism, the resulting extended formalism is termed BPMN for Decision Making or BPMNdm.

3.5 Steps 2 to 4: defining BPMNdm extension elements

Using Table 4 and the methodology described in Section 3.2.2, the next section describes the development and definition of each extension identified.

3.5.1 Cost Swimlane

Cost is a scalar quantity that apportions a given value to an item, process or activity. In this work, cost is attributed to a financial value. In this context, cost is a vital part of any business process as it is a finite variable that is directly correlated with operating profits. Operating profits, specifically in the corporate world, and also in areas of resource management affect productivity and overall functionality of the business. Controlled, precise allocation of these resources optimizes the business workflow and maintains peak efficiency.

Integrating cost into BPMN is a natural choice. Being able to associate values to a given decision space compliments tool support. If a budget is limited, node traversal can be impacted by a cost value that is declared within the XML. Below we see the possible application and impact of such a strategy. This represents **Step 2** of our methodology for defining BPMNdm extensions.

Step 2:

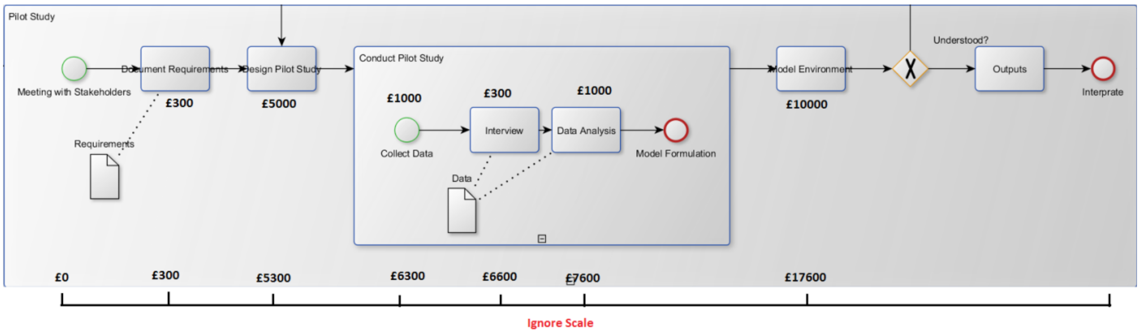

*Figure 29: Theoretical cost swimlane (**Step 2**)*

Relating to thrombolysis, we visualise the cost function as an assessment of resource management and optimisation. If budget is allocated on a per person per operation/consultation basis and the cost of the activity is documented, BPMN with tool support is an effective method of aiding the decision making problem with respect to budget. Tool integration would allow hospital staff to be informed of budget constraints and make decisions with this in mind.

Within the United Kingdom, every resident citizen is entitled to free health care via the National Health Service (NHS). Within the NHS, there is a specific protocol that depicts how treatment is funded and paid for. NHS treatment is payed from local practices' allocated budgets. Therefore, there is an incentive to local practices to minimise treatment costs.

This is a business process that naturally benefits from the introduction of cost into BPMN. Being able to document cost at the atomic level for a given patient in a given scenario allows accurate modelling and understanding of where costs are accumulated. This allows hospital managers to better allocate funds in an effort to save money and streamline the care of patients.

Cost integration would enable a detailed, accurate representation of financial expenditure. The scale in Figure 29 denotes a dynamic scale that updates in real-time to reflect the decisions made. This allows for a structured view of expenditure once you have exited the decision space.

Step 3: UML class diagram for cost

The metamodel provided in Figure 30, depicts a "`CostLane`" element inheriting from the "`SwimLane`" element with a string array of associated "`CostTask`" elements. The "`CostLane`" has one or more "`CostInterval`" attributes, which is accompanied by each "`CostTask`" element. Each "`CostInterval`" represent accumulated cost of all preceding "`CostTask`" elements as well as the current "`CostTask`" element is associated with. This can be used to validate the cost values when the model is run. A "`CostTask`" elements inherit from the "`Task`" element and have an additional attribute "`cost`" associated with them.

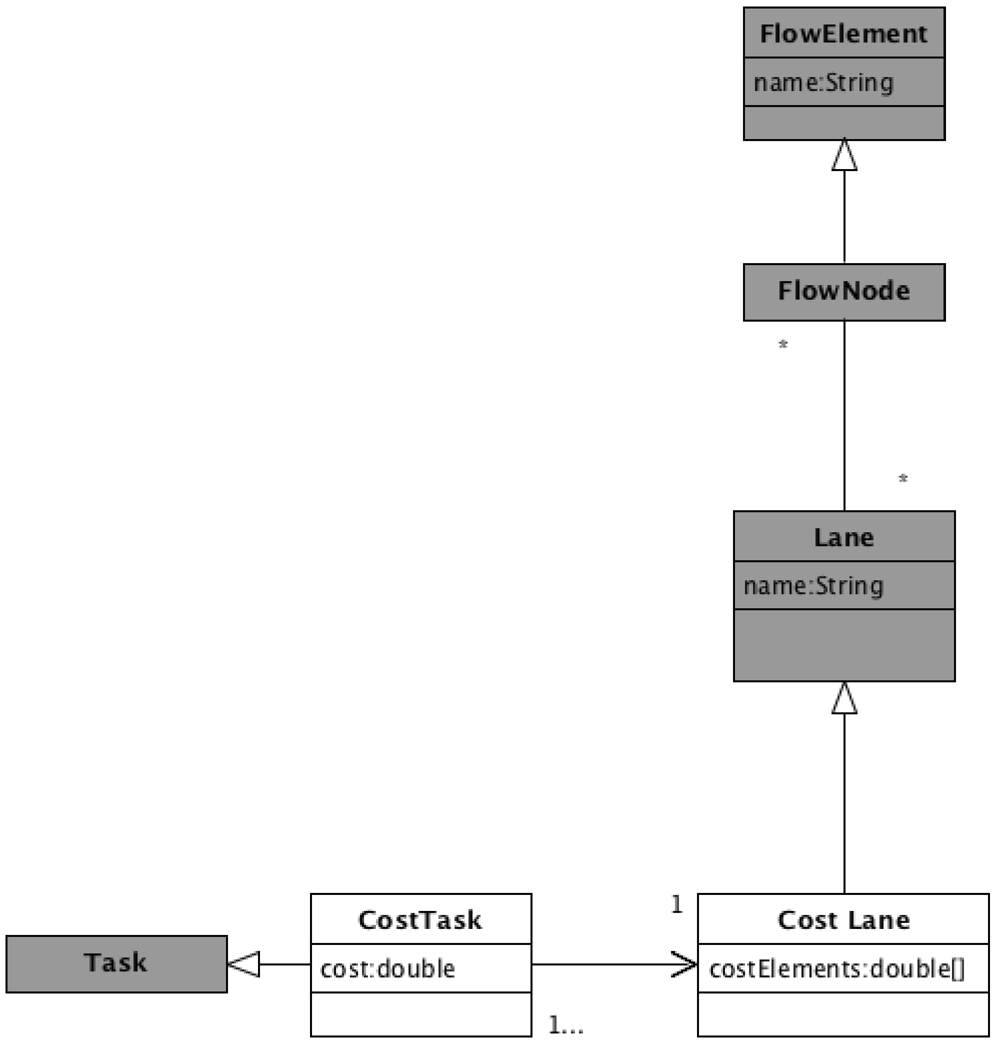

*Figure 30: UML class diagram for CostLine and Cost task elements **(Step 3)***

Step 4: BPMNdm XSD representing cost

Using the UML diagram in Figure 30, and the template XSD file in Figure 22, the following XSD was produced to represent the Cost Swimlane:

```
<?xml version="1.0" encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">



<group name="costline">

        <sequence>

                <element name="costInterval" type="double" minOccurs="1"
maxOccurs="unbounded" />
```

```
        </sequence>
</group>


<group name="costTask">
        <sequence>
                <element name="cost" type="double" minOccurs="1" maxOccurs="1" />
         </sequence>
</group>


</schema>
```

Figure 31: XML schema definition (XSD) for CostLane and CostTask elements (**Step 4**)

3.5.2 Time Swimlane

Like cost, we consider time to be a scalar quantity. As an attribute, time can be assigned to the tasks an individual or group does, which indicates the duration of the task. In Computing Science, time is a variable that can be manipulated in different scenarios. Time does not necessarily reflect the same time that is associated with the "real world" (i.e. 24 hours in a day). In CS1, time is used as a constraint in order to define the maximum amount of time a person can be treated with a thrombolytic drug from the onset of their stroke (4.5 hours). Therefore, for decision making, we assume that time is defined as a scalar quantity built upon the accumulation of individual tasks that have an individual duration of time. Time can also be defined as a constraint, such as the maximum time allowed for a process to complete.

To define these properties, we firstly define time as a value that represents the duration of a given task. We assume that the duration value for each task is independent of other duration values defined in other tasks. It is independent of any formally defined scale in reference to future paths that are not predefined. Furthermore, time increments are non-uniform. Nodes are not uniformly distributed as activities have different associated completion times. This provides a highly dynamic environment where time is not a constant progression.

Having defined the method in which time will be integrated into BPMN, it is necessary to discuss how this will be utilised. The application of time will allow for an assessment of productivity and efficiency in terms of the start of task and the end completion time of the task. From a modelling perspective, and further abstraction (system analysis), this will allow performance monitoring that has a plethora of real-world advantages. To discuss this, we must refer to our example scenario.

In a hospital environment, time (typical definition as denoted by real-time, real-world progression), budget and resources are intrinsically linked. Optimal and cost effective Patient care are imperative to both the success and purpose of the service. In both these cases we assume that budget and resources are a subset of time by deducing that budgets are allocated at service level (with time scale and investiture) and resources are dynamically distributed depending on demand (a value that changes with real-time, real-world time). With this assumption we can refer to an implementation of time into BPMN.

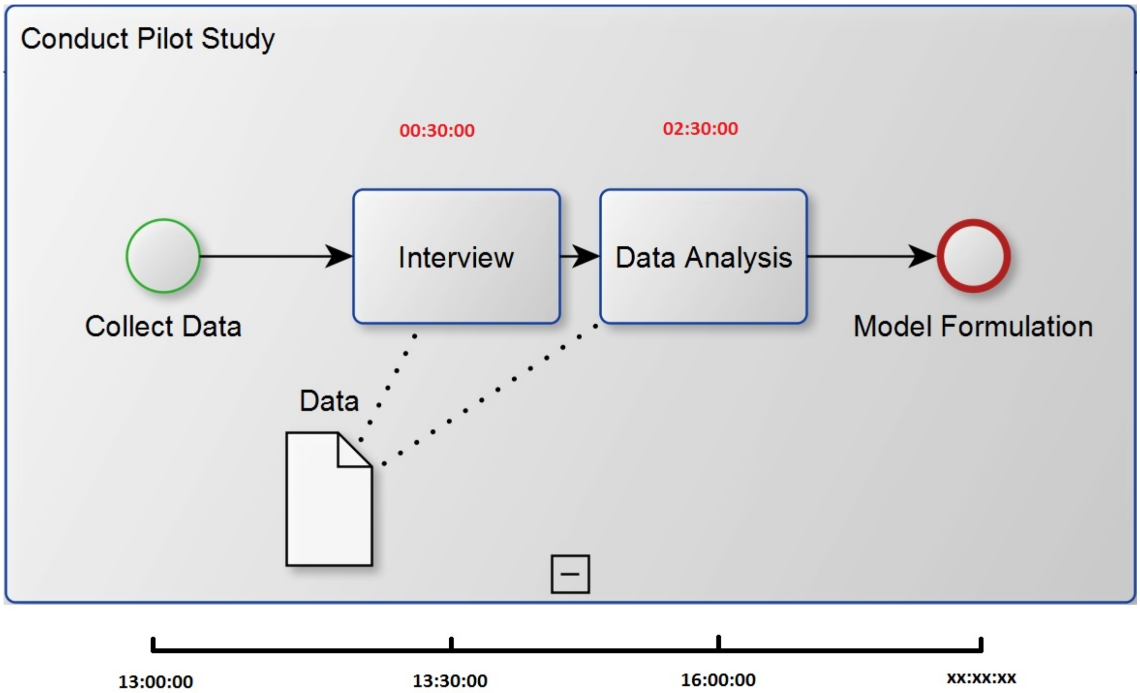

*Figure 32: Theoretical Time swimlane (**Step 2**)*

Step 2:

Figure 32 demonstrates the implementation of time but it is necessary to discuss the specific adaptation. For tool support we see immediate benefits in terms of the real-world real-time representation of the model. Specific routes are defined by their timing and subsequently plotted on a timeline where scale indicates the distribution of events. This can be further extended by dynamically updating time values relating to any chosen path. In essence, the timeline provides an evaluative data set to be analysed for optimisation purposes. To realise these extensions, it is necessary to provide a timeline that visualises a set of time intervals. Additionally, tasks that have specific time durations and are placed on a timeline require a time interval, representing the time duration of the task. This definition represents **Step 2** of our design methodology.

Within this example, a system designer may decide that node transitions should be affiliated with tool support. Subsequent route traversals are defined by given constraints (for instance a typical restraint may be that a patient cannot wait longer than 30 minutes).

Specifically, relating to decision points time is critical. Stroke victims have a small window where action is necessary in order to prevent further medical complication. Translating this to a physical environment we can assume a party of 1 doctor, 1 patient, 1 other member (possibly family/friend). Given this, our model would contain a run-time time variable that incrementally updates with node transition. However, this would be bounded. For example, a specific set of nodes (or task) cannot take longer than *x* minutes. Referring to the above example we can visualise a scenario where certain nodes are circumvented depending on the time. If the patient has too long to transition from the waiting room to consultations, it may be necessary for immediate medical intervention based on a doctor/consultant's approval. A system designer may infer this specific part of the scenario as an area where tool support would be beneficial.

As time progresses through the system, the optimal route for node transition also changes. This allows the BPMN to not only be a process model, but instead to dictate optimal transition based on predefined constraints. Based on these constraints, the system designer may infer that mobile tool support would allow these decisions to be made in a mobile environment increasing productivity and potentially availability. This type of tool suggestion could also be specified based on the design pattern of the workflow model as well as considering the time constraints.

Step 3: UML class diagram for time

It is this flexibility that makes adding time to BPMN an important feature. The metamodel for a timeline element is provided in Figure 33. We have defined two extensions to the BPMN formalism. Firstly, there is a `TimeLine` which extends the existing BPMN lane construct. A

`TimeLine` contains the attribute `timeElements`, which has an array of individual time variables.

The second extension is the `TimeTask`, this inherits from the Task element and has two attributes: `startTime`, indicating when the task starts and duration, which indicates how long the task lasts. One or more `TimeTasks` are associated with one `TimeLine`. The UML metamodel for these extensions is detailed in Figure 33.

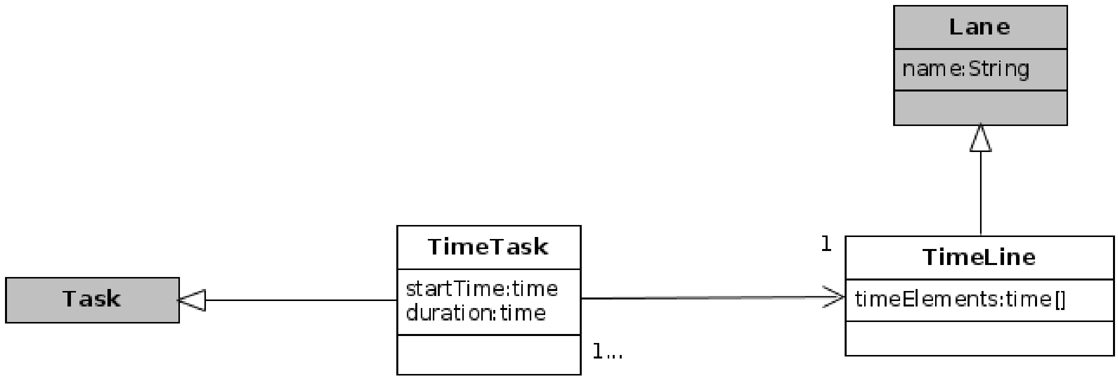

*Figure 33: UML class diagram for Timeline (**Step 3**)*

Step 4: BPMNdm XSD representing time

Using this UML diagram in Figure 33 and the XSD file in Figure 39, the following XSD was produced:

```
<?xml version="1.0" encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">


<group name="timeline">

        <sequence>

                <element name="time" type="time" minOccurs="1" maxOccurs="unbounded" />

        </sequence>

</group>


<group name="timeTask">

        <sequence>

                <element name="startTime" type="time" minOccurs="1" maxOccurs="1" />
```

```
              <element name="duration" type="time" minOccurs="1" maxOccurs="1" />
</sequence>


</group>


</schema>
```

*Figure 34: XML schema definition (XSD) for Timeline (**Step 4**)*

In Figure 34, the `TimeLine` UML class is represented by the `TimeLine` group, with a sequence of one or more time elements. The Time Task is represented by a sequence of one `startTime` element, representing the time that an individual task starts and one `duration` element, representing how long a task will take. In this example, the modeller is required to specify the time format and the unit of time that represents the duration of the task (I.e. seconds or hours).

3.5.3 Workflow and Scope data

In this section, we discuss how we implanted workflow and scope data. As the principle difference between workflow and scope data is that scope data is considered data that is associated with a user, therefore is private, while workflow data is considered accessible to all tasks and users defined in the model, therefore the only distinction between the two types of data is the indication if a data item is scoped data (termed "scoped") or not. The following subsections discuss Workflow and Scoped data in greater detail. The BPMN workflow models developed using these descriptions represent **Step 2** of our methodology.

<u>*Workflow data*</u>

Russel et al. define workflow data as: "*Data elements are supported while are accessible to all components [for example, the flow objects in a BPMN workflow diagram] in each and every case of the workflow [for example, the runtime instance of the BPMN workflow diagram] and are within the control of the workflow system*" [118]. In decision making contexts, workflow data can be considered public data that is accessible to many/all in the decision making environment. Examples of this include documents that are shared between participants or files on a server that are accessible to all users. The BPMN formalism does not support the representation of workflow data as the existing data object does not specify the ownership of the document [119]. In the context of decision making, using the BPMN data object, workflow data could be represented by data objects that are not explicitly connected to any other object and are outside of any swimlane or similar construct.

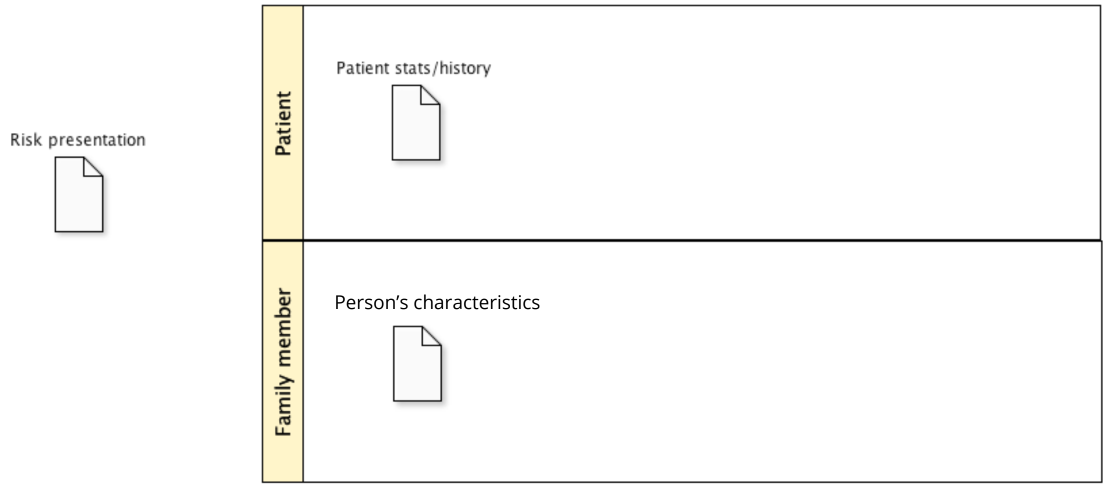

Figure 35: Workflow data object (Risk presentation) and Scope data objects (Patients stats./history, Person's characteristics) ***(Step 2)***

Figure 35 demonstrates the concept of workflow data (Risk Presentation) seen alongside Scope data (Patient stats/history and Persons characteristics). The definition of workflow data follows Russel et al's definition with all given tasks and processes having access to the object.

<u>*Scope data*</u>

Scope data is defined as "*Data elements [that] can be defined which are accessible by a subset of the tasks in a case*" [118]. Scope data is useful in a decision-making context as this notion can be used to account for private artefacts that a user or group possesses.

Private artefacts could be defined as anything that holds information, such as an address book, report or anything else that can be considered private to a user or group. Private artefacts should be defined by the system designer, based on the real-world status of the artefact, which could be obtained using ethnographic approaches such as meeting observations, or asking the stakeholders about the status of each artefact used in the scenario. BPMN does not support scope data [119]. In order to visualise scope data, data objects should be placed in swim lanes or visually grouped with objects in order to imply where the object belongs.

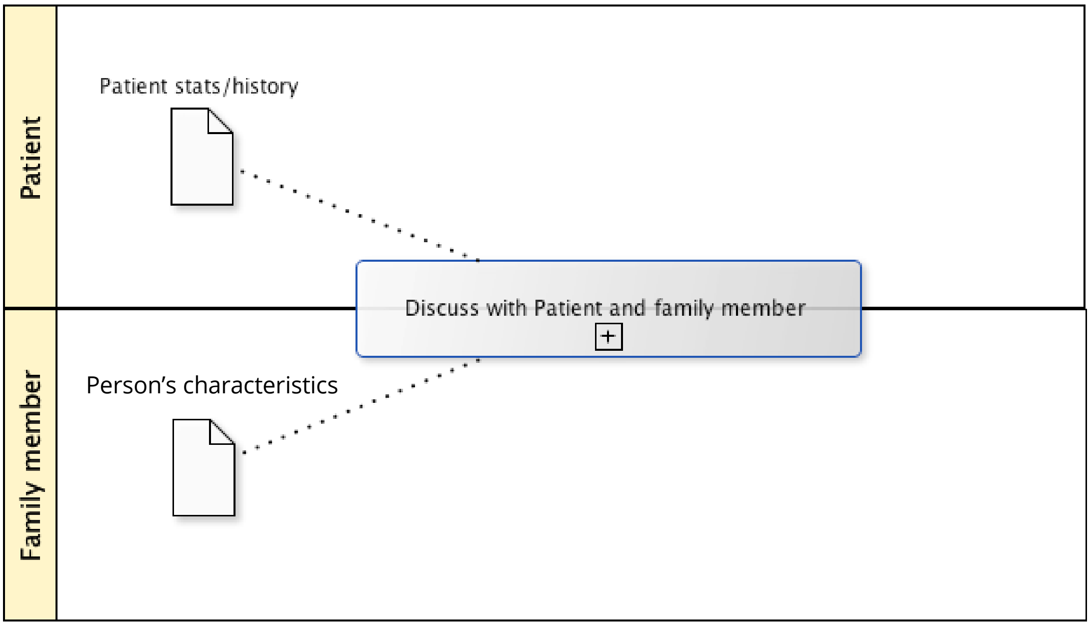

*Figure 36: Scoped data objects (**Step 2**)*

Figure 36 shows the use of scope data and how it can be passed from one subset to another. In the example, Patient Stats/History and Persons characteristics are data objects that belong to two actors, a Patient and a Family Member respectively. Scope data is only visible to the actor or group it belongs to unless the object is passed to another process as part of a flow. An example of this is the consultation process between a clinician, patient and the family member, which involves discussing the patients characteristics as well as adjusting the course of the consultation to minimise distress to the patient and family member by the clinician judging the emotional states of both of the other actors.

Unless the object is explicitly duplicated by the clinician (i.e. writing down the necessary information), the clinician will retain the object for the length of the task it is assigned to, only after which it is assumed it is passed back to the sender process or destroyed (i.e. the document is physically destroyed in the real-world and removed from memory in the model runtime engine).

The metamodel for Workflow and Scope data is shown in Figure 37. This represents **Step 3** of our methodology:

Step 3: UML class diagram for Scoped data

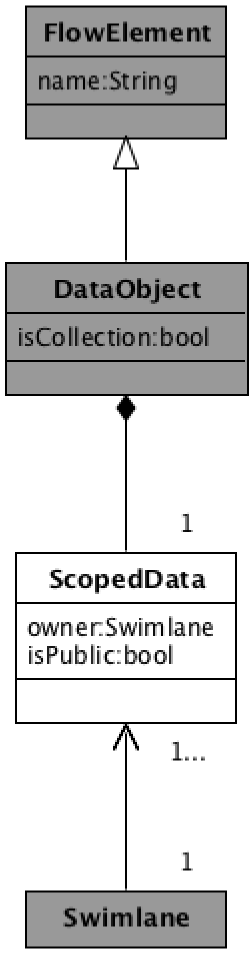

Figure 37: UML class diagram for Scoped Data ***(Step 3)***

In order to simply the name of our extension, Workflow and Scoped data is represented with a single `ScopedData` element. Each Scoped data item contains two key attributes, the `owner` swimlane the data item belongs to (or *none* if the item is a Scoped Data resource that is shared by all participants) and a `isPublic` attribute, which signify if the item is publicly accessible to all other participants.

Step 4: BPMNdm XSD representing scoped and workflow data:

Using the UML diagram in Figure 37, the following XML schema definition was produced:

```
<?xml version="1.0" encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">


       <group name="scopedData">

          <sequence>

             <element name="owner" type="string" minOccurs="1" maxOccurs="1" />

             <element name="isPublic" type="boolean" minOccurs="1" maxOccurs="1" />

          </sequence>

       </group>


</schema>
```

Figure 38: XML schema definition (XSD) for Scoped Data object ***(Step 4)***

Figure 38 shows that the attributes `owner` and `isPublic` defined in Figure 37 are mapped directly to the elements owner and `isPublic` in the group “`ScopedData`”.

3.5.4 Weighted paths

The BPMN 2.0 formalism does not provide any method for defining the likelihood of node traversal after a gateway. Whilst this is satisfactory for defining a process model, it can be improved to provide more detailed knowledge of a given process by visualising preferred sequence flows based upon weighting. Figure 39 is an example of weighted path with the most likely route highlighted with a bold arrow and the probability value is provided. The diagram presented in Figure 39 represents **Step 2** of our methodology.

Unless an outgoing sequence flow is indicate as the default flow, the BPMN 2.0 formalism does not specify how to visualise sequence flows based upon probability or preference scores. When using probabilities to determine an outgoing sequence flow, a weighted path can be seen as a virtualisation of the likelihood with which the next node is chosen. There are several methods in which this can be implemented.

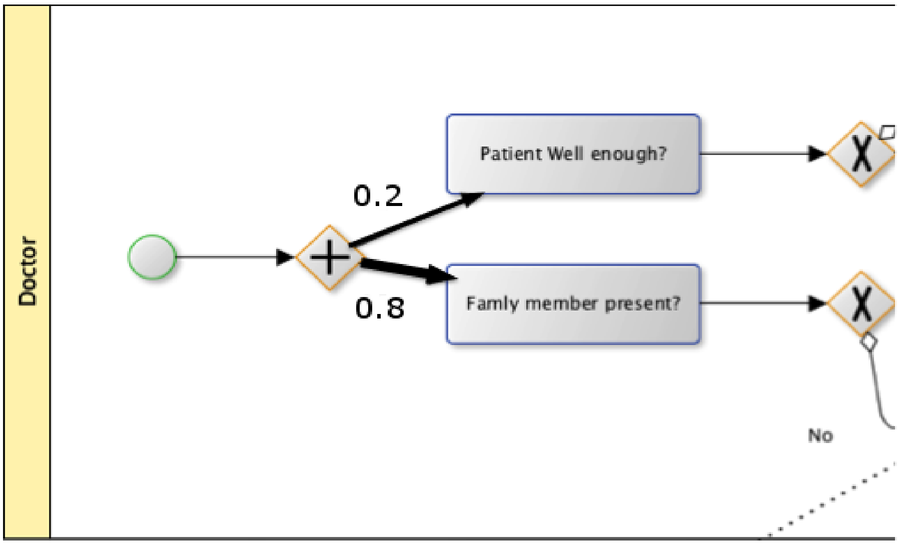

*Figure 39: Weighted path (**Step 2**)*

Firstly, if data relating to the target environment is available it will be possible to highlight particular trends or patterns in either the users behaviour or how the system reacts to given stimuli. This likelihood can be statistically represented with a value between 0-1.

In the case of the thrombolysis, there is a plethora of applications. For example, weighted paths can be adopted to visualise the outcome of treatment based on previous patients, or visualise a preferred care pathway. The logic to determine this would also embedded in the corresponding gateway, as these are used to decide which outgoing sequence flow is used. Figure 39 illiterates the point at which a doctor will seek consent for administrating thrombolysis. In a typical instance, the patient is unlikely to be sufficiently conscious to offer consent, therefore consent from a relative tends to be the most likely path. As drug trials are extensively tested and quantified it is a simple process to incorporate this probability into the XML. Patients will also have a record that can be visualised as a data object. In essence, this is a predefined document stating all medical knowledge. This data is incredibly useful in determining how treatment should progress and is simple to incorporate in a model based approach. In time critical scenarios such as this, it allows for optimisation of node traversal. This is both a time and cost saving feature as it is possible to plan future activities and processes based on a given probability. Being able to inform staff or potential surgery or general hospital logistics will greatly optimise the productivity of the process. Having such data available on a tool would empower medical staff to make these decisions in a mobile environment where decisions are based on prior knowledge and quantified statistical likelihood.

Financial assessment is also an area where this notation change is beneficial. Via resource management and finance allocation it would be possible to improve those services that are most commonly used in an effort to streamline and management patient throughput. Resource allocation would be a natural conclusion to this.

Step 3: UML class diagram for WeightedPath

The metamodel for Weighted paths is provided in Figure 40.

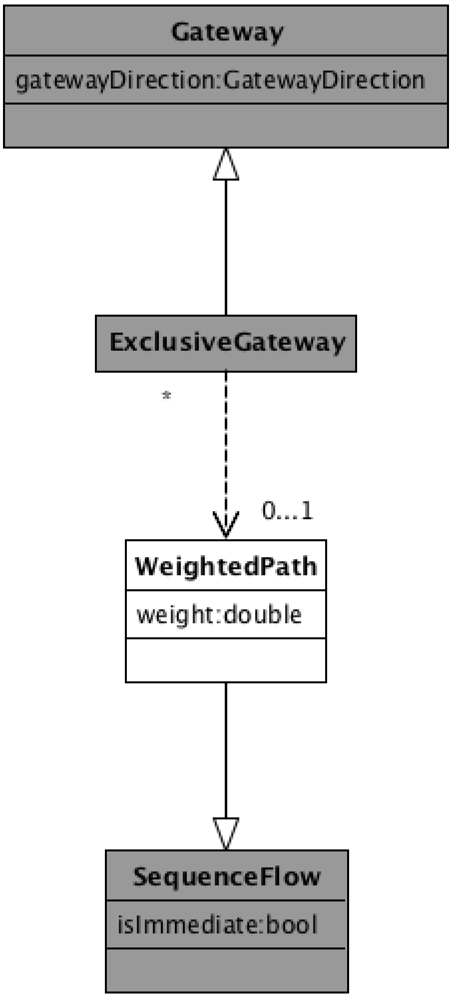

*Figure 40: UML class diagram for weighted paths (**Step 3**)*

The UML class diagram in Figure 40 shows that the `WeightedPath` extension inherits from the BPMN 2.0 sequence flow object. A `WeightedPath` contains the attribute `weight`, specified as a double number, and zero or more weighted paths are associated with an exclusive gateway.

Step 4: BPMNdm XSD representing weighted paths:

Following the same procedure outlined in Section 3.2.1, using the UML diagram depicted in Figure 40, the following XML schema definition was produced:

```
<?xml version="1.0" encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">


<group name="weightedPath">

        <sequence>

        <element name="weight" type="double" minOccurs="1" maxOccurs="1" />

        </sequence>

</group>


</schema>
```

*Figure 41: XML schema definition (XSD) for Weighted Paths.(**Step 4**)*

Figure 41 shows the definition of a `WeightedPath` group, representing the extension, which contains the attribute `weight`. This conforms to the UML class diagram in Figure 40.

3.6 Discussion

We have proposed a set of extensions to the standard BPMN 2.0 specification that would help model decision-making scenarios with the aim of providing recommendations of appropriate tool support to enhance the decision-making scenario. Using three scenarios as a starting point, we have devised a list of requirements that a BPMN 2.0 extension should fulfil in order to support the appraisal of decision support approaches for decision making scenarios: Time, Multi Actor, Confidentiality, High Risk, Defined Goals and Outputs. The extensions developed to support these activities are Scoped/Workflow document elements, weighted paths, cost and time swimlanes. Our extended BPMN formalism is named BPMN for decision making or BPMNdm. We have also modified the methodology described by Stroppi et. al. for developing our extensions which is summarised as follows:

Step 1: Construct BPMN workflow models for decision making scenarios using the existing BPMN 2.0 formalism and identify desired decision making activities

Step 2: For each decision making activity, build a visual prototype of how the decision making activity should be visualised by using existing BPMN components and visualising new components where necessary.

Step 3: For each visual prototype, construct a UML class diagram, indicating the extension element and it's relationship to the existing BPMN elements in the formalism. This visualisation is the metamodel for the extension.

Step 4: Implement the extension as a XSD file that can be used in BPMN workflow models.

The need for the provision of tool support appraisal is the increasing usage of decision-making tools in many decision making domains. In the medical domain alone, there is an increased usage of decision making tools that are delivered via manual boards, software loaded on a desktop computer but increasingly on smartphones and tablet PCs. Each device and platform has different strengths and weaknesses in regards to their features and appropriate in different contexts and environments.

In the case of acute stroke treatment, BPMNdm would be able to model the decision making encounter as well as capture and store contextual information about both the environment, artefacts and users. As indicated in Table 3, it is possible to identify and classify a set of basic attributes of devices that could be potentially used as decision support tools. Using contextual information that is collected at design time and possibly run time, BPMNdm would be able to offer recommendations of what tools would be most appropriate for supporting the decision making encounter, or none if appropriate. This has the potential to benefit system designers as it would offer suggested tools that would be appropriate for the given scenario.

An example of each extension being used in a BPMN workflow model is provided in Chapter 6.

Chapter 4 - Person-to-person interaction rules

4.1 Introduction

In a BPMN workflow model, gateways are used to control the sequence flow of the model. As discussed in Chapter 2, gateways can be considered the representation of decision making activities, as they symbolise an option being chosen and define which sequence flow will be executed when the model is run.

When considering decision making scenarios, a modeller may be interested in simulating complex activities between multiple actors, such as conflict resolution during a scenario that includes a dispute. The extensions to the BPMN formalism in Chapter 3 provide visual and semantic representations of four salient decision making activities: scoped data, weighted paths, time constraints and cost constraints. If a modeller wishes to model activities with more granularity, one option is to provide additional logic for the runtime engine to evaluate. This is useful when the desired activity is deemed too complex to be visualised on the workflow model, such as if the activity involves many individual sub tasks, conditions or would produce a model that was visually cluttered, undermining the visual affordances the workflow model may have.

A limitation of workflow modelling formalisms with sequence flows between tasks is that this design approach is unsutuable for activities that occur spontaneously in the scenario. For example, if a workflow model was constructed showing a meeting between three participants, and each were given a turn to discuss a random topic, the idealised workflow model would be three tasks representing each person talking. This scenario is depicted in Figure 42. If the modeller was designing a tool to support this scenario, one of the purposes of the tool could be to mediate the discussion, by for example setting a time limit on how long each person can talk. One problem with the workflow model in Figure 42 is that it does not take into account any possible interruptions, for example if a person asks the current speaker a question on the topic they are discussing. If the workflow model was produced based on the observations of several group conversations, it cannot be guaranteed that every conversation would have an interruption, or when it would happen.

A solution to this problem is to use one of the workflow flexibility patterns by Schonenberg et al. [120] described in Chapter 2. In this thesis, we use flexibility by design and BPMN intermediate throw/catch events to handle specific conditional sequence flows that cannot be guaranteed to occur in all instances of the workflow model.

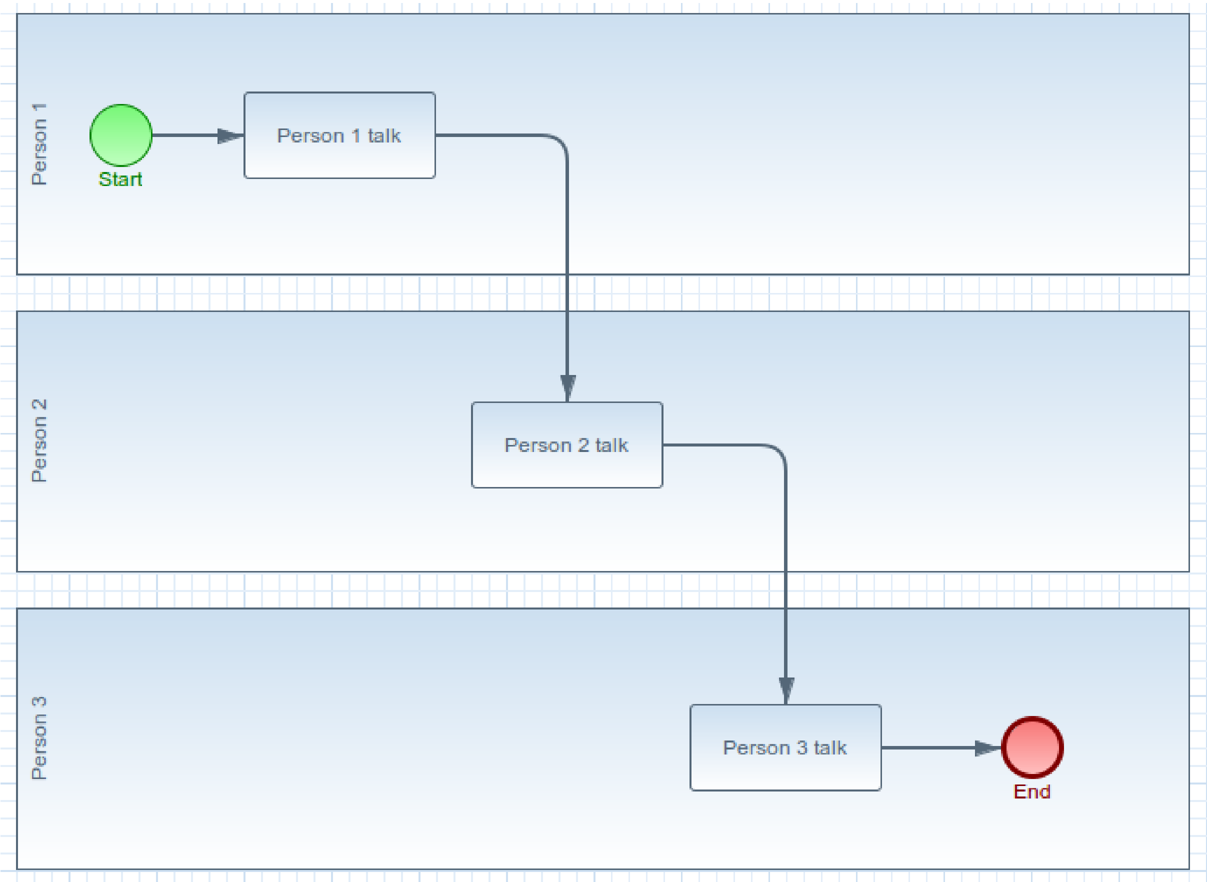

Figure 42: BPMN workflow diagram showing three people talking in turn

Figure 43 depicts the addition of a catch event and associated tasks representing a person asking a question and the current speaker answering back. This sequence flow is not placed in a swimlane as it indicates that any person can ask a question at any time. In order to simulate a question being asked, the modeller must write executable code in the task to trigger the question event when desired.

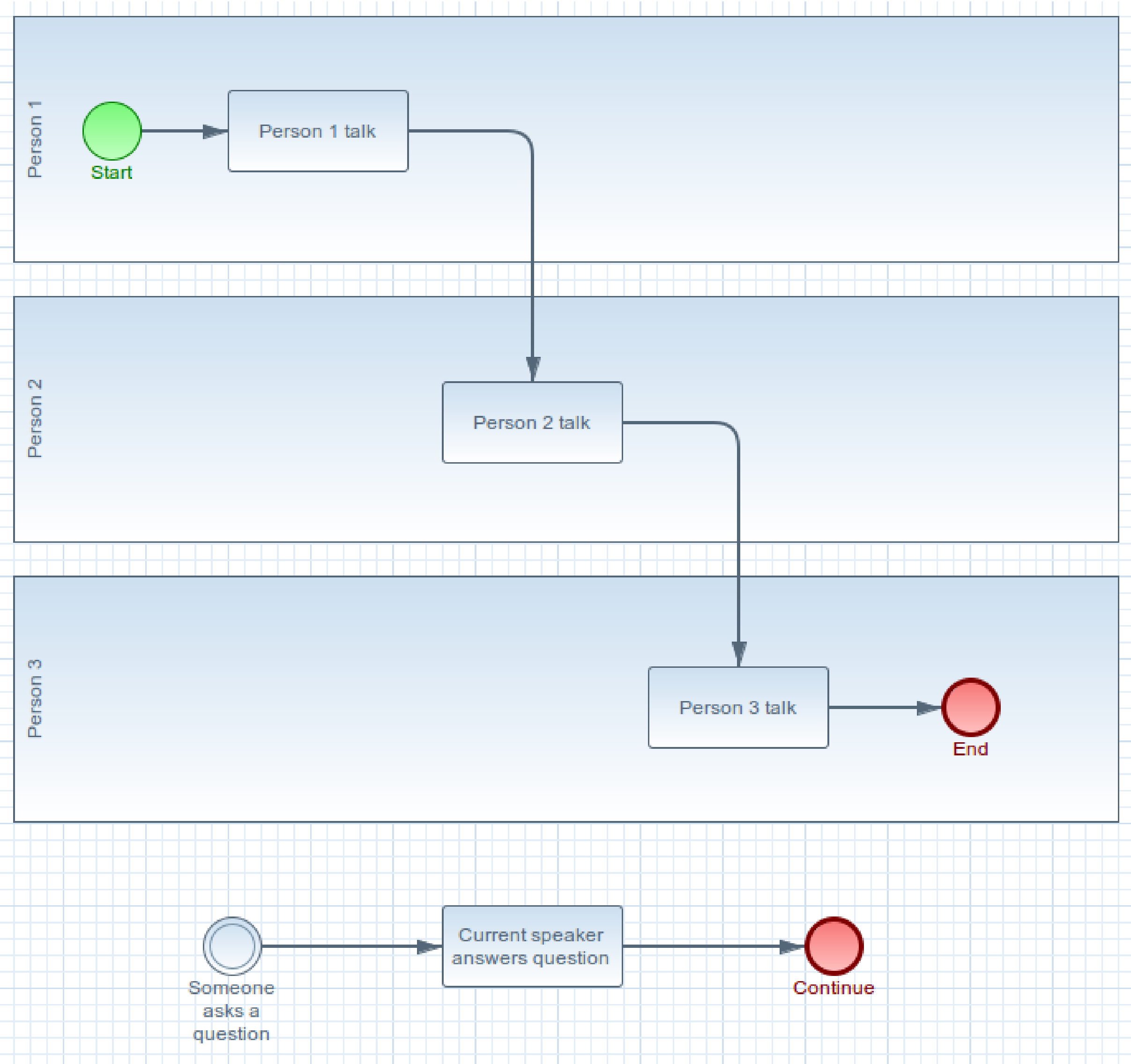

Figure 43: BPMN workflow diagram showing a catch event representing a question being asked.

While the BPMN diagram specifies and visualises the sequence flow for handling a question, the modeller needs to embed the necessary code into the workflow model in order to define the necessary control logic for specifying the behaviour of the gateways in a BPMN workflow model [31], [41]. jBPM 5 supports the Java language for defining the sequence flow at any given gateway in the model [31].

An alternative to using Java for the logic of a gateway is Drools, a business rule management system that provides a high level syntax for defining constraints and conditions that would normally be represented using `if` and `switch` statements in Java [121], [122]. Using Drools has the advantage of abstracting the logic governing the behaviour of a BPMN gateway or task from the BPMN formalism as well as the underlying jBPM workflow engine, allowing the rule to be re-used elsewhere in the process, or for a different process [121].

However, for the Drools rule to be a meaningful representation of the activity, it should reflect what was observed in the actual real world scenario, such as capturing the event that lead to the interaction, for example if a person did not understand the topic being discussed, the conditions that constitute the overall interaction, as the person asking the question and the person's role in the meeting (i.e. the chair person asking the question) and the resulting actions performed in response to the question, such as the current speaker answering the question.

Therefore, we have constructed person-to-person Drools rules, based upon data in the form of a case study that involved the observation of eight MAPPA (Multi Agency Public Protection Arrangements) meetings. The observation notes were transformed into Drools rules using a three stage process:

1. Using thematic content analysis to identify key themes identified in the meeting notes and any attributes that constituted to each theme.

2. Specify a rule from each theme by expressing the theme using first order syntax to identify the logic and requirements of each rule.

3. Convert the first order logic into a drools rules by translating the conditions into Drools conditional syntax.

The rest of this chapter is structured as follows: a description is provided of the MAPPA process, and a summary of the meetings are proceeded. Next, a summary is provided of thematic content analysis as well as the process of transforming qualitative data into drools rules. Finally, for each key theme identified, a description is provided using the three stage process outlined above.

4.2 Research methodologies

This section discusses the research methods for collecting the observational data from the observations of the MAPPA process. This is followed by a discussion on how the data was analysed and categorised to allow the identification of interaction activities (also termed themes) using thematic content analysis. Finally, this section describes how the resulting themes were converted into Drools rules using the process described by Kaiser et. al. [123].

4.2.1 Collecting data produced by ethnographic methods

The method chosen to collect the observational data was to directly observe the meetings. The meeting observer conducted their observations strictly as a "Pure participant observation" [124]. This entailed the investigator assuming the role of an observer in the meeting venue without participating in the affairs of the meeting. This was necessary as the observer is not authorised or qualified to participate in the meeting and the observer is unfamiliar with the meetings, the subject matter and the participants. Furthermore, the presence of the observer should not influence the proceedings of the meetings as much as possible. This method was also adopted to reduce any possible bias in the decisions made for each case [124]. The MAPPA meetings deal with confidential information about offenders and other people who are potentially vulnerable, such as victims. Therefore, in order to simplify the data collection process and maintain confidentiality, the observer used a password protected laptop computer to directly type the observations during the course of each meeting. All meeting participants were breifed before the meeting started and any participant could ask the observer to leave at any point if they wished. The author of this thesis was the observer.

Documentation strategy

The documentation strategy follows the salient points described by Jorgensen [125] including recording the room layout, what were the meeting participants roles and statuses in the meeting (i.e. meeting chair, minute taker etc.) and observing the activities, triggers, discussion points that lead to the group deciding if the offender is, or continued to be a MAPPA case.

In order to simplify the note recording process, the investigator used a laptop computer to type the notes during the course of the meeting.

All artefacts used or produced in the meetings were also noted. The definition of an artefact is based on Beyer and Holtzblatt's contextual design principles [9]. They define an artefact as: "People create, use and modify things in the course of doing work. The things they use become artefacts, like archaeological findings" [9].

For the purposes of this study, artefacts are defined as the items such as documents that the participants either bring or produce during the course of the meetings. These artefacts can either be physical (such as a note on a piece of paper) or virtual (such as a meeting invite or a word document).

Documents are useful as they provide additional contextual information about the meetings. They also provide a written record of an event and are likely produced long before the study has commenced [124].

When documenting each meeting, the following principles were followed when recording information:

Before each meeting started:

1. Documenting the meeting room layout, specifically if there was any computerised equipment in the room, such as a PC or overhead projector.
2. A diagram was produced of the table with each participant labelled according to their seating position and the organisation they represented. If a participant had a special role (in the MAPPA meetings, this was the chairperson and the minute taker), this was also indicated on the diagram.
3. If any participant brought documents to the meeting, a summary of the document was recorded. The summary included the type of document (i.e. agenda), if the document was based on a template or was handwritten and if the document was edited.
4. For any document, it was recorded if the document was a personal document, or shared with others in the meeting, or placed in a location that was accessible to other participants.

During the course of each meeting:

1. The observation notes were separated into sections according to the current stage of each meeting. A stage was identified when the chair person indicated that the meeting would move onto a specific section. In the MAPPA meetings, the chair was guided by the meeting agenda. This clearly indicated when a new section was to begin.
2. For each part of the meeting, attention was paid to the activities that were undertaken by the group in general. An activity was a sequence of discreet tasks done by an individual. It was noted how the activity was triggered (such as if a person was asked

to start the activity), who did the activity, what the activity was and if there were any interruptions from other participants and why the interruption occurred.

3. If any documents were created, edited or passed to other participants, this was also recorded.

4.2.2 Analysis strategy

At the end of the data collection phase, a number of meeting observation documents were produced. These document contained notes that were based around the observations of activities of interest discussed in Section 4.2.1. These notes contain mostly qualitative data, such as descriptions of the activities undertaken at a given point in a meeting.

In order to begin to derive the activities and attributes that determine how the meeting participants interacted in the meetings, it is necessary to analyse the qualitative data in order to first identify these activities and secondly to identify any further activities or attributes that constitute a person-to-person interaction from the meeting notes. In this thesis, we use thematic content analysis [126], [127] to analyse the meeting notes in order to group the qualitative data into key themes. Thematic content analysis is a process of categorising qualitative information based on coding the information contained within the meeting notes. Once the information is coded, it is possible to categorise the information around common areas of interest, also termed themes [126].

The generation of themes is dependent on how well a section of data relates to the original research question and represents a pattern that can be identified in other related sets of data elsewhere in the data source. For the purposes of this thesis, the desired themes are based around person-to-person interactions.

The methodology of thematic content analysis used for this chapter was defined by Braun and Clarke as [126]:

1. Familiarisation of data set:

Before the analysis begins, the data should be transcribed if necessary, then read by the researcher, with any initial ideas or comments on the contents of the data being recorded as necessary.

2. Generating initial codes:

A code represents a discrete feature inferred from the data set. A set of codes should be systematically generated throughout the data set.

3. Searching for initial themes:

Initial themes are produced by collating relevant and related codes into potential themes.

4. Reviewing initial themes:

Once initial themes have been generated, the researcher should ensure that the themes realistically relate to the initial codes. A thematic map is produced to visualise the conceptualisations developed, and their relation to the dataset.

5. Defining and naming themes:

Themes should be continually refined until the specifics of each theme are clearly defined and each theme can be named.

6. Production of report:

For each theme reported, there should be clear examples from the data set that support and justify the theme being discussed, with support from literature where appropriate. For this thesis, the report is considered this chapter where the themes are discussed in Section 4.5.

4.2.3 Methodology from generating Drools rules from specific themes

The methodology adopted for generating a set of Drools rules follows the methods described by Kaiser et. al. [123]. Their conference paper describes a computerised audiovisual virtual director system for recording group communication scenarios. Their system takes multiple video streams of the scenario and uses a set of rules to choose the optimal video source to capture the communication activities taking place. This is similar to the activities a human video director would undertake to record an interaction between multiple people, such as showing a person talking and cutting the camera to another person who is asking a question.

Before the rules were generated, video data was automatically analysed to identify activities or cues that occurred in the environment, and their corresponding meaning. For example, a cue could be an individual glancing at another individual in the room, indicating that it was expected that the other person would start to talk. This process of analysing cues is similar to using thematic content analysis on a qualitative data set, as the end results is a set of themes with an associated set of concepts that are linked to each theme.

Using the themes produced using thematic content analysis, we used the classifications defined by Kaiser et. al to transform each theme to form a primitive definition of a rule:

- **Events:** these are items that are described in the content analysis that indicate there will be a trigger of a specific sequence of events. Kaiser et. al give an example of a Voice Activity occurring, where a person starts talking, which is identified using content analysis of the video stream. Each event can be defined as a single trigger, or my be composed of more than one events.
- **Conditions:** these are the state of the interaction at a given time frame. Using the Voice Activity example, a condition will be if the person perform the Voice Activity is the current person expected to talk, or if the camera is pointed at the current person talking.
- **Actions:** these are the specific events that are undertaken to deal with the event and associated conditions. This can be considered as the ultimate composition of events that result in a final action being performed. In the Voice Activity example, this would indicate to the virtual director system that it will need to change the camera view to the speaker. We deviate from Kaiser et. al. by not providing formally defined actions in the rule. Instead, actions are delegated to the BPMN workflow model that uses the rule to specify the sequence flow.

Using the classifications above to form primitive rule definitions from the data, each primitive rule was formally defined using First Order Logic (FOL) like syntax. The syntax used [128] is summarised in Table 5:

Symbol	Description
$\wedge$	Logical AND
$\vee$	Logical OR
$\neg$	Not
$Talk(P_1)$	Formula (*Talk*) with parameter (P_1)
TimeTaken	Variable
$\rightarrow VoiceActivity(P_1)$	Function (*VoiceActivity*) with parameter (P_1)

Table 5: Summary of First Order Logic syntax used

Using the classifications for the Voice Activity, the primitive rule can be defined as:

$$Talk(P_1) \wedge isTurn(P_1) \rightarrow VoiceActivity(P_1) \quad (1)$$

In the example above, *VoiceActivity,* on right hand side of the arrow, represents an event based on the composition of two sub events on the left hand side of the arrow, $Talk(P_1)$ and $isTurn(P_1)$. $Talk(P_1)$ represents an event of a predefined person P_1 talking. $isTurn(P_1)$ indicates if it is the turn of P_1 to participate in the scenario. Both events would return *true* or *false* dependant on the state of the scenario. All of these events are associated with a Person (P_1) which is analogous to a variable in the runtime engine mapped to a meeting participant.

If the *VoiceActivity* event formed part of an action, for example to instruct the camera to pan over to the person talking, the following example depicts how this would be done:

$$VoiceActivity(P_1) \wedge \neg CameraAt(P_1) \rightarrow PanCameraTo(P_1) \quad (2)$$

In the example above, the action is composed of the event $VoiceActivity(P_1)$ and the condition $CameraAt(P_1)$ indicating that camera is not pointing to the person P_1 performing the $VoiceActivity(P_1)$. If the person is performing the voice activity, but the camera is not pointing at them, this would trigger the action instructing the camera to pan to the person.

In the case of developing the person-to-person interaction rules, the Events are directly analogous to the BPMN throwing event elements and tasks where applicable, with Actions being the resulting sequence flow of tasks that are undertaken starting with the BPMN catching event or diverging sequence flow.

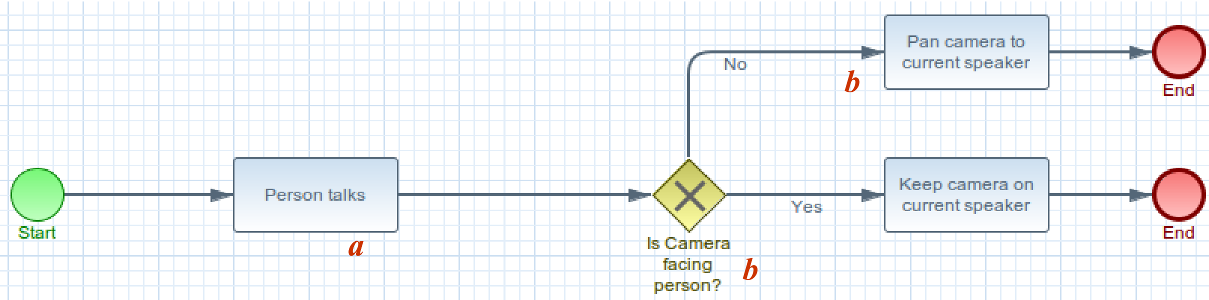

Figure 44: BPMN diagram encapsulating VoiceActivity rule

In the example workflow model in Figure 44, the *VoiceActivity* event shown in Equation (1) is analogous to the elements indicated with "***a***", with the "Person talks" task being similar to the Talk event. Equation (2) would be mapped to the elements indicated with "***b***" on the diagram, represented by the gateway "Is camera facing person" assessing if the camera is pointing at person talking, and if not, panning the camera to the person.

The methodology of producing a Drools rule source files developed from the FOL syntax and used as part of the jBPM 5 workflow engine is as follows [121], [122]:

1. The first lines indicate the location of the file in the project package folder.
2. The second set of lines import any Java classes or packages that are used.
3. The third set indicates the language dialect used for the actions. For this thesis, we use Java as the language dialect, as that is the default language for jBPM.
4. The fourth set of lines provide the logical statements *when* the rule triggers. These are written in Drools syntax [121], [122]. When evaluating fields from Java objects using getter methods such as `getName()` on a person object, the Drools engine automatically affixes *get* onto the method call, therefore the statement can be shorted to `Name()`.
5. The fifth set of lines define the actions that the rule should perform if the statements in the *when* section evaluate to true. This is known as the *then* block.

Using Equation (2) and the conventions above, we can construct the following Drools rule:

```
package uk.ac.ncl.cs

import org.drools.runtime.StatefulKnowledgeSession; //Import all Java
objects used
import uk.ac.ncl.cs.Camera
import uk.ac.ncl.cs.Person

rule "PanCameraTo"
dialect "java"
    when
     p1: Person( activity == "Talking" ) //VoiceActivity event
     c: Camera( cameraAt != p1 ) //CameraAt condition
    then
        kcontext.getKnowledgeRuntime().setGlobal("PanCamera", true);//Pan
Camera

end
```

Figure 45: Drools rule developed using PanCameraTo activity

The example Drools rule for the *PanCameraTo* activity depicted in Figure 45 shows the usage of a `Camera` and `Person` object. The *when* block checks if the person is currently talking. This calls a `getActivity()` method on the person object instance `p1`. The rule also evaluates if the camera (assigned to the variable `c`) is not pointing to the person by calling the `cameraAt()` method. If both of these conditions produce true, the *then* block is evaluated. This causes the jBPM system to set the global variable "`PanCamera`" which is embedded in the workflow model to `true`, which can be used by the gateway in Figure 44 to determine the appropriate sequence flow. In this case, if the *then* block is executed, the "No" sequence flow will be chosen.

The two statements in the *when* block are directly analogous to the *VoiceActivity* event and *CameraAt* condition in Equation (2), with the overall rule file representing the action *PanCameraTo*.

4.2.4 How the development of person-to-person interaction rules is presented.

Using thematic content analysis and Kaiser et. al's methodology for producing Drools rules from observational data, we describe the development of Drools rules encapsulating person-to-person interactions in the following way:

Firstly we give an overview of the MAPPA meeting observations, which includes a description of the MAPPA process, plus a discussion of the meetings observed. BPMN workflow diagrams are provided to illustrate the activities undertaken as part of these meetings. The overview is provided in Section 4.3.

Next, we describe results of using thematic content analysis on the meeting observation notes, with the aim of identifying themes around interactions as well as generic meeting activities that were performed in the meeting. The methodology for this is as follows:

Step 1: Describe a theme identified in the meeting notes, providing a description of the theme and if it relates to a comparable BPMN element. If the theme is perceived be analogous to a person-to-person interaction type of activity that cannot be fully expressed as a BPMN element, the theme is developed into an interaction Drools rule using the following stages:

Step 2: Produce a thematic map to show the associated attributes of the theme that are identified in the meeting observation notes. For clarity, in the thematic map, themes are identified with a blue background, people with a yellow background, and attributes to the them with an orange background.

Step 3: Using Kaiser et. al's definitions, map the theme to **Events, Conditions** and **Actions**, expressing these mappings using FOL syntax.

Step 4: Using the FOL formula produced in Stage 3, produce a Drools rule by mapping the Events and Conditions to the *when* section of the rule that defines the logic of the rule. Actions are mapped to the *then* block of the rule, which define the action the rule should perform. As we are using these rules as part of a BPMN workflow model, the actual actions are defined in the sequence flow of the workflow model, that is in turn determined by the rule.

4.3 Background and purpose of MAPPA meeting observations

The purpose of observing MAPPA meetings was to elicitate requirements for the development of a shared decision making tool that would visualise the risk an offender would pose to the general public, staff the offender interacts with, a known person, children and themselves. These risks are the salient items to be considered by each MAPPA meeting, with the aim of developing strategies to manage the risk the offender poses for each group of people. As part of this elicitation process, we wished to devise a method of capturing the interactions between the meeting participants, such as when a person asked a question about the current topic being discussed. The probation services wished to investigate the feasibility of developing tools to capture the decisions made as well as the provenance of the information used to support those decisions, should it be necessary to audit the decisions made at a later date.

MAPPA is an arrangement of local authorities and relevant organisations who are tasked to manage the risk posed by sex offenders, violent offenders or any other type of offender who is deemed to be a risk to the public [129].The MAPPA process described in this Chapter is based upon the observations in the North East of England. The MAPPA guidance document [129] categorises the participating authorities and organisations as:

1. The responsible authority, which are the police, prison services and the probation services. These organisations have a duty to ensure that risk posed by an offender is managed appropriately. Depending on the case representatives from responsible authorities may not be based in the local area. For example, if an offender is in prison in another part of the country.

2. Duty to care agencies, such as social services, the offender's general practitioner, other medical professionals such as psychiatrists and the offender's local housing authority. These organisations are required to work with the responsible authority on particular aspects of the offender's life, such as education, healthcare and housing.

3. Strategic management board (SMB). This organisation is responsible for overseeing each MAPPA region in England and Wales. Each SMB monitors the performance and compliance of the MAPPA meetings and produces an annual report for its region.

4. Lay advisers. These are members of the general public who provide an independent perspective on the work undertaken by the other organisations.

The MAPPA guidance document defines three categories of offender who may be potentially eligible for the MAPPA process: [129]

"Category 1: Registered sexual offender as specified under Part 2 of the Sexual Offences Act 2003.

Category 2: Violent Offenders and Other Sexual Offenders:

(a) An offender convicted (or found not guilty by reason of insanity or to be unfit to stand

trial and to have done the act charged) of murder or an offence specified under Schedule 15 of the Criminal Justice Act 2003 (CJA 2003) who received a qualifying sentence or disposal for that offence or

(b) An offender subject to a Disqualification Order for an offence listed under Schedule 4

of the Criminal Justice and Court Services Act 2000).

Category 3: Other dangerous offenders: a person who has been cautioned, reprimanded, warned or convicted of an offence which indicates that he or she is capable of causing serious harm and requires multi-agency management at level 2 or 3. The offence might not be one specified in Sch.15 of the CJA 2003."

A similar process for managing potential offenders is termed PDP or Potentially Dangerous Person. A PDP is defined as:

"...a person who has not been convicted of, or cautioned for, any offence placing them into one of the three MAPPA Categories but whose behaviour gives reasonable grounds for believing that there is a present likelihood of them committing an offence or offences that will cause serious harm." [130].

For each individual MAPPA case, there two distinct phases. Firstly there is the pre-screening meeting. These meetings are attended by the responsible authorities to access if an offender is suable for the MAPPA process. The second phase is one or more MAPPA case meetings, should an offender be deemed appropriate for MAPPA. These meetings take place between the responsible authorities and duty of care agencies. These meetings are repeated until the offender is taken off the MAPPA process.

An important consideration for all MAPPA meetings is the confidentiality of the offender, victims and any other person deemed to be at risk should any information should be released to the public. This may include a person’s name, address or the offence committed. Therefore, the documents produced and information used falls under the provision of the Data Protection Act which outlines the legal requirements for protecting identifiable information [129], [131].

A total of 8 meeting observations were undertaken. The number of observations undertaken were based on the time constraints of the MAPPA meeting participants, as well the length of time given for the overall project. The meetings are summarised as follows:

Pre-screening meeting: 4 individual cases were assessed, 3 MAPPA and 1 PDP. The PDP case was dropped from the caseload.

Case 1 – Female: this case was dropped from the MAPPA caseload after the 1st case meeting. This case was first observed in the pre-screening meeting.

Case 2 – Young offender: two MAPPA case meetings were observed for this case. This case was first observed in the pre-screening meeting.

Case 3 – Male: this case was dropped from the MAPPA caseload after the 1st case meeting. This case was first observed in the pre-screening meeting.

Case 4 – Male: one MAPPA case meeting was observed.

Case 5 – Male: one MAPPA case meeting was observed.

Case 6 – Male: one PDP case meeting was observed.

The cases are summarised in the following table:

Meeting	Date	Comment
MAPPA pre-screening	20/05/2013	Cases 1-3 first discussed in this meeting. A PDP (Potentially Dangerous Person) case was discussed but dropped from caseload.
Case 3 (Male)	11/06/2013	1st Meeting, dropped from MAPPA caseload
Case 2 (Young Offender)	13/06/2013	1st Meeting
Case 1 (Female)	12/06/2013	1st Meeting, dropped from MAPPA caseload
Case 4 (Male)	16/07/2013	1st Meeting
Case 5 (Male)	16/07/2013	1st Meeting
Case 6 (Male)	16/07/2013	PDP, First Meeting
Case 2	22/08/2013	Case 2 follow up

Table 6: List of observed and documented MAPPA meetings

Using the activities described in the meeting notes, each part of a typical MAPPA case is described in detail, with a BPMN diagram being constructed to show how each part of a MAPPA case can be expressed as a workflow model.

4.4 MAPPA screening

Regular meetings are undertaken to screen MAPPA referrals. The screening meetings consist of members of the Northumbria Probation Trust and Police officers of Northumbria Police accessing MAPPA and PDP referrals. Other bodies submit MAPPA referral forms (these are termed MAPPA A documents which contain the offender's details and details of the crime(s) committed) that are subsequently collated together. The meeting members will review each referral in turn, using the probation services computer as well as the Police National Computer to collect background information on the offender that is relevant to their assessment (i.e. previous offending, family connections). Once all members had collected a sufficient amount of information, they discussed whether the offender's referral should be accepted and seen by the MAPPA panel. There decision was recorded in a "Feedback to referrer" document along with their rationale for accepting or declining adding the offender onto list of MAPPA cases that are dealt with in the area, also termed the MAPPA caseload. This process is visualised in the BPMN diagram presented in Figure 46.

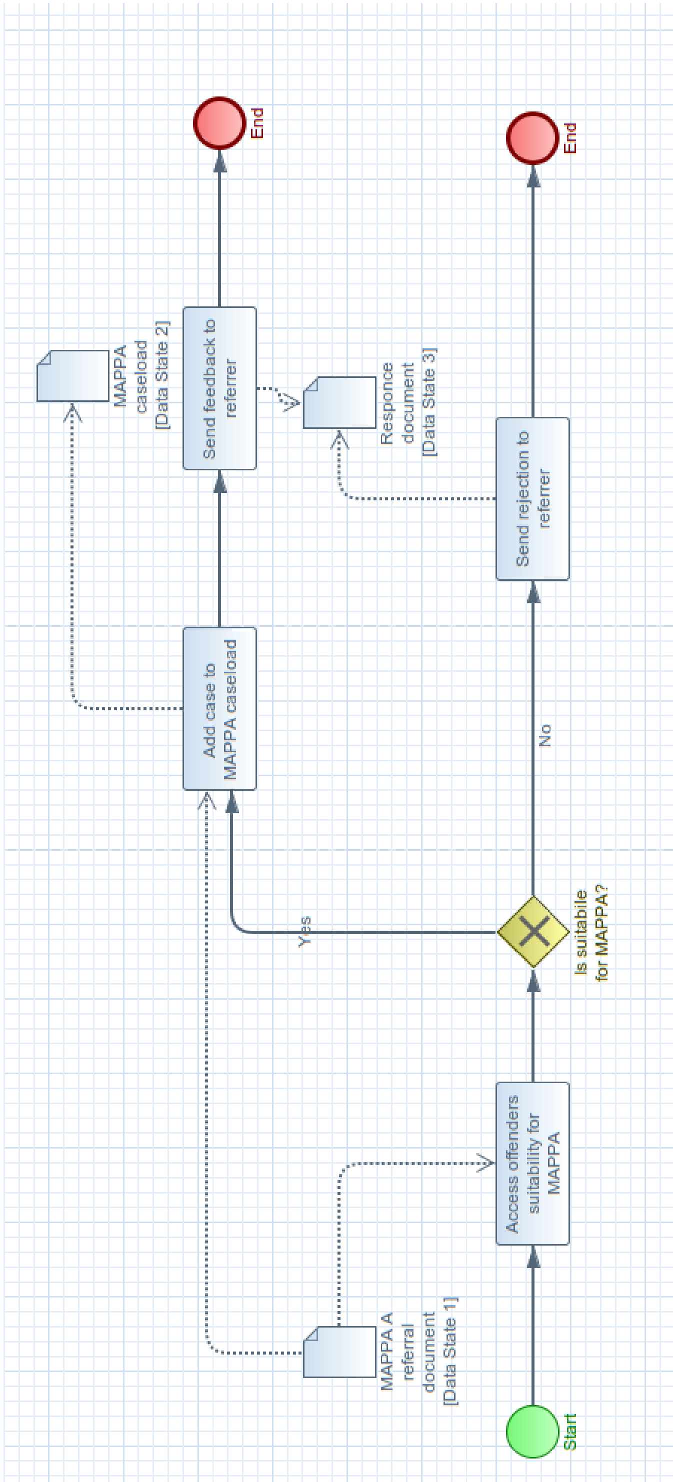

Figure 46: BPMN workflow of MAPPA screening meeting

4.4.1 MAPPA case meetings

If the screening panel adds a case to the MAPPA caseload, the MAPPA administrator uses the listed contacts on the MAPPA A referral document to invite referral agencies to each MAPPA meeting (e.g. Local council, mental health works etc.). Each invitee produces a MAPPA report detailing if the offender is known to the agency and if so, details of how they are known. The report also indicates if the person completing the report will be attending the meeting, or if a colleague will attend in their absence. These reports are faxed or sent via secure email to the MAPPA administrator prior to each meeting. This process is summarised with a BPMN diagram in Figure 47.

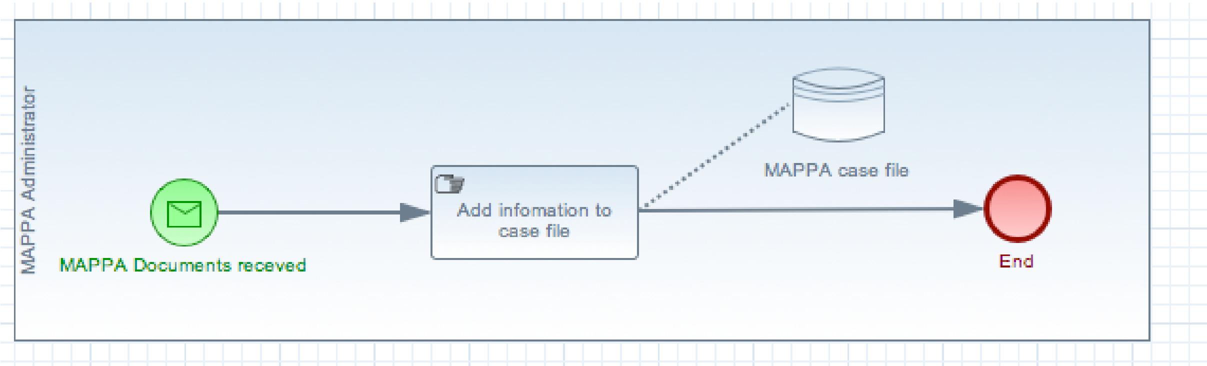

Figure 47: BPMN workflow diagram showing the collection of documents for a future MAPPA case meeting.

Each MAPPA case meeting is attended by a chair. MAPPA case administrator (who takes notes for the minutes) along with a representative from the various agencies. The chair's role is to direct the meeting and to summarise what is being discussed at each stage.

Participants will bring their own MAPPA documents they have sent to each meeting, other MAPPA documents such as referrals, minutes (if applicable) and other documents on the offender are provided by the chair to each member. The chair as well as meeting attendees also frequently make their own notes as the meeting progresses.

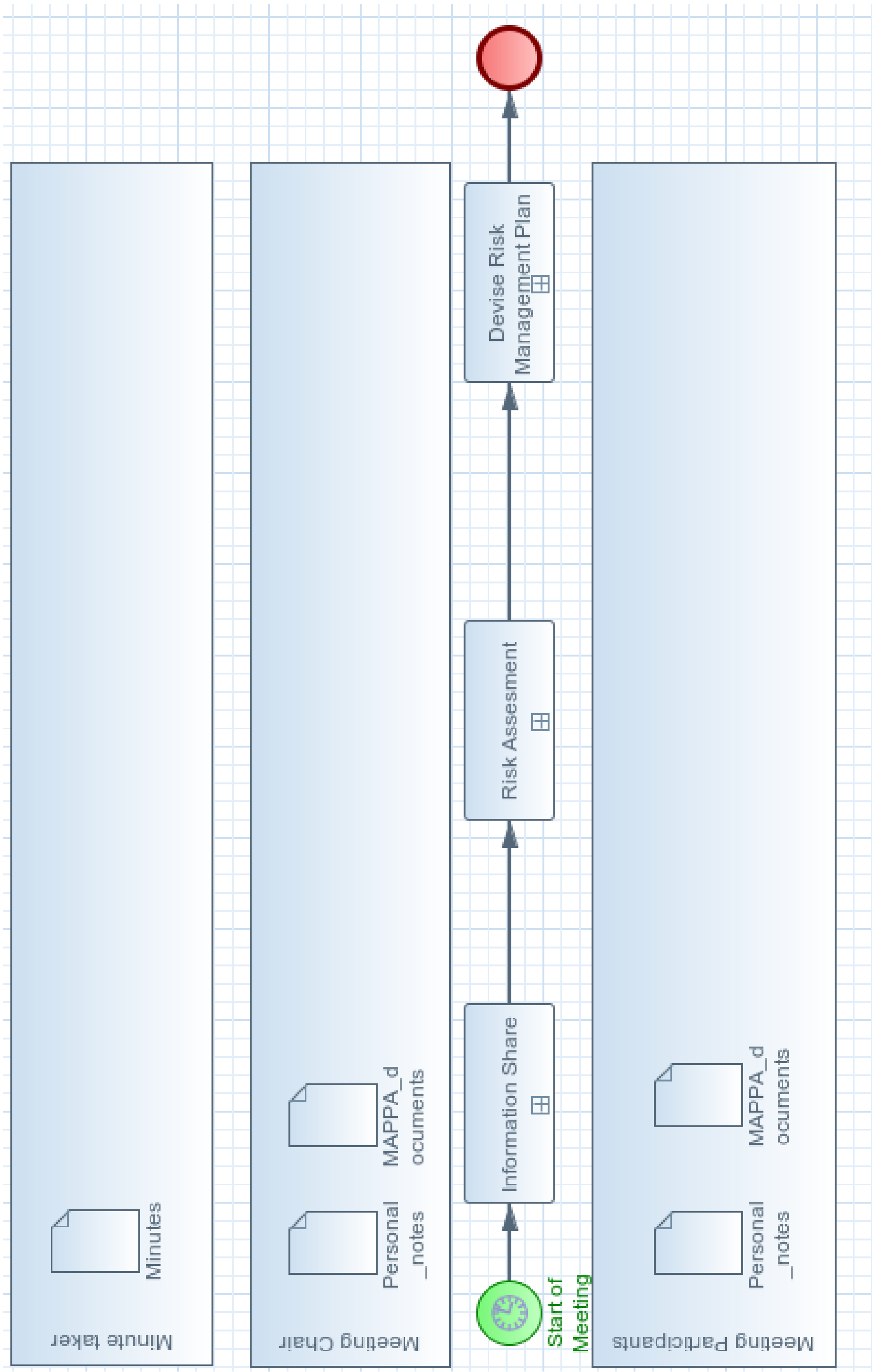

Figure 48: BPMN workflow diagram of the high-level activities that occur in a MAPPA case meeting. For clarity, the data associations between each document and sub-process have been omitted.

The chair will open the MAPPA meeting by summarising who is being discussed in the meeting and informing the panel the confidentially requirements. This is followed by three high level processes: information share, risk assessment and devising the risk management plan. This process is summarised in Figure 48 where each individual task is represented as containing a BPMN sub process encapsulating the individual tasks involved. For clarity, Figure 48 does not show the associations between data items and sub-processes. Instead, it is assumed that each data item can be accessed by each individual sub-process.

During the information sharing process, each panel member will share with the group their experiences/contact with the offender from the perspective of the organisation they represent. For those who do not attend, the chair will share the information that was provided on their MAPPA report.

For cases that have had a previous MAPPA meeting, the chair will go through the minutes and ask for updates regarding the agreed risk management plan. In this process, each meeting participant will disclose information they have about the offender. The minute taker will record all information that is said during this phase. This process is visualised in Figure 49.

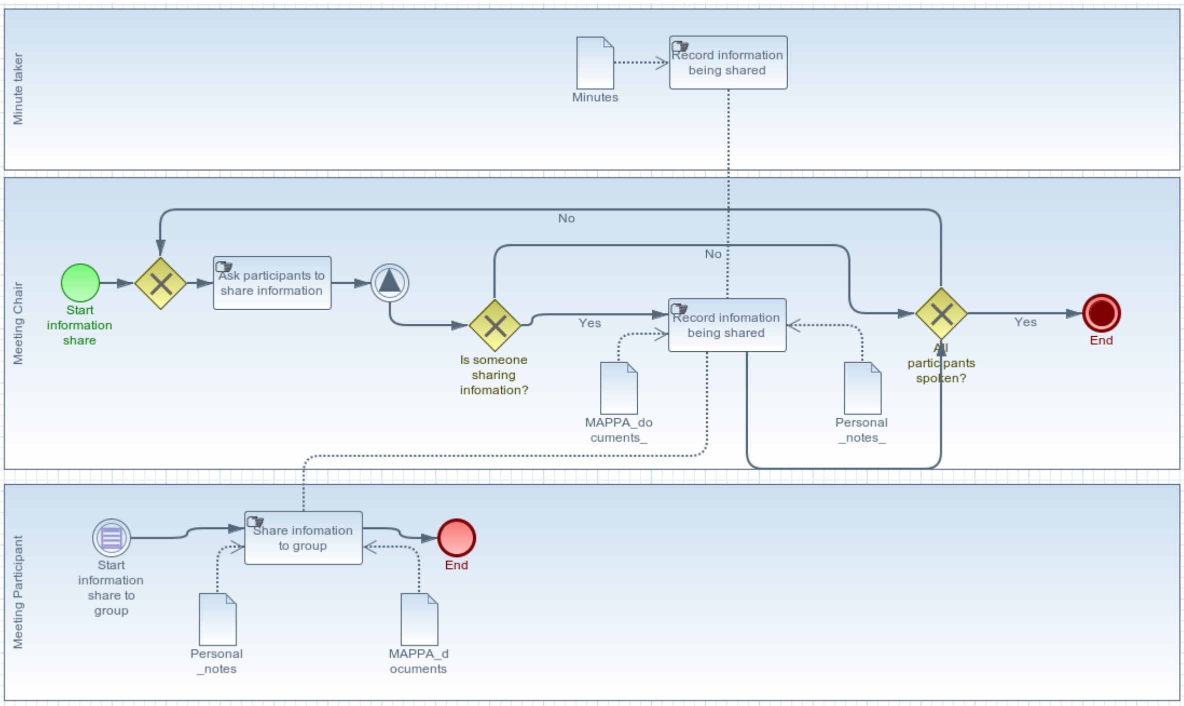

Figure 49: BPMN workflow diagram of the Information share process

Following from the information sharing phase, the meeting chair will ask participants to devise an outline risk assessment. This will involve the chair asking participants to infer the risk the offender poses based upon the information disclosed in the “Information share” phase. Based of the information that was disclosed during the “information sharing phase” the panel deliberates on the level of risk the offender poses to:

- Public
- Prisoners
- Known adult
- Children
- Staff
- Self

For each category, the offender will be ranked Low, Medium, High or Very High risk. The ranking is done empirically, based on the information shared during the previous phase and are accessed based on the likelihood of the offender causing harm and the likely impact the risk poses. This process is expressed using a BPMN diagram in Figure 50. The “Infer risk” and “Agree on risk” activities are based around six categories above, with the amount of information being used to base the decisions regarding the level of risk varying depending on the type of offender involved and their history of offences and incidents.

Once the risk assessment has been achieved, the chair will ask the panel members to consider and devise a risk management plan in order to manage the risk factors that have been identified. Panel members identify specific actions their organisation can do to manage a specific risk (i.e. arrange home visits). The Chair will ask the member who will do the action and how long the action will take. This is recorded by the minute taker. The development of the risk management plan is expressed as a BPMN diagram in Figure 51. As part of this process, meeting participants will use their notes to assist them with their tasks. At the same time, the minute taker will record the proceedings of the meeting, which is distributed to the members after the meeting has finished.

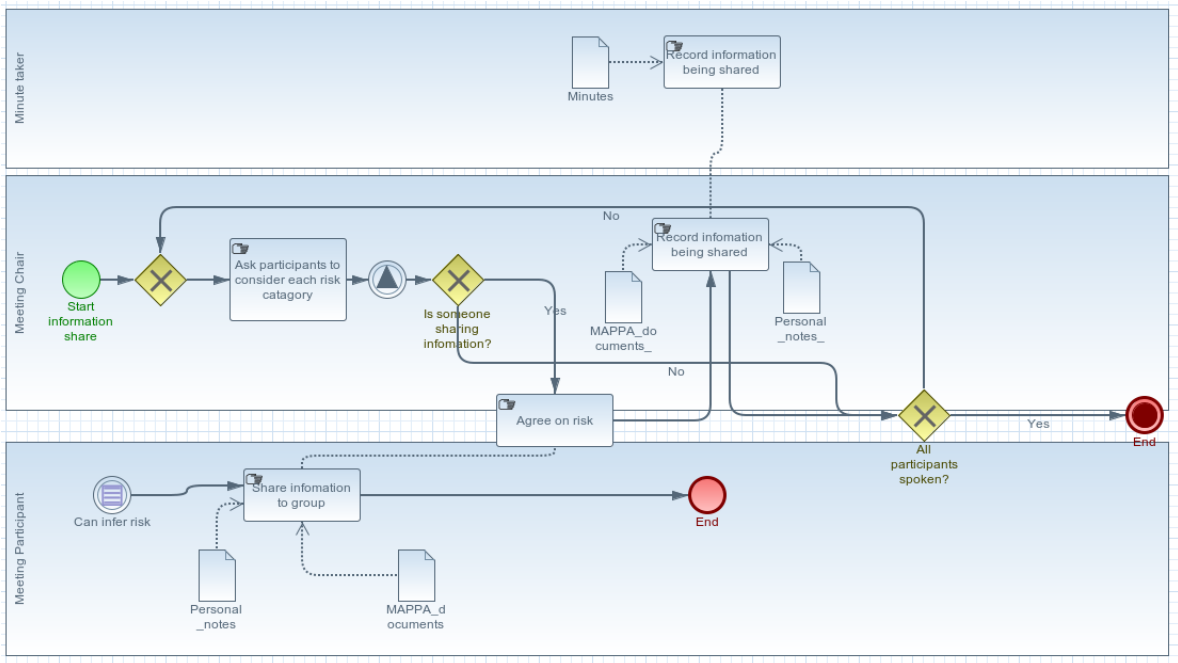

Figure 50: BPMN diagram showing the steps taken to form a risk assessment.

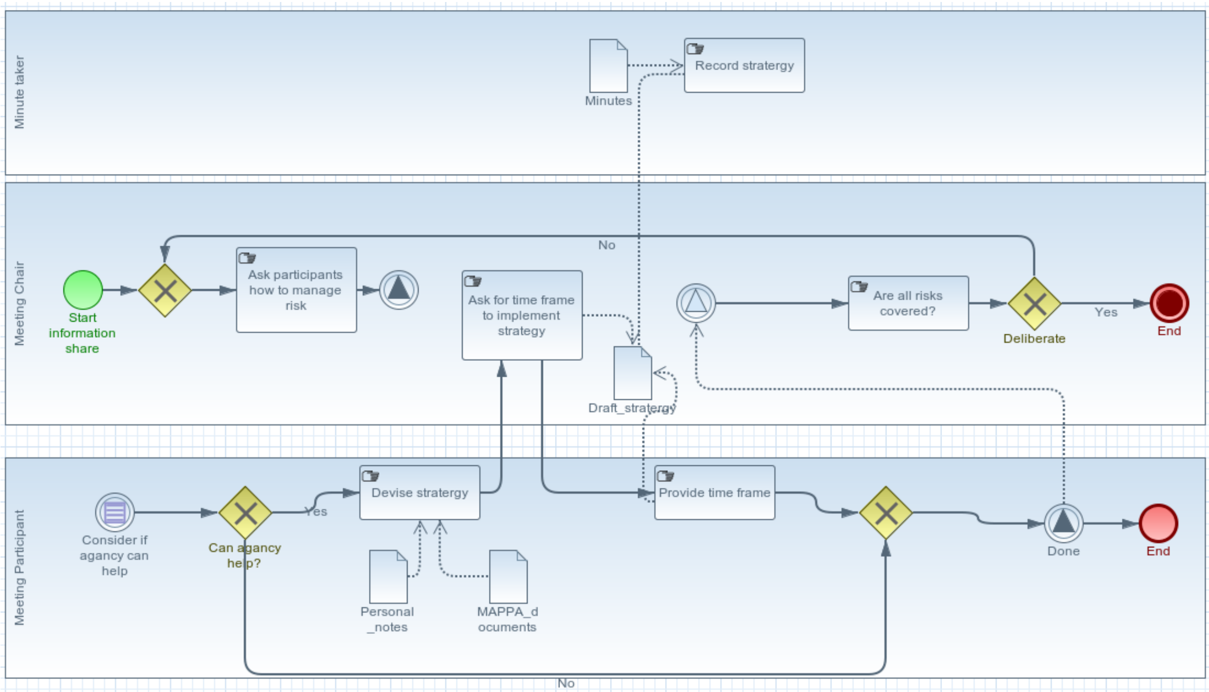

Figure 51: BPMN diagram showing the formulation of a risk management plan

Once a sufficient number of risk management strategies have been identified to manage the risk the offender poses, the meeting closes with the chair asking if there are any data protection and human rights issues and asks who will update the ViSOR computer record to indicate that the meeting has taken place.

In this chapter, we describe the development of Drools rules to represent rules that symbolise person-to-person interaction in a decision making scenario. We decided to develop rules for this type of interaction as the jBPM engine does not support the BPMN 2.0 collaboration type of interaction [31], [132], and as the scoped data object defined in Chapter 3 for BPMNdm does not visualise associations between documents and tasks, some types of interactions modelled will be defined without a visual representation.

4.4.2 Summary of the MAPPA meetings

The salient purpose of the MAPPA meetings appears to be to access the risk the offender poses to the public, how best to manage the risk identified and whether there is an agenda to continue monitoring the offender under the MAPPA framework.

- The MAPPA meetings follow a consistent structure, with each portion of the meeting beginning with an "information sharing phase" where meeting participants are invited to inform the panel of their experiences/contacts with the offender. This is followed by a "deliberation phase" where the risk the offender poses is deliberated followed by an "action phase" where panel members nominate themselves or their agency to undertake specific actions in order to mange the risk.

- While MAPPA reports can be submitted electronically before each meeting, all meetings have used paper printouts and/or handwritten notes exclusively. Computerised records are updated after each MAPPA meeting by the MAPPA administrator where necessary.

Each meeting observed followed the guidelines provided in the *MAPPA Guidance 2012* document [129], with each case having a pre-screening meeting in order to determine if each case merited being part of the MAPPA process and if so, would be followed up with MAPPA case meetings, which would be held periodically until the case was dropped from the MAPPA caseload. The following sections detail firstly the MAPPA pre-screening meetings, follow by the MAPPA case meetings.

The MAPPA meetings pose a number of challenges for the adoption of decision-support technologies. As the meetings can be held back-to-back and with participants who may have travelled a considerable distance, it is essential that any tool developed does not require a significant amount of time to set up, use or shut down at the end of each meeting.

The MAPPA guidance document identifies confidentiality as being vital for safety and well-being of the MAPPA members, the general public and the offender. If a decision support tool were to contain identifiable information, or interact with other systems that contain confidential information then the tool would require stringent safeguards to minimise the risk of unauthorised access to this information.

The risk assessment phase is one area that could potentially benefit from the support of a computerised risk communication tool. The risk assessment phase is dependent on the information disclosed during the information sharing phase, and in turn is a dependency on devising the risk management plan. Therefore, the risk assessment can be considered a method that filters and categorises information on the offender into categorised data that can be used to infer the risks the offender poses.

In terms of the activities undertaken, the risk assessment requires each member of the meeting to deliberate with the rest of the group, using the information that was shared in the previous phase of the meeting to access the risk the offender poses.

The MAPPA risk assessment forms an important part of the overall MAPPA process as it directly influences the decision as to what level the offender should be managed at in the MAPPA framework as well as the risk management plan.

In the meeting observations, it was observed that the risk factors were inferred from all available information sources, such as the information provided by the agency representatives as well as written documentation such as offender records where applicable. In two meetings, it was noted that information on risks were provided as a printed risk matrix. The final decision, as to which category the offender best fitted in, was observed to be achieved empirically based on the inferred information.

When deliberating the risks as defined by the MAPPA guidance [129], the principle means for assessing and deciding on the risk level was achieved using verbal discussions using the information disclosed during the information sharing phase as a basis on accessing the risk levels. The finalised risk levels of the risk assessment where then recorded by the minute taker and the scores were further used as a basis for devising the risk management plan.

Given the risk assessment and risk management phases require all participants to collaborate in order to share relevant information each person has on the offender, these phases are of particular interest when consider the design of a tool to support the meetings, due to the deterministic types of social interactions, such as interruptions and requests for clarifications. This issues are discussed in greater detail in the next section. The development of the risk communication tool intended to support the MAPPA meetings was not developed due to funding constraints with the probation services at the time of development.

4.5 Summarising salient activities using thematic content analysis

Using the thematic content analysis methodology discussed in Section 4.2.2, five themes were identified in the meeting observation notes. As we aim to describe the semantics of person-to-person interactions using Drools rules and the BPMN formalism [41], we map each theme to the existing BPMN formalism, and where necessary, describe the development of each theme into a Drools rule using the methodology described in Section 4.2.3. For each theme that has a Drools rule representation, we have provided a thematic map to show how each theme relates to the information contained in the meeting observation notes.

Theme 1: Information assets

Step 1:

Information assets are artefacts that each meeting participant has in the MAPPA meetings. These assets are typically composed of documents based off the MAPPA template documents provided in the MAPPA Guidance Document, customised to indicate who is filling out the report and the relevant information the organisation the person represents has on the offender.

Other types of documents included personal notes either brought or written down by each participant and in one meeting, a paper based information grid provided by the meeting chair.

Information artefacts are directly analogous to the BPMN document element.

Theme 2: Activities

Step 1:

Activities are the task performed by each meeting participant during the course of the meeting. The activities identified were:

- Note taking
- Information sharing, where each participant shares any relevant information that the representative organisation has on the offender with the rest of the group.
- Planning; either to perform a specific activity after a meeting, or as an agenda item for the next meeting.
- Interventions; these are done by the meeting chair, in order to move the meeting forward, either to time constraints, or due to the meeting members being unfocussed.
- Interjections; these are done by any meeting member who is not the current person speaking. Interjections are additional pieces of information or statements that complement the subject being discussed by the original speaker.

We consider activities to be analogous to BPMN task elements. However, Interjections and Interventions are not defined as standard tasks as they are not deterministic, i.e. they may occur randomly and are dependant on the information being discussed and the level of knowledge of the participants on the information etc. Unless the modeller was constructing a workflow model that precisely mapped every activity and interaction that was undertaken in sequence, it is difficult to map interjections and interventions to tasks that would be executed as part of the default sequence flow. Therefore, by mapping these two themes to Throw and Catch events, they allow the modeller to specify the activities needed to handle interjections and interventions at design time, and specify how these events are triggered at run time as necessary. Using events also helps to simply the workflow model, as they encapsulate repetitive tasks (such as a person asking multiple questions at different times), without needing to model the specific activities performed when a question is asked every time during the duration of scenario being modelled.

Intervention theme

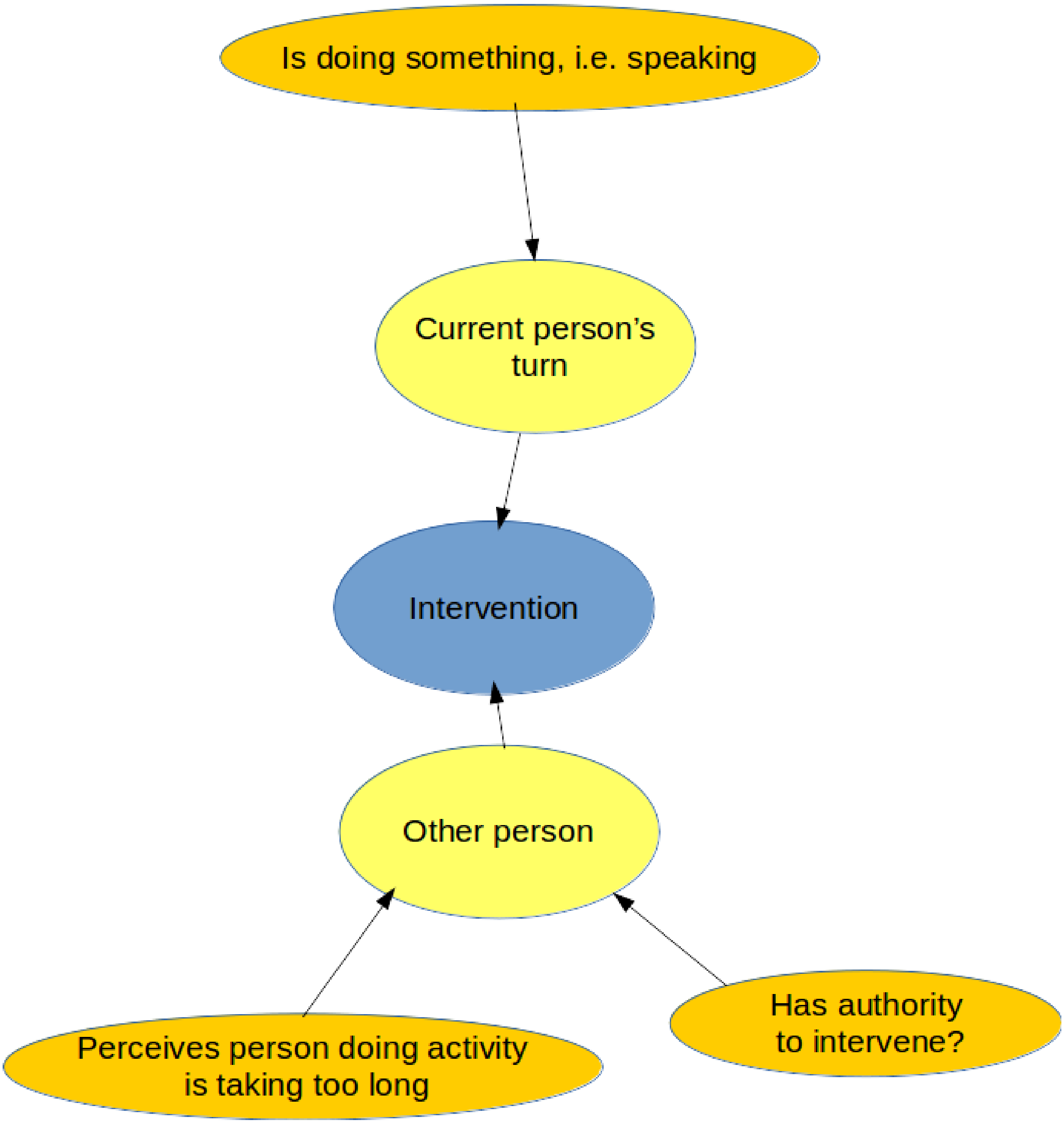

Figure 52: Thematic map of an Intervention theme

Step 2: An intervention occurs when a meeting participant deems it necessary to stop the current person talking in order to perform a specific action. For example, in a MAPPA meeting, if a person was perceived too take to long to discuss the current item (this is especially important if the meeting is running late), the meeting chair would intervene and ask the person to stop talking or to summarise the rest of their discussion. This theme is now expressed using FOL syntax.

Step 3:

$$isTurn(P_1) \land (isChair(P_2) \land \neg isPerson(P_1) \land TimeTaken(P_1) > TimeAllowed) \rightarrow Intervention(P_1, P_2) \quad (3)$$

The FOL-like syntax for intervention theme for a meeting participant taking too long to perform an activity is provided in Equation (3). Other types of interventions could be considered a variant of the Interjection theme described in Equation (3) where a person may wish to interrupt the current speaker to provide additional information. For this theme, it is Person P_1's turn. The intervention is triggered when P_2 is the meeting chairman, is not person P_1 and the time taken for P_1 to perform their current activity has exceeded a set value.

Step 4:

```
package uk.ac.ncl.cs

import org.drools.runtime.StatefulKnowledgeSession;
import uk.ac.ncl.cs.Actor

rule "Intervention"
dialect "java"
ruleflow-group "QIntervention"
    when

     a1: Actor( Role == "Chair", !Turn )
     a2: Actor( Role != "Chair", Turn, a2.TimeTaken > 5)
    then
        //actions
        a1.setTurn();
        System.err.println("The participant has spent too long talking");
        kcontext.getKnowledgeRuntime().setGlobal("chairIntervention",
true);

end
```

Figure 53: Drools rule for Intervention theme

The Drools rule for the Intervention in Figure 53 The conditions are that when an `Actor` class instance (`a1`) is the Chair and it is not their turn, and if a second actor who is not chair, and it is their turn, and they have taken over 5 minutes to perform their current activity, the

Chair `a1` takes over by calling the `setTurn()` method that yields control from `a2`. The global variable "`chairIntervention`" is set to to `true`.

Interjection theme

Step 2:

Based on the meeting observation notes, a specific pattern was identified where a given subject was being spoken by a person in the meeting. A second person had additional information on the subject matter. During a pause, the second person speaks to provide this additional information to the rest of the group. This type of theme was classified as an interjection, where additional information is provided about the subject matter when there is a natural pause in the meeting (i.e. when the current person talking stops). The thematic map showing how this theme is constructed is provided in Figure 54.

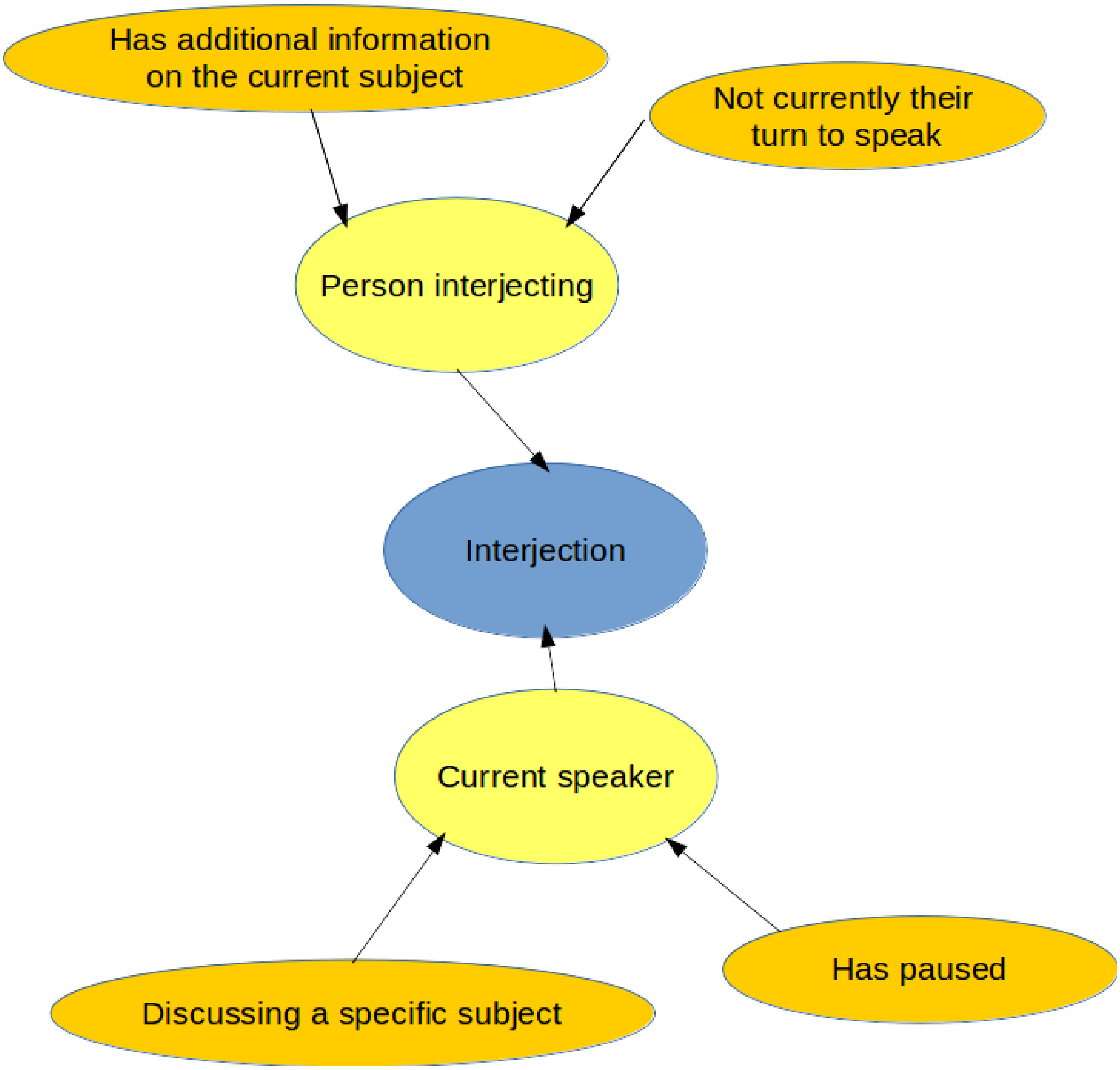

Figure 54: Thematic map of interjection theme

Using the definitions for data by Kaiser et. al., the conditions for an interjection are if it is a persons turn, and they have shared information about a given subject and have paused, there is a second person, it is not their turn to talk and they have additional information on the subject. These conditions form the Interjection event. The FOL syntax for this theme is given in Equation (4).

Step 3:

$$\begin{gathered} Subject(S_1) \land isTurn(P_1) \land \neg Talk(P_1) \\ \land (AdditionalInformation(S_1, P_2) \land \neg isPerson(P_1)) \rightarrow \\ Interjection(P_1, P_2, S_1) \end{gathered} \tag{4}$$

Equation 4 shows that an Interjection between persons P_1 and P_2 regarding subject S_1 occurs when S_1 is being discussed, it is P_1's turn to talk, but P_1 is not talking and P_2 has additional information on S_1 and it they are not P_1. The equation is then translated into a Drools rule as provided in Figure 55.

Step 4:

```
package uk.ac.ncl.cs

import uk.ac.ncl.cs.Subject
import uk.ac.ncl.cs.Actor


rule "Interjection"
dialect "java"
ruleflow-group "SubjectInterjection"

    when
        //conditions
        sub : Subject() //Subject in question
        a1 : Actor( Turn, Paused, Activity == "Talking" ) //Speaker
        a2 : Actor( a2.id != a1.id, !Turn, hasAdditionalInfomation(sub) )
//Other person

    then
        //actions
        System.out.println(a2.getId() + " has additional information, will
interject.");
        kcontext.getKnowledgeRuntime().setGlobal("willInterject", true);
end
```

Figure 55: Drools rule for Interjection theme

The syntax of the Drools rule in Figure 55 specifies that when the model instance has a given `Subject` (`Sub`) that is being discussed, an `Actor` (`a1`) who has the current turn, has paused and their current activity is "Talking", and a second `Actor` (`a2`) who is not `a1` and it is not currently their turn, has additional information on the subject being discussed. If these conditions evaluate to true, this will set the global variable "`willInterject`" to `true`, signifying that `a2` will interject.

The Drools rule does not define any actions on how an Interjection would be handled. The ultimate action is dependent on the modeller providing the necessary BPMN sequence flow of tasks as part of the workflow model.

Theme 3: Information snippets

Step 1:

Information snippets are pieces of factual information that are verbally given by one meeting member to the rest of the group. These can range from given information about an offender, such as the offender's contact with probation services, to simple statements, such as "No intelligence" when a meeting member's organisation does not have any intelligence reports on the offender.

Information snippets are analogous to the BPMN document element if they are recorded, otherwise they are analogous to a message flow between tasks if they are communicated verbally between each meeting participant.

Theme 4: Statements

Step 1:

In contrast to Theme 3, statements are based on a meeting member inferring information from a data source. For example, once a meeting member speaks, the chair may summarise what the person was saying i.e. "*Appears to have returned to [the] drug culture and violent behaviour*". These types of statements are further used when forming the risk management plan for the offender, or when deciding if the offender should be dropped from the MAPPA case load.

Information snippets are analogous to the BPMN document element if they are recorded, otherwise they are analogous to a message flow between tasks if they are communicated verbally between each meeting participant.

Theme 5: Questions

Step 1:

Questions are a specific form of an activity. Questions are typically asked by the meeting chair, particularly when reaching a section of the meeting agenda. The chair, as well as other meeting members also ask questions to the current person speaking when necessary, such as to ask for a clarification, or to ask for information that is relevant either to the meeting, or the questioner's organisation.

Stage 2:

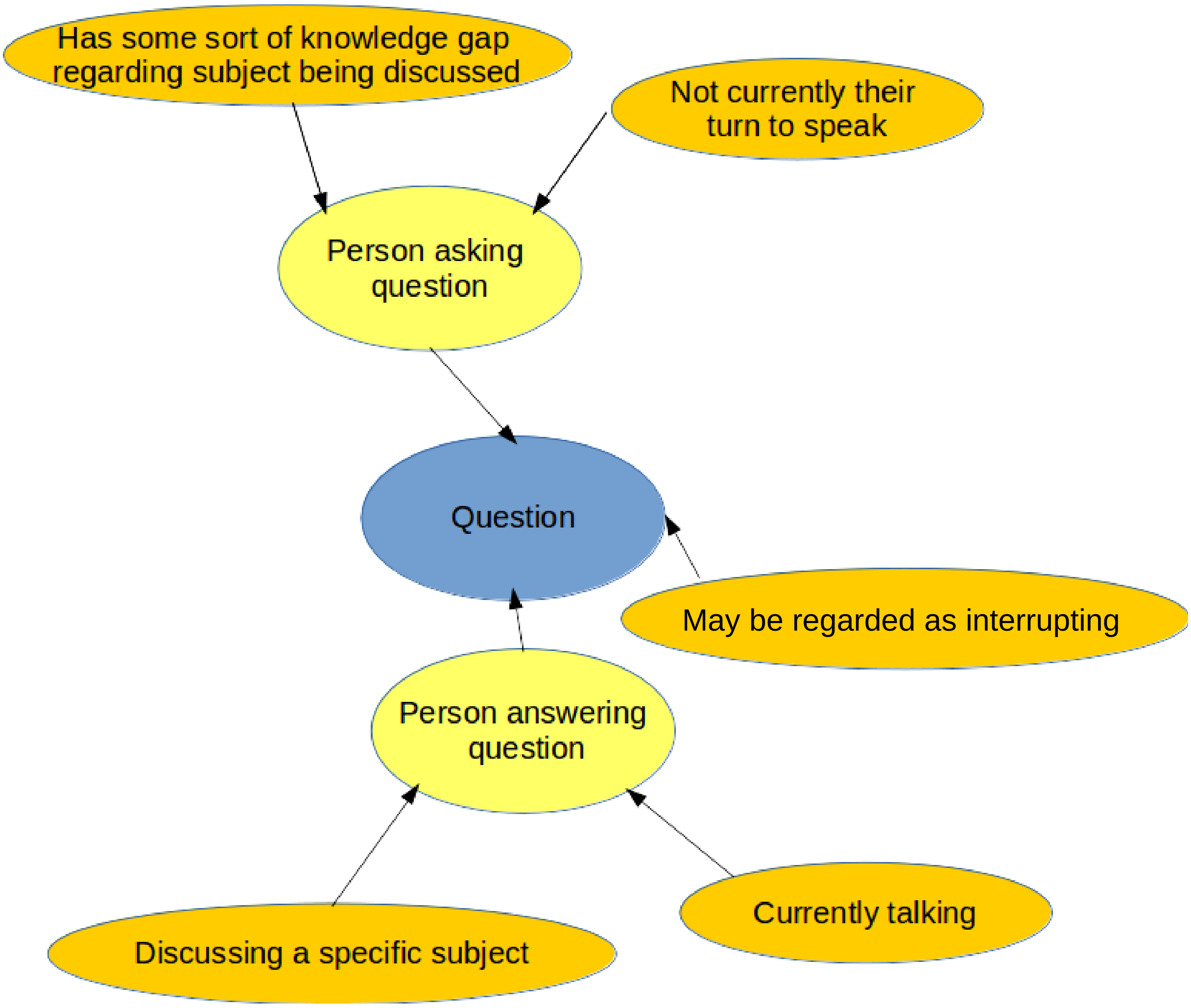

Figure 56: Thematic map of a question theme, showing the elements that constitute the theme.

Questions can be expressed in BPMN using the throw and catch intermediate events, as shown in Figure 43. While the BPMN formalism can be used to specify how a question is handled, the modeller is required to provide the necessary logic to trigger a question event. The principle of event based triggering of specific sequence flows is also useful if the modeller wishes to similar questions being asked at intermediate times during the execution of the workflow model. The thematic map in Figure 56 shows the composition of the question theme based on the analysis and categorisation of the data. For the question theme, we consider it to be composed of a person who is currently talking, and is discussing a specific subject matter. There is a second person who has some sort of gap in their understanding or knowledge of the subject being discussed, which will invoke a question event. This event may be considered an interruption, especially if it is not currently their turn to talk. Considering the classifications by Kaiser et. al., The conditions for this theme are if it is a person's turn to talk, the current subject being discussed and if a person understands the current subject. Depending on the scenario, it may be inappropriate for the person to ask the question at the current time (i.e. in court), then this will prevent the person from asking a question. The question theme can be considered the event that encapsulates these conditions. The FOL syntax for this theme is given in Equation 5.

Step 3:

$$Subject(S_1) \wedge isTurn(P_1) \wedge \neg isTurn(P_2) \wedge canTalk(P_2) \wedge \neg understandsSubject(S_1, \rightarrow Question(P_2, P_1, S_1) \quad (5)$$

The FOL syntax states that for a given Subject S_1, if it is P_1's turn and not P_2's turn, at the current time, P_2 cold talk, and P_2 does not understand Subject S_1, this will trigger a question event. The corresponding Drools rule for the question theme is provided in Figure 57. We assume that a subject item is linked to a data item contained in the BPMN workflow model, and a Person is represented by a BPMN swimlane containing a set of tasks and gateways associated with work each meeting participant does in a MAPPA meeting.

Step 4:

```
package uk.ac.ncl.cs

import uk.ac.ncl.cs.Subject
import uk.ac.ncl.cs.Actor


rule "Question"
dialect "java"

    when
        //conditions
        sub : Subject() //Subject in question
        a1 : Actor( Turn ) //Speaker
        a2 : Actor( id != a1.id, !hasSubjectById(sub.id), !Turn, canTalk )
//Other person

    then
        //actions
        System.out.println(a2.getId() + " does not have information,
firing question task.");
        a2.setToAskQuestion(true);
end
```

Figure 57: Drools rule for Question theme

The syntax of the Drools rule in Figure 57 specifies that when the model instance has a given `Subject` (`Sub`) that is being discussed, an `Actor` (`a1`) who has the current turn, and a second `Actor` (`a2`) who is not a1, does not have information on what is being discussed, it is not their turn and finally, when the rule is triggered, at the moment in time the model specifies a person can currently talk. If these conditions evaluate to true, this will set the variable `toAskAQuestion` to `true` for `a2`, which will be evaluated at a later point in the workflow model. This delegates the Action necessary to handle this event to the BPMN model rather than dealing with the event with a further rule.

4.5.1 Summary of themes and person-to-person interaction rules developed

The thematic content analysis discussed in Section 4.5 identified 5 themes that are relevant for the goal of producing models that encapsulate the decision making process: Information assets, Activities, Statements and Questions. The themes identified are mapped to the existing BPMN formalism and summarised in Table 7.

Theme	Comparable BPMN construct [41]
1. Information assets	*Data object*
2. Activity (except Interjections and Interventions)	*Task Element*
3. Interjection and Intervention activities	*Throw and Catch events, but require further definition on how they are generated.*
4. Statements	*Incorporated into Tasks or form a Data Object*
5. Questions	*Throw and Catch events, but require further definition on how they are generated.*

Table 7: Summary of thematic content analysis themes mapped to BPMN constructs

Information assets are analogous to BPMN Data Objects, which specify what information is needed for an activity be performed and/or what an activity produces. Activities (except for interjections and Interventions) are analogous to the BPMN task element, as every activity represents a discrete piece of work undertaken and does not provide further aggregation.

Interjections and Interventions are not defined as tasks due to their indeterministic nature. Unless the modeller was constructing a workflow model that precisely mapped every activity and interaction that was undertaken in sequence, it is difficult to map interjections and interventions to tasks that would be executed as part of the default control flow. Therefore, by mapping these two themes to Throw and Catch events, they allow the modeller to specify the activities needed to handle interjections and interventions at design time, and specify how these events are triggered at run time as necessary. Using events also helps to simplfy the workflow model, as they encapsulate repetitive tasks (such as a person asking multiple questions at different times), without needing to model the specific activities performed when a question is asked every time during the duration of scenario being modelled.

Statements can use Data Objects in a similar way to Information assets, especially if the statements are complex or long. Otherwise, statements can be assumed to form part of the activity being undertaken. Questions are similar to Interjections and Interventions in that they are indeterministic, especially if the modeller wants to generalise the scenario being modelled. Therefore, questions can be represented using Throw and Catch events.

Based on these assumptions, Interjections, Interventions and Questions were chosen to be defined as Drools rules, as they cannot be mapped to a single BPMN element, and by their nature, how they are generated and dealt with in a workflow model requires a formal definition. We used a four stage process, described in Section 4.2.4 to produce a Drools rule from the theme identified as part of the thematic content analysis phase.

4.6 Discussion

This chapter presented three Drools rules representing three person-to-person interaction themes identified during the MAPPA meeting process. These themes were interjections, interventions and questions. As visualised in Figure 43, indeterministic activities, such as interventions, interjections and questions can be visualised in BPMN using throw and catch events, which are separate from the sequence flow. However, it is necessary for the modeller to provide the appropriate code to trigger these events, which can be achieved by providing a rule that provides the logical code to initiate the interaction. The rules have been developed in such a way that they are specific for the MAPPA project, instead are intended to represent an aggregation of the salient person-to-person activities that were observed. This was intentional as the resulting rules and workflow models are abstract enough to be used for constructing models of other decision making scenarios.

The steps necessary for transforming data produced by ethnographic methods into Drools rules is summarised as follows:

Pre step: Perform thematic content analysis on the data to identify relevant themes.

Step 1: For each them, map the theme to an analogous BPMN element.

Step 2: For each them that does not map well to a BPMN element, show the thematic map of the theme identifying the key attributes that constitute the theme.

Step 3: Using Kaiser et. al's definitions, map the theme to **Events, Conditions** and **Actions**, expressing these mappings using FOL syntax.

Step 4: Using the FOL formula produced in Step 3, produce a Drools rule by mapping the Events and Conditions to the *when* section of the rule that defines the logic of the rule. Actions are mapped to the *then* block of the rule, which define the action the rule should perform. As we are using these rules as part of a BPMN workflow model, the actual actions are defined in the sequence flow of the workflow model, that is determined by the rule.

It has been intended to keep the rules and workflow patterns as basic as possible, as building more complex rule sets and associated workflow patterns would risk specialising the models to the MAPPA scenario. In order to build more complex person-to person interactions, multiple instances of the rules and models can be used, by making the models a sub-process for a larger process. This is demonstrated with the use of re-usable sub processes in Chapter 5. This allows the models and rules to be re-used when necessary. In cases where the modeller wishes to simulate person-to-person interactions between more than two parties, this can be achieved by the modeller implementing this functionality as a Java program that is run as part of the jBPM workflow engine. For example, in the larger process, a script task would iterate over a list of actors associated with the interaction, and re-run the desired rule until all of the actor have participated in the sub-process. A second advantage of this approach is that it can be assumed that for a given interaction between a Chairperson and other meeting participants, multiple participants can be associated with a single swimlane, and an assumption can be made that these Actors will perform the same tasks.

A limitation of the three rules and associated workflow patterns is that both do not encapsulate simultaneous many-to-many or 1-to-many interactions, such as if two people interject immediately. This would require the modeller to modify the rules and workflow patterns to associate more actors with the Drools rule files and swimlanes when necessary.

Chapter 5 - Subjective data in workflow models

5.1 Introduction

The chapter describes a methodology of encapsulating subjective information in workflow models, particularly as a method of encapsulating a person's preferences and uncertainties when making a decision. In this thesis, Triangular Fuzzy Numbers (TFNs) was chosen as the method of representing subjectivity in workflow models. We provide an example of transforming ethnographic data using patient responses indicating their preference for shared decision making using the Control Preference Scale categories [133]. This data is expressed as a TFN and is used in a BPMN workflow model to show how TFNs can be used to specify the sequence flow in a workflow model.

The chapter continues by outlining how the usage of TFNs has been used by other researchers to describe encapsulating people undertaking decision making activities that involve assessing multiple criteria by using the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) algorithm. The chapter then describes how TFNs and TOPSIS can be incorporated into BPMN workflow models to determine the relevant sequence flow at a specific gateway.

Finally, the chapter will outline how BPMN can be used to provide the functionality for BDI (Belief-Desire-Intention) software agents. A modeller can use BDI type agents to simulate human appraising and coping activities [61], where the scenario being modelled requres the usage of such functionality.

5.2 Triangular Fuzzy Numbers

Triangular Fuzzy Numbers (TFNs) are proposed as a method of representing subjectivity and uncertainty when modelling decision-making workflows in BPMNdm. We chose this approach for representing subjective data, such as people's preferences, has already been used to simulate decision making choices [134]. TFNs are used in the domain of multi-criteria decision-making (MCDM). For example, where the data used to model choices can be subjective or uncertain, such as for risk analysis [135], building site selection [136] and selection of weapons systems [137]. In this chapter, ethnographic data regarding patient and doctor preferences for treatment obtained as part of the DASH (Development and Assessment of Services for Hyper-acute Stroke) Project [4] is used as an example of using TFNs to encapsulate subjective preferences within a workflow model.

Triangular fuzzy numbers are a representation of an interval of confidence, represented by a triplet [69]:

$$A=(a_1, a_2, a_3) \tag{6}$$

The first value of the triplet, a_1, represents the lowest value in the set, a_2 represents the geometric mean and a_3 the highest value. This is generalised in Equation (117).

$$\begin{aligned} a_1 &= min(x_1, \dots, x_n), \\ a_2 &= \left(\prod_{k=1}^{n} x_k\right)^{1/n}, \\ a_3 &= max(x_1, \dots, x_n). \end{aligned} \tag{7}$$

The geometric mean is used instead of the mean value as the geometric mean is seen as analogous to the cognitive process that human decision-makers undertake to form judgements when evaluating multiple criteria in a real-life decision making problems [139] and when satisfying consistency axioms between sets of judgements to ensure the set of judgements are consistent [140]. However in cases where pairwise comparisons were not used, for example if the data is composed of number of responses for a set of categories, the weighted average can be used instead [141]. This is shown in Equation (8), where each category weight is assigned to w_k:

$$\begin{aligned} a_1 &= min(x_1, \dots, x_n), \\ a_2 &= \sum_{k=1}^{n} w_k x_k, \\ a_3 &= max(x_1, \dots, x_n). \end{aligned} \tag{8}$$

As a TFN set is an aggregation of a data set, a membership function is used to indicate the *degree of membership* an individual value has for the set as shown in Equation (9).

$$\mu_a(x)=\begin{cases} 0, & x \le a_1 \\ (x-a_1)/(a_2-a_1), & a_1 \le x \le a_2 \\ (a_3-x)/(a_3-a_2), & a_2 \le x \le a_3 \\ 0, & x \ge a_3 \end{cases} \tag{9}$$

Given the hypothetical TFN:

$$A=(2,3.5,5) \tag{10}$$

The use of the membership function can be made using Equation (9) and the values 1,2,2.75,5 and 6, where 0 represents a number outside of the set and any positive number represents a degree of membership for the number:

Number	Degree of Membership
1	0
2	0
2.75	0.5
3.5	1
4	0.666 (3 d.p.)
4.5	0.333 (3 d.p.)
5	0
6	0

Table 8: The degree of membership for TFN set A for a range of numbers

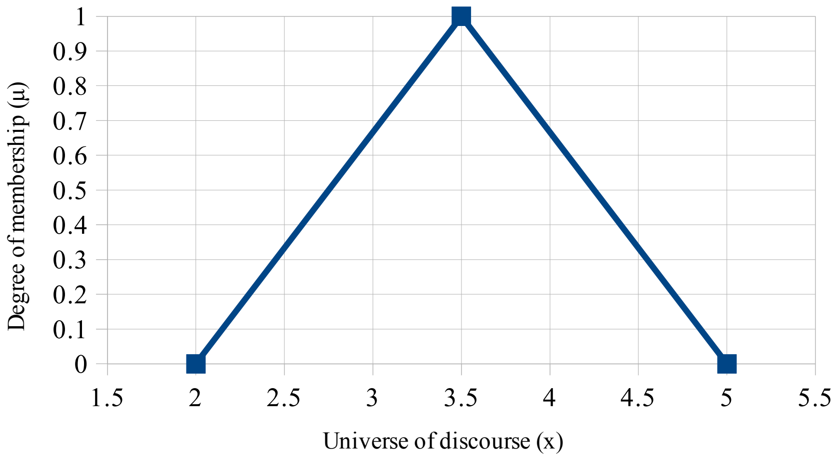

Figure 58: Line graph of the TFN A

A line graph can be used to visualise the numbers in a TFN set. This is shown in Figure 58. In the example of the TFN in equation (10), the value 3.5 may be incorrectly assumed to be certain, as $\mu_a(3.5)=1$. However the correct interoperation for the degree of membership of 1 is to consider the fuzzy number "being around 3.5" [142].

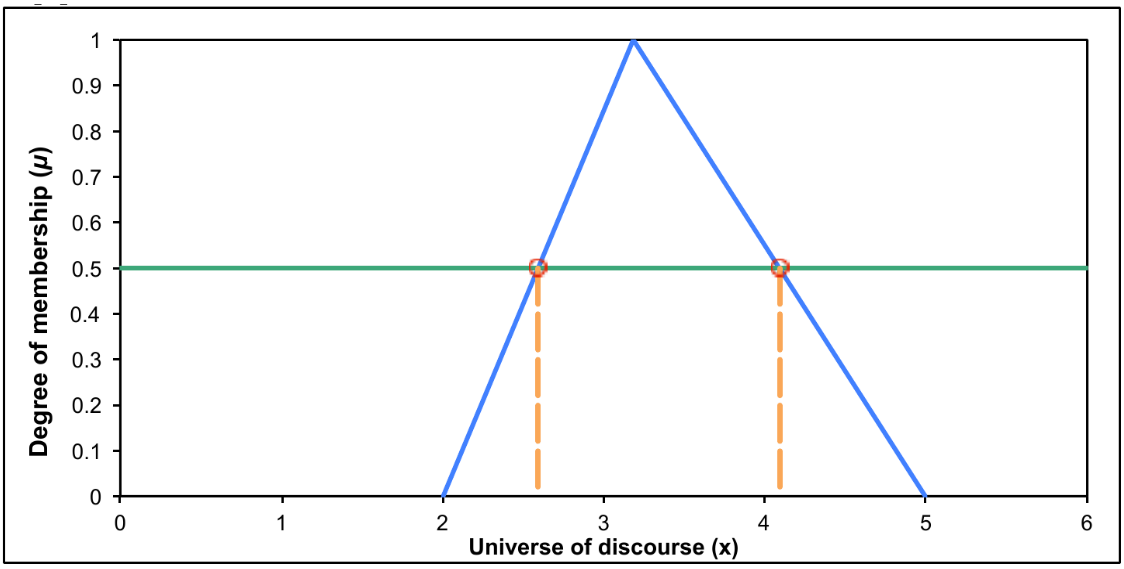

Figure 59: Line graph showing the intersection points of α = 0.5

In order to get absolute (also called crisp) values from a TFN set, α-cuts are used produce a set of crisp numbers that have a degree of membership of α or greater. An α-cut is a subset of the TFN set that only include values with a degree of membership greater than, or equal to a given α value. α-cuts are used to reduce the interval of the TFN therefore specifically excluding values that are distant from the mean value of the set. [69], [143]In Figure 58, the interval of crisp numbers with *α=0.5* would be the two line intersections if a line was plotted at each point of *x* and *μ=0.5,* as shown in Figure 59. The minimum and maximum values of the interval can also be mathematically calculated to produce an interval using Equation (11):

$$\begin{aligned} A_\alpha &= (a_1(\alpha), a_3(\alpha)) \\ &= [a_1 + \alpha(a_2 - a_1), a_3 - \alpha(a_3 - a_2)] \end{aligned} \tag{11}$$

Using the TFN in Equation 10 and *α = 0.5*, Equation (11) produces the crisp interval:

$$A_{\alpha=0.5} = (1.467, 2.967) \tag{12}$$

Yeh and Chang propose [134] that a larger *α* value represents the decision makers' degree of confidence in how they weighted each assessment criteria. In order to get a single crisp value from the crisp interval, Yeh and Chang use an attitude index *λ* between 0 and 1 to determine the chosen value. The attitude index is analogous to the decision makers' attitude towards risk with *λ=0* representing pessimism, *λ=0.5* representing a moderate view and *λ=1* representing an optimistic view. The index is calculated using Equation (13):

$$x_\alpha^\lambda = \lambda x_r^\alpha + (1 - \lambda) x_l^\alpha, \qquad 0 \le \lambda \le 1 \tag{13}$$

Using the values $A_{\alpha=0.5} = [1.467, 2.967]$ (representing a moderate degree of confidence in a person's confidence in their comparison, A pessimistic value (*λ=0*) would produce a crisp value of 1.467, a moderate value (*λ=0.5)* producing 2.217 and an optimistic value (*λ=1)* producing 2.967.

Where necessary, asthmatic operations can be performed on TFNs using the following equations using the TFNs $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$: [69], [144]

Addition:

$$\boldsymbol{A}(+)\boldsymbol{B} = (a_1 + b_1, a_2 + b_2, a_3 + b_3) \tag{14}$$

Subtraction:

$$\boldsymbol{A}(-)\boldsymbol{B} = (a_1 - b_1, a_2 - b_2, a_3 - b_3) \tag{15}$$

Multiplication:

$$\boldsymbol{A}(*)\boldsymbol{B}=(a_1*b_1, a_2*b_2, a_3*b_3) \tag{16}$$

Division:

$$\boldsymbol{A}(/)\boldsymbol{B}=(a_1/b_3, a_2/b_2, a_3/b_1) \tag{17}$$

Power:

$$\boldsymbol{A}^{\boldsymbol{n}}=(a_1^n, a_2^n, a_3^n) \tag{18}$$

Square root:

$$\sqrt{\boldsymbol{A}}=(\sqrt{a_1}, \sqrt{a_2}, \sqrt{a_3}) \tag{19}$$

The division operation presented in Equation (17) is done using the inversion of the TFN *B* to ensure the product of the operation retains the features of a TFN, specifically a low, mean and high value [145]. The multiplication of *A * A* is problematic as fuzzy arithmetic assumes both factors are instantiated independently, in this example the operation applies to the same set. This can cause inconsistencies, therefore the operation A^2 *(A squared)* should be used instead [146].

5.2.1 Example of using TFNs – Using stroke patient preferences to determine who will make choice for treatment

Part of the ethnographic phase of the DASH project involved the elicitation of stroke patient and clinician preferences for how an individual stroke patient's treatment should be determined. The overall purpose of the project was to identify the optimal pre-hospital and acute stroke care procedures by expanding the evidence base for future treatment interventions [138]. One outcome of this project was the production of a computerised tool to visualise the risks and benefits of treating a patient who had an ischaemic stroke with the thrombolytic drug alteplase [11].

A key theme of the DASH project was the role of shared-decision making (SDM) between the patient and clinician in order to improve engagement between the patient, family and clinician, with one strategy being decision-making aids [11]–[13].

SDM is also viewed as a mid-point between a paternalistic care model where the clinician makes the decisions regarding care, through to an informed model where the patient makes all of the choices [148].

To determine patient preferences for their degree of involvement in decisions about their treatment, *n=36* stroke patients and their relatives/carers were asked to complete the Control Preference Scale categories [133]. The participants' responses are summarised in Table 9.

Category	**Description**	**Number of responses**
4	*I prefer to make the final selection of my treatment after seriously considering my doctor's opinion.*	12
3	*I prefer that my doctor and I share responsibility for deciding which treatment is best for me.*	10
2	*I prefer that my doctor makes the final decision about which treatment will be used, but seriously considers my opinion.*	9
1	*I prefer to leave all decisions regarding my treatment to my doctor.*	5
	Total participants:	**36**

Table 9: Summary of stroke patients/carers' responses to their Decision making preference

Generalising the results in Table 9, the patients surveyed indicated a stronger preference for making informed decisions, follows by shared decision making, while a paternalistic decision was the least preferable. Applying Equation (8) to the values in Table 9, a TFN number *B* is produced in Equation (20).

$$B = (1, 2.810, 4) \quad (20)$$

Equation 9 can be re-written as:

$$\mu_a(x)=\begin{cases} 0, & x\leq 1 \\ (x-1)/(2.810-1), & 1\leq x\leq 2.810 \\ (4-x)/(4-2.810), & 2.810\leq x\leq 4 \\ 0, & x\geq 4 \end{cases} \quad (21)$$

Using the membership function in Equation (21) the degree of membership can be obtained for the values 1,2,3,4.

Number (corresponding to the categories of Table 9	Degree of Membership
1	0
2	0.552 (3 d.p.)
3	0.840 (3 d.p.)
4	0

Table 10:The degree of membership for TFN set B for the numbers corresponding to the Control Preference Scale category numbers

Using the TFN *B,* and the example scenario Thrombolytic stroke treatment scenario in Chapter 3, the following BPMN model was constructed:

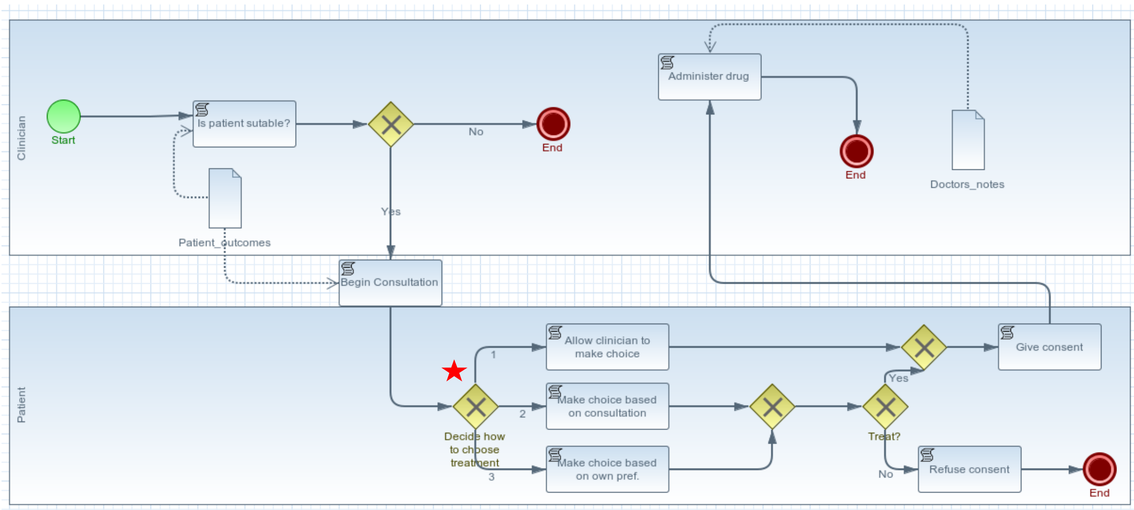

Figure 60: BPMN model of a Patient and Doctor deciding if the patient should receive a particular treatment

The BPMN model in Figure 60 depicts a scenario that starts with the Doctor assessing the patient to see if they are suitable for the drug altaplase (thrombolytic treatment). We assume that the modeller is designing a system to support the doctor and patient during the consultation process by visualising the risks of the drug, for example undesirable side effects. Therefore the objective of the model is to determine the sequence flow based on the characterisation of patients who would prefer to make to make the final decision whether to be treated, (indicating a high degree of patient autonomy), those who may prefer to consult with the doctor to make a shared decision (indicating a moderate degree of patient autonomy), or those who would prefer to delegate the decision to the doctor entirely (which would indicate a low degree of patient autonomy).

In the workflow model, both actors (Doctor and Patient) refer to a document called “Patient outcomes” If the patient is not suitable for treatment (i.e. does not meet the licensing criteria for the drug, such as having a high blood glucose level), then the model is terminated as the drug will not be administered.

If the patient is suitable for treatment, a shared process called “Begin Consultation” happens between the Clinician and the Patient, using the “Patient outcomes” document as part of this task. Next, the patient will need to decide to precede with the administration of the drug. Based on the categories in Table 9, three tasks were devised: “Allow clinician to make choice”, representing categories 3 and 4 in the Control Preference Scale as these are analogous to the paternalistic form of decision-making. The second task was “Make choice based on consultation”, analogous to SDM and category 2 of the Control Preference Scale. The finial option is “Make choice based on own pref” which is analogous to the informed model of decision making and corresponds to the first category on the Control Preference Scale.

In the scenario modelled, if the patient allows the clinician to make the choice, the drug will be administered once consent is given as the patient already meets the licensing criteria for the drug. This is analogous to the paternalistic care model, where the clinician makes the salient decisions for the patient [149].We assume as the patient has delegated the choice to the clinician, they will not refuse consent. If the patient chooses any other option, the patient will be involved to some extent in the decision whether or not they want the drug, and will either give consent for the drug to be administered, or refuse consent and the model will end without the “Administer drug” task being triggered. This is analogous to the shared-decision making pattern where the choice of treatment is shared between the patient and the clinician [149].

The gateway condition controlling the sequence flow (identified with a star (*) in Figure 61 for the three decisions the that could be made is set using the TFN *B,* set to the patient's corresponding Actor instance at run-time, α and λ values determined by the modeller at run-time.

α and λ values produce a crisp number by using Equations (11),(12) and (13). Considering patient autonomy and the possible choices for deciding treatment, $\alpha=0$ where it is assumed that patients have a low confidence in the value they chose for the survey (Control Preference Scale), the actual treatment option chosen is dominated by the λ value. A value of $\lambda=1$ (optimistic view of risk) is assumed that the patient will make the decision themselves (make final selection of choice), indicating a high degree of autonomy.

Proving patients with a higher degree of autonomy is regarded as beneficial to patients as it provides empowerment, improves their understanding of their options and is likely to increase the chances a patient will continue to adhere with their ongoing treatment. [11], [114], [150].

These additional features required the development of a Java Class for holding and manipulating TFNs and fields and methods for an Actor class, representing the decision maker, which is described in Chapter 6.

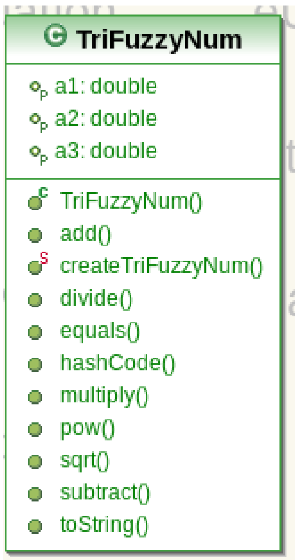

Figure 61: UML class diagram of the TriFuzzyNum class

Figure 61 Illustrates the fields and methods for the `TriFuzzyNum` class, which also provides methods for common mathematical operations on TFNs.

The following features are necessary for an Actor class to handle TFNs:

1. **Map structure to store taskId as key and a TFN as key value**
   This is necessary as other gateways may use TFNs to indicate the desired sequence flow.
2. **Add getter and setter methods for modifying and accessing α and λ values**
3. **Create `decide(taskID)` method**
   This method should return a crisp value based on the Actors TFN associated with the task and pre-set α and λ values.

Using the TFN provided in Equation 20 and the definitions for α and λ from Yeh and Chang, the BPMN model in Figure 60, was run with the following α and λ *values*:

α value	**linguistic assumption**	***λ value***	**linguistic assumption**
0	Participants' have **low confidence** in the value they chose in the survey	0	A **pessimistic** view of risk
0.5	Participants' have **moderate confidence** in the value they chose in the survey	0.5	A **moderate** view of risk
1	Participants' have high **confidence** in the value they chose in the survey	1	**An optimistic** view of risk

Table 11: Summary of Yeh and Chang's linguistic equivalents of α and λ values

α and λ values	**Crisp value**	**Task triggered**
$\alpha = 0, \lambda = 0$	1	Allow clinician to make decision
$\alpha = 0, \lambda = 0.5$	2.5 (rounded to 3)	Make choice based on consultation
$\alpha = 0, \lambda = 1$	4	Make choice based on own pref.
$\alpha = 0.5, \lambda = 0$	1.905 (rounded to 2)	Allow clinician to make decision
$\alpha = 0.5, \lambda = 0.5$	2.655 (rounded to 3)	Make choice based on consultation
$\alpha = 0.5, \lambda = 1$	3.405 (rounded to 3)	Make choice based on consultation
$\alpha = 1, \lambda = 0$	2.810 (rounded to 3)	Make choice based on consultation
$\alpha = 1, \lambda = 0.5$	2.810 (rounded to 3)	Make choice based on consultation
$\alpha = 1, \lambda = 1$	2.810 (rounded to 3)	Make choice based on consultation

Table 12: Results of running workflow model

The results in Table 12 show that when $\alpha = 1$, the return value is always 2.810, the weighted average of the TFN. This is consistent with the behaviour of TFNs as an α value of 1 intersects the value with the highest degree of membership, which corresponds to the weighted average. In terms of the data in Table 9, this assumes that a decision maker has a high degree of confidence in the answer they gave, therefore the answer is more likely to be around a mid-point value in the dataset.

If it is assumed that the decision makers had low confidence in their choices, the crisp preference values return by the Actor instance are roughly equivalent to the TFN when varying λ. This is interpreted as representing a person's overall view of risk is the greatest indicator of which open they will choose, and this choice would be made independently of their confidence in how they chose their option.

When $\alpha = 0.5,$ moderate and optimistic views of risk indicate a stronger preference for SDM and informed decision making. This is consistent with α-cuts as a value of $\alpha=0.5$ results in a set of crisp numbers that are closer to the geometric mean than a lower α value.

The use of α-cuts should be carefully considered. When using α-cuts, an assumption is made regarding the decision makers' self-perceived confidence in their choice of option. Unless participants are asked to rate their confidence using a similar scale to the eventual α-cut value chosen, the choice of α-cut value is subjective. As a value of α=1 always returns the geometric mean, its application for modelling subjectivity is not useful as it does not reflect the variance seen in the set.

The use of TFNs to model subjectivity is only as good as the data collected and the how well the workflow model reflects the real-world scenario it encapsulates. The model presented in Figure 60 is based on an empirical understanding of the tasks undertaken by a clinician and patient during the diagnostic and consultation process, yet is still a generalisation of the actual activities that may take place between a patient and clinician in the scenario. An important procedure after the development of a model, and associated data that defines its control-flow, is to verify the model and results are a reasonable reflection of the real-world scenario with the stakeholders involved in the project.

5.2.2 TFNs and TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution)

If a decision making scenario required participants to make a decision based on the assessment of multiple criteria, this can be simulated using the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) algorithm. TOPSIS works on the principle that the optimum choice should be closest to the theoretical best choice and furthest from the theoretical worst choice [151], [152].

Typically, a TOPSIS approach assumes that decision makers perform pairwise comparisons to appraise one option with another rather than comparing options against a fixed criteria [18]. The pairwise comparison process has been popularised as part of the broader work of the analytic hierarchy process (AHP) [139], [154], [155] where the salient concerns are the perceived importance of the decision-making criteria which are evaluated by each decision-maker.

AHP suggests a 1 to 9 ratio scale for comparing two criteria indicating the strength of their relative performance: [134], [156]

1. Equally important
2. Weak
3. Moderately more important
4. Moderate plus
5. Strongly more important
6. More than strongly important
7. Very strongly more important
8. Very, very strongly important
9. Extremely important

The ratio scale allows decision makers to weigh up each criterion based on their own subjective assessment based on their own attitudes and preferences. This process produces a qualitative decision matrix that can be used in the TOPSIS algorithm. A drawback of the pairwise comparison process is the time required for each decision maker to perform the comparisons [9], [22]. Therefore, Yeh and Chang recommend limiting the number of elements in a pairwise comparison to no more than seven [134].

The principle steps of the TOPSIS algorithm are outlined below and are based on using crisp values [158]:

Pre-step: Construct a decision matrix $X\,(m * n)$ consisting of individual attributes, x_{ij} representing option i in respect of criterion j. A separate matrix corresponding of weights w_j which are also provided for each individual criterion j.

Step 1: Normalise the decision matrix using the equation:

$$y_{ij} = x_{ij} / \sqrt{\sum_{i=1}^{m} x_{ij}^2} \tag{22}$$

Step 2: Construct the weighted normalised decision matrix:

$$z_{ij} = w_j \, y_{ij} \; where \\ i = 1,2,\dots, m \, ; \; j = 1,2,\dots n \tag{23}$$

Step 3: Determine the ideal and worst ideal solutions. The ideal solution is a theoretical solution that contains the highest weighted value of each criterion set, where was the worst ideal solution contains the lowest weighted value of each criterion set.

Ideal solution:

$$A^{+} = (z_1^{+}, \dots, z_n^{+}) \; where \\ z_j^{+} = max\,\{ z_{1j}, \dots, z_{ij} \} \, , \, j = 1,2,\dots, n \tag{24}$$

Worst ideal solution:

$$A^{+} = (z_1^{-}, \dots, z_n^{-}) \; where \\ z_j^{-} = min\,\{ z_{1j}, \dots, z_{ij} \} \, , \, j = 1,2,\dots, n \tag{25}$$

Step 4: Determine the separation of each option from the ideal and worst ideal solution. This is necessary as the ideal solution should be as close as possible to the ideal solution. [152], [158]

Separation from ideal solution:

$$S_i^{+} = \sqrt{\sum (z_j^{+} - z_{ij})^2} \tag{26}$$

Separation from worst ideal solution:

$$S_i^{-} = \sqrt{\sum (z_j^{-} - z_{ij})^2} \tag{27}$$

Step 5: Calculate the relative closeness of each option to the ideal solution:

$$C_i = \frac{S_i^{-}}{(S_i^{+} - S_i^{-})} \tag{28}$$

The ideal solution has a theoretical closeness value of 1, as each criteria has the highest possible ideal value, therefore the best option is the one that is closest to 1 [134], [158].

Yeh and Chang use TFNs, the arithmetic operations provided in equations (14) to (19) to extend the TOPSIS algorithm to incorporate TFNs representing the subjective appraisal of purchasing aeroplanes by multiple decision makers [134]. Aeroplanes were used as an example as decision makers responsible for purchasing planes are required to appraise each type against multiple criteria, such as fuel economy, noise and maximum range. By varying both α and λ values of the TFNs produced by equation (28), the process produced the same ranking order for each model of aeroplane, which Yeh and Chang believing that this indicated the approach was stable for the specific problem and decision makers' preferences.

5.3 Workflow patterns for BDI agents

This section outlines how BPMN workflow models can be used to implement the functionality for Belief-Desire-Intention agents. If a modeller wishes to simulate human appraising and coping activities [57], a BDI software agent can provide the functionality that encapsulates these psychological processes [60], [61].

For each BDI agent designed feature outlined by Maalal and Addou [60] the corresponding BPMN feature from the BPMN specification [41] is identified and described in the rest of this section:

Roles

Roles correspond to the agent's functionalities, which are analogous to the BPMN *Task* element, which represents a discrete unit of work done by a human actor or another process.

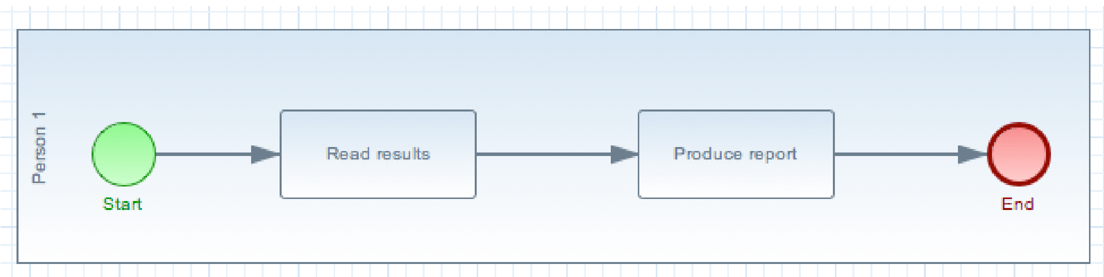

Figure 62: Roles mapped to BPMN Tasks

Attributes

An attribute is information the Agent possesses. The BPMN *DataObject* provides a visual representation of attributes used in the workflow model. However the specification does not provide the implementation, which instead requires the modeller to specify this for the runtime engine.

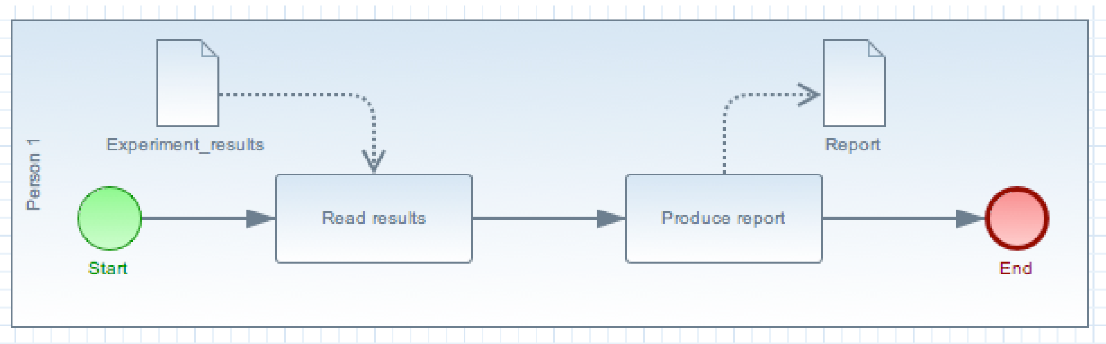

Figure 63: Attributes mapped to BPMN data objects

Action

Maalal and Addou define actions as all of the possible actions the agent can execute on their environment [60]. This property is implicitly provided using multiple BPMN constructs, firstly *Tasks* to represent the actions an agent can perform, *Throw/Catch/Boundary Events* to signify specific situations that occur. Finally, connections between elements, such as *Sequence Flows* between tasks, events and gateways and *Message Flows* between Data Objects and Tasks and Lanes representing multi-user choreographies.

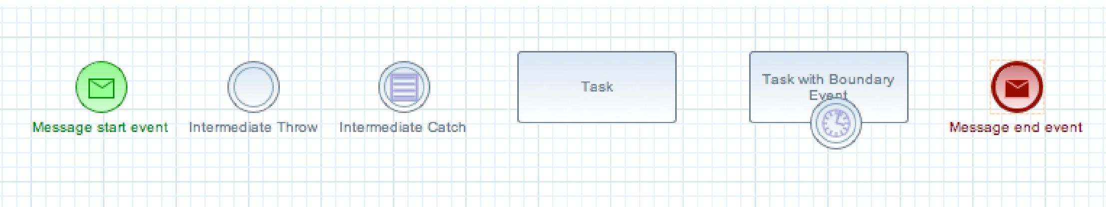

Figure 64: BPMN Events and tasks analogous to agent Actions

Decisions

The decisions an agent makes analogous to BPMN *Gateways, which define the direction,* a sequence flow will take, depending on a condition or event.

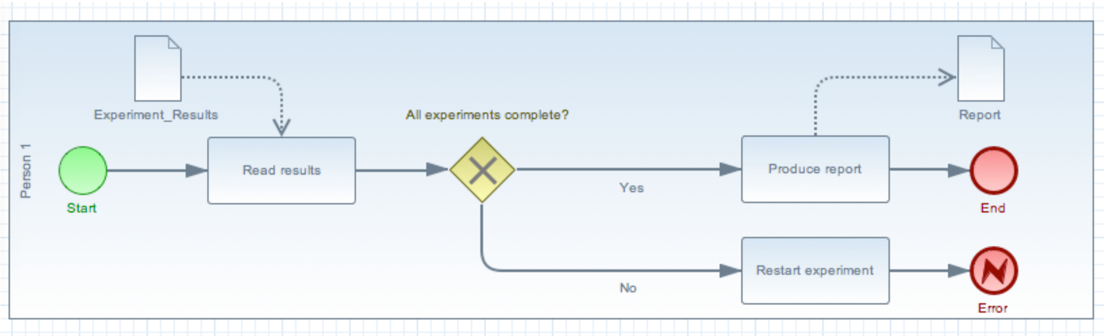

Figure 65: A decision mapped to a BPMN gateway

Interaction

Agent interaction with other agents is analogous to BPMN's multi-user choreography design patterns, specifically with Lanes representing individual Agents, and message flows representing the interaction between the agents. This is summarised in Figure 66.

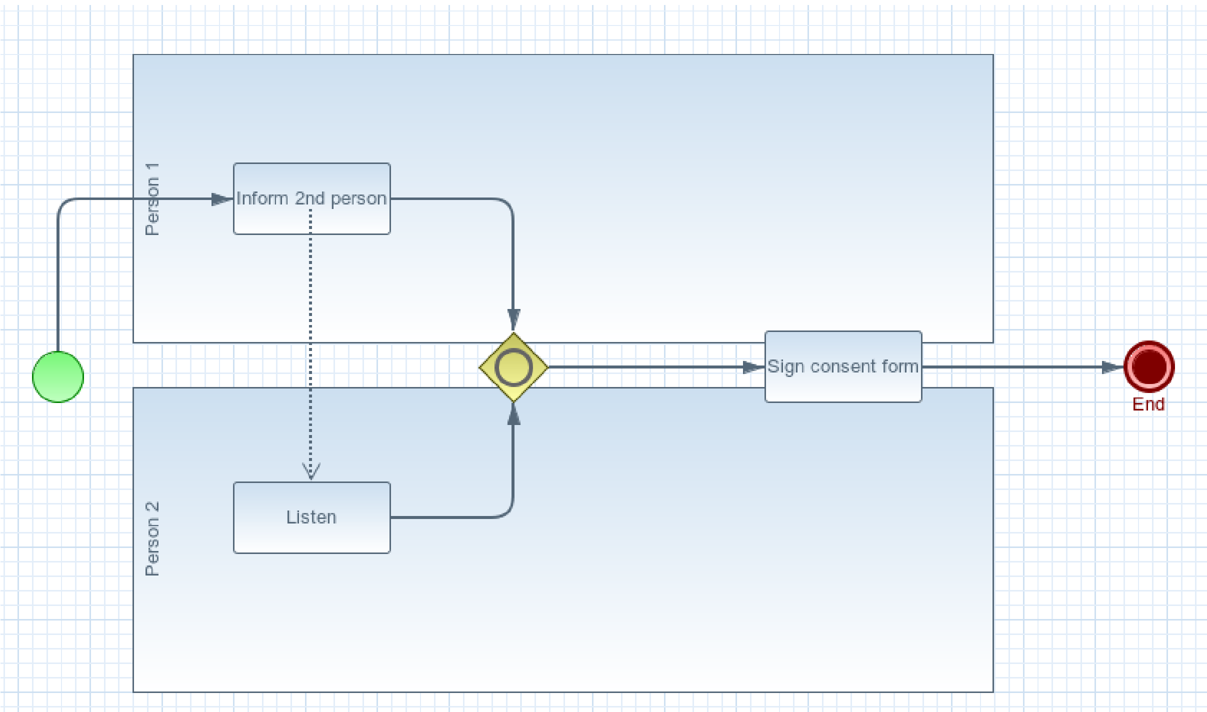

Figure 66: BPMN workflow model showing contemporary collaboration/interaction design pattern on the left, and BPMNdm shared task design pattern on the right

On the left hand side of Figure 66, the BPMN workflow model depicts the conventional design pattern [36], [41] of two people interacting, indicated by the message flow from the task “Inform 2nd Person” in the Person 1 lane, connecting to the “Listen” task in the Person 2 lane. The right hand side of the model shows the *shared task* design pattern defined in Chapter 3, indicating a collaboration on the shared task “Sign consent form”.

Perception and Beliefs

Perception and Beliefs are subjective by nature. An agent’s beliefs about the environment and of other entities may be incorrect, incomplete or uncertain [60]. In this context, Yeh and Chang’s approach of using triangular fuzzy numbers to represent subjective data can be incorporated into the agent’s functionality to encapsulate and allow the use of the information in a similar manner to crisp or absolute data.

Goals, Intentions and Desires

BPMN does not provide a visual representation of these features, though these features can be inferred by the BPMN tasks belonging to the agent, as well as the runtime logic. In the case of desires, these can be visualised using default outgoing sequence flows from Gateways, or *Weighted Paths* from BPMNdm, in order to show preference.

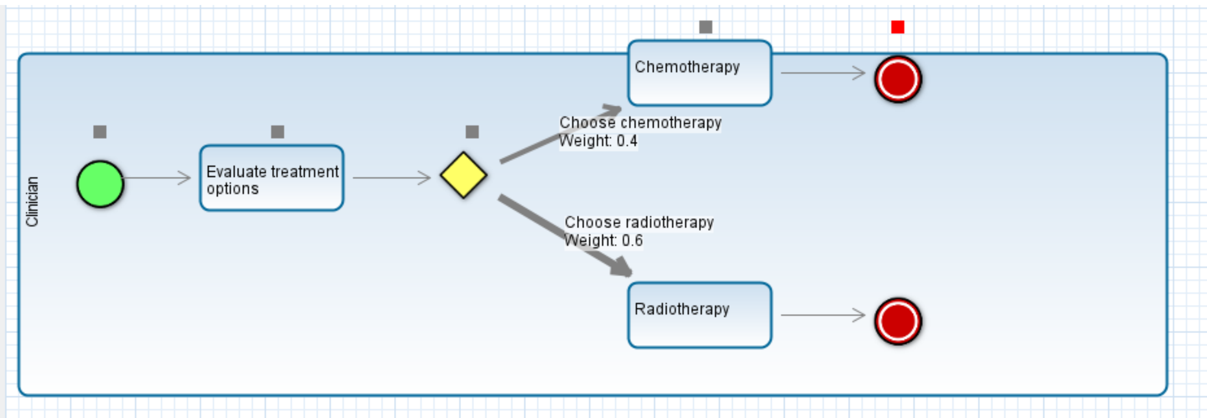

Figure 67: Weighted Path BPMNdm visualisation from Chapter 5

The BPMNdm workflow depicted in Figure 67 from Chapter 5 shows two treatment options for cancer, with a larger weight for Radiotherapy. In this example, the clinician’s intention and/or desire could be to give radiotherapy rather than chemotherapy. Therefore, the sequence flow to radiotherapy has a larger weight than the sequence flow to chemotherapy.

In this model, it can be assumed that both Chemotherapy and Radiotherapy would fulfil the goal of offering a treatment to a patient, however the modeller would be required to manually incorporate this into the model, either by setting a condition when a desired task is executed. The goal element is not visualised in Figure 67, however if the goal was to eliminate the cancer, this could be evaluated once the model is executed by analysing the literature on the success rate of radiotherapy for the type and stage of the cancer.

A summary of the BDI agent features and their corresponding mapping to the BPMN formalism, or to the jBPM system is provided in Table 13.

Agent feature	Mapping to BPMN formalism or jBPM system feature
Roles	Tasks
Attributes	Data objects
Perception	*Relies on runtime engine* *i.e. Drools rules and jBPM*
Action	Tasks, Events, outgoing arcs
Interaction	Events, outgoing arcs
Decision	Gateway
Beliefs	*Relies on runtime engine, i.e. Drools and jBPM*
Goals and Intentions	Sequence of tasks to be completed, or fulfilment of condition based on the traversal of tasks.
Desires	Combination of Runtime Engine components, and appropriate tasks.

Table 13: BDI Agent features and Corresponding BPMN constructs

5.4 Discussion

This chapter has presented the usage of Triangular Fuzzy Numbers (TFNs) as a method for encapsulating and using subjective data in a workflow model, for example controlling the sequence flow of the model at certain gateways. In conjunction with a scalar preference set, when simulating decision-making, TFNs allow the modeller to simulate user responses based upon their perceived confidence in their choices and attitudes to risk. As the TFN represents an interval of confidence, the modeller can specify whether a decision is made based on a person's moderate attitude to risk, therefore a value close to the geometric mean of the set, or with an optimistic view of risk, which is closer to the optimum (also called supremum) value of the set, or based on a pessimistic view of risk, based on the worst (also called infimum) value of the set.

Using TFNs in this manner requires that the TFN values represent scalar values, like those used in Likert scales. In this chapter, a practical example of how TFNs can be used is provided by aggregating patient responses to Control Preference Scale categories into a TFN. The control preference scale has clear, discrete scalar categories, where a low value is associated with the patient's preference to delegate the decision to their doctor, and a high value where the patient prefers to make the choice themselves (termed λ values). The responses have been assumed to correlate with the patient's attitude towards risk, where a person with an optimistic attitude to risk and preference to make an informed viewpoint decision (make final selection of choice) will likely be comfortable with making the decision made about treatment. Conversely, a person who is more pessimistic about risk and less confident in their choices will be more likely to delegate the decision to someone they perceive as an expert in the area (i.e. a doctor or similar type of expert in the given area). A practical example is provided showing how TFNs can be incorporated into a BPMN workflow model to determine the sequence flow of the workflow model. In this chapter, the aggregated Control Preference Scale of patients are used to assist the construction of a BPMN workflow model that maps out the treatment process for stroke patients, based on their preference of being involved in the decision to be administer the thrombolytic drug alteplase.

In order to produce a crisp value for the outgoing sequence flow in a BPMN model, a second consideration is the type of question that was put to the participants. For questions that do not indicate an attitude to risk (i.e. given preferences for a preferred colour), then λ values cannot be used and the modeller must select a value based on the crisp intervals produced by α-cuts.

A further extension to deal with subjectivity is to use the TOPSIS algorithm that is used to solve MCDM problems. The TOPSIS algorithm is regarded as being appropriate for encapsulating decision making activities undertaken by human actors [152]. However, the TOPSIS algorithm assumes that participants have used a pairwise comparison process to compare one option against another option using a set of different criteria. Each participant should indicate the perceived importance of each criteria, with one method being the AHP 1 to 9 ratio scale. This assumption means that the TOPSIS approach is only suable for scenarios where comparisons are made and multiple criterion are evaluated as part of the decision making process.

The chapter concluded with an overview of how workflow models can fulfil the requirements of Belief-Desire-Intention agents. Each salient feature of a BDI agent is mapped to the corresponding feature of the BPMN formalism. The BDI features Belief and Perception require additional functionality to be provided beyond the BPMN model, such as the use of Drools rules and the underline workflow engine to provide additional implementation details that are out of scope of the BPMN formalism.

This chapter does not provide an implementation for BDI agents or TOPSIS, instead discussing how they could be implemented by by describing the steps necessary for their incorporation into the jBPM system and how BPMN workflow models can satisfy the design features of BDI agents.

Chapter 6 - Technical implementations

6.1 Introduction

This chapter shows our methodology of incorporating the BPMNdm extensions developed in Chapter 3 into BPMN workflow models, as well as how these extensions can be visualised. Second, this chapter also shows our methodology of incorporating our Drools rules developed in Chapter 4 into BPMN workflow models.

This chapter provides a description of how we provide support for the visualisation of BPMNdm extensions, offer design suggestions based upon the workflow design patterns of BPMNdm models and use the interaction rules using the jBPM 5 workflow engine.

The chapter will first describe the additional functionality written for jBPM 5 to allow the visualisation of the BPMNdm elements.

Next, the chapter will describe how our BPMNdm extensions can be interoperated into BPMN workflow models and show the extensions being visualised with our visualisation tool.

Finally, the chapter will describe how the interaction rules developed in Chapter 4 can be incorporated into a BPMN workflow model and a sample implementation is provided, as well as the classes developed to encapsulate the decision making activities undertaken by meeting participants, or actors in a decision making scenario.

6.2 Overview of jBPM 5 additions to support BPMNdm and interaction rules

jBPM 5 provides facilities for creating BPMN 2.0 models as well as a workflow engine to execute these models [159]. jBPM has functionality to run workflow models either as a Java program running on a local computer, or to act as a server to allow the model to be run using a web interface [159]. In this thesis, we run the jBPM 5 system as a desktop application. The desktop version of jBPM does not visualise the runtime execution of BPMN workflow models, Yeong and Harari developed a set of Java classes to visualise a BPMN workflow model without the need of the BPMN 2.0 modeller [160].

For this thesis, we extended Yeong and Harari's visualisation class to show the real-time execution of a BPMN workflow model and visualise the BPMNdm extensions defined in Chapter 3. We have named this the BPMdm Visualiser. To visualise the execution of a BPMN model in real-time, the visualiser implants the jBPM `ProcessEventListener` interface [161] which allows developers to write code that captures the state of the workflow model at every node the workflow engine traverses.

To demonstrate the usage of the BPMNdm visualiser, a simple BPMN model was constructed; a Swimlane representing a clinician who has two tasks: "See Patient 1" and "See Patient 2".

Step 1: Develop the BPMN model using the BPMN 2.0 modeller provided as part of the jBPM suite.

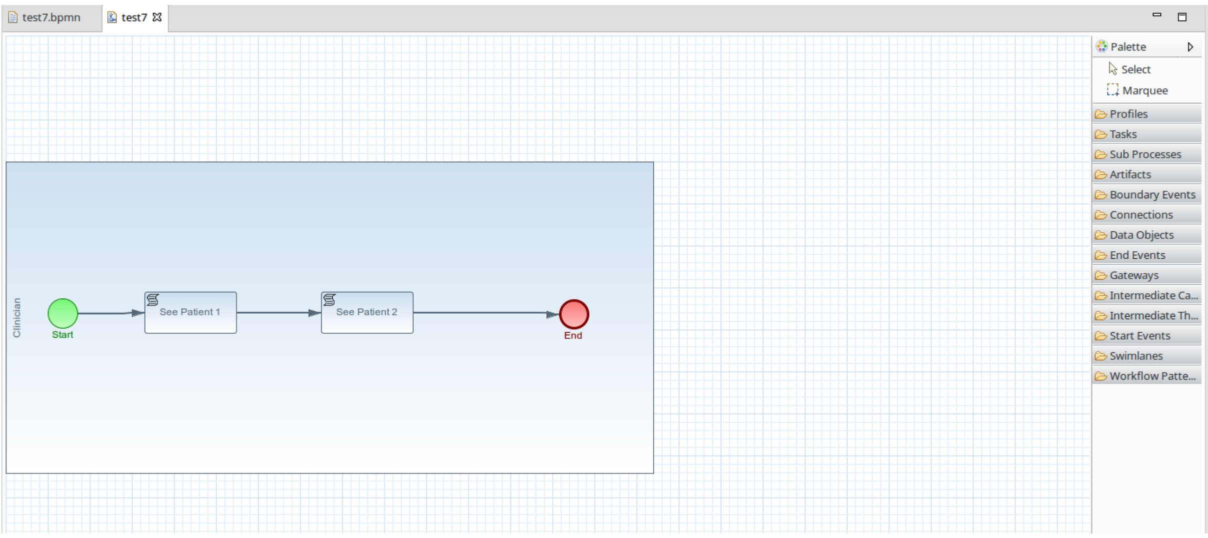

Figure 68: Developing a BPMN workflow model in the BPMN 2.0 modeller

Figure 68 shows the BPMN model being developed using the BPMN 2.0 modeller included with the jBPM system installation.

Step 2: Create a jBPM BPMN project and add the visualiser as an event listener.

When a BPMN project is created in jBPM, the software package automatically generates the necessary Java Main class to run the BPMN workflow model. The code generated is shown in Figure 69 with modifications shown in bold.

```
public class Main {

private static final long serialVersionUID = 1L;
public static KnowledgeBase kbase;

public static void main(String[] args) throws Exception {


kbase = readKnowledgeBase();
StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();

DrawEventListener del = new DrawEventListener("test.bpmn");
ksession.addEventListener(del);

WorkflowProcessInstance test = (WorkflowProcessInstance)
ksession.startProcess("com.sample.bpmn");


}

private static StatefulKnowledgeSession restoreSession(
      StatefulKnowledgeSession ksession, boolean b) {
      return ksession;
}



private static KnowledgeBase readKnowledgeBase() throws Exception {
KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();
      kbuilder.add(ResourceFactory.newClassPathResource("test.bpmn"),
ResourceType.BPMN2);
            return kbuilder.newKnowledgeBase();
}


}
```

Figure 69: Java code used to execute the BPMN model, with modifications shown in bold.

The additional code in Figure 69 is summarised as follows:

```
1. DrawEventListener del = new DrawEventListener("test.bpmn");
2. ksession.addEventListener(del);
```

The first line of code creates a new instance of `DrawEventListiner`, which contains the BPMNdm Visualiser. It requires the BPMN model file (`test.bpmn`) as a parameter. The second line inserts the `DrawEventListiner` instance into the jBPM workflow engine, in order to allow the BPMNdm Visualiser to detect the changes in the model execution.

Step 3: Run the jBPM process.

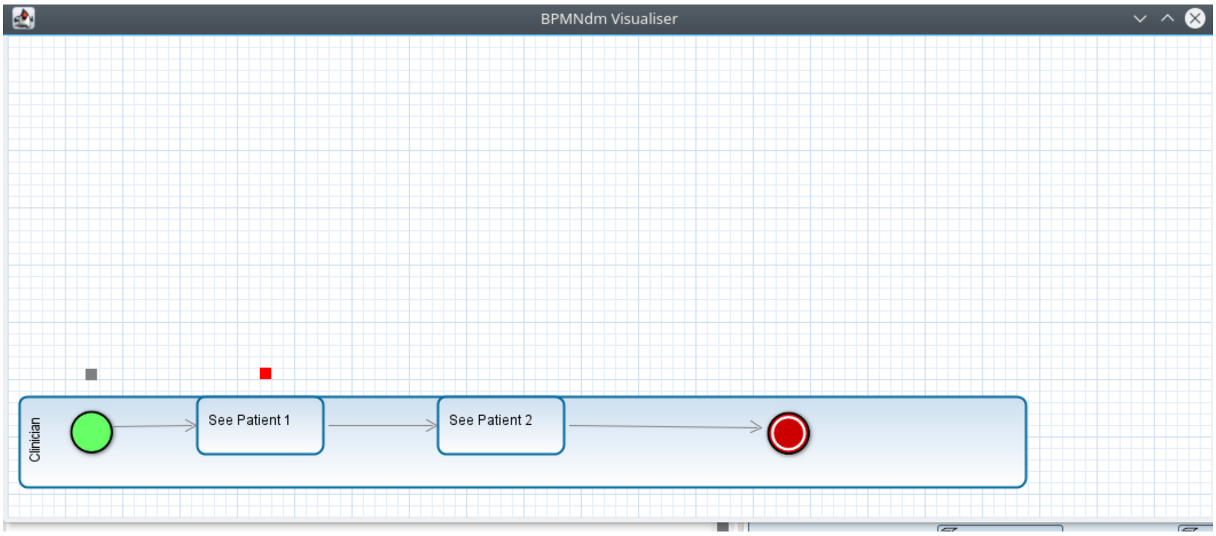

Figure 70: BPMNdm Visualiser showing the execution of a BPMN workflow model

When the jBPM process is run, the BPPMdm visualiser appears and shows the execution of the workflow model. Figure 70 shows the BPMNdm Visualiser showing the execution of the task "See Patient 1" while Figure 71 shows the end of the process. The current node being executed is indicated with a red square above its icon, with the previously executed nodes in the sequence flows being indicated with grey squares.

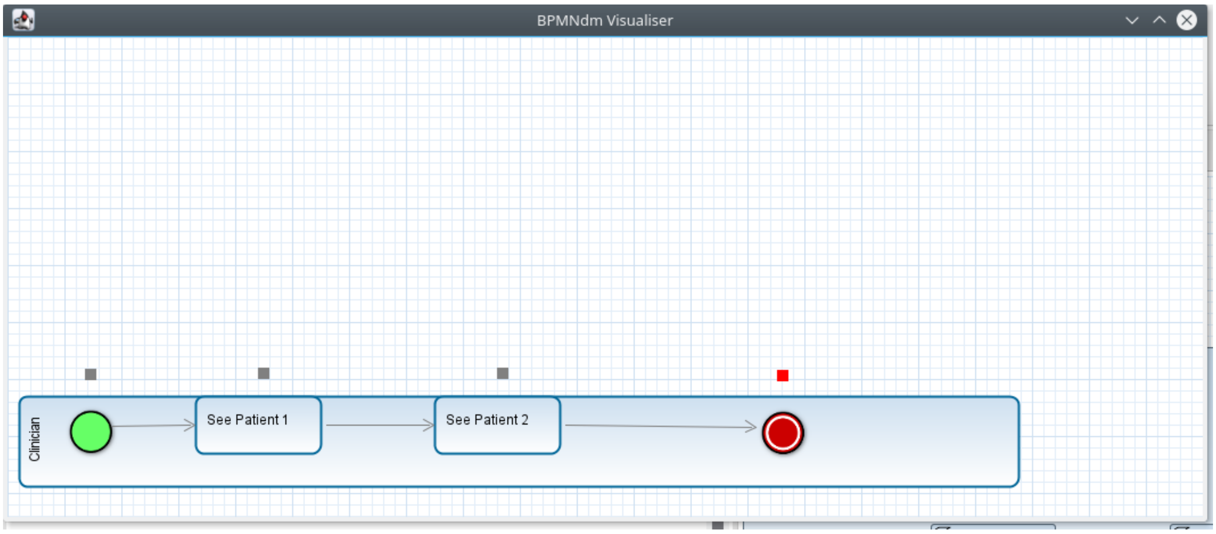

Figure 71: BPMdm Visualiser showing the end of the execution of a workflow model

In the rest of this chapter, the BPMNdm Visualiser was extended to visualise the extensions developed in Chapter 3, as well as provide design suggestions for decision support tools based on the workflow patterns developed by the system designer. These extensions are discussed in the following section of this chapter.

6.3 BPMNdm implementation

Chapter 3 described the development of four extensions to the BPMN formalism with the aim of better describing decision making scenarios using BPMN workflow models. The extensions described were as follows:

Proposed BPMN construct	Description
Cost Swimlane	Modified Swimlane that represents increasing or decreasing cost of a scale with child nodes
Time Swimlane	Modified Swimlane that represents increasing time on a scale with child nodes
Scoped and Workflow Data objects	Scope Data objects are owned by an actor and not accessible to other actors unless shared. Workflow data objects are accessible to all users.
Weighted Paths	Weighted paths represent the predetermined preference of an outgoing edge

Table 14: Summary of the extended BPMN formalism constructs derived from the decision-making scenarios from Chapter 3

Each extension was formally defined using a modified version of Stroppi et al.'s methodology of defining BPMN extensions from a set of requirements detailed in Chapter 3 [42]. For each extension, the XML Scheme Definition (XSD) from Chapter 3 is presented with an accompanying description and example of each extensions' technical implementation.

All of the extensions and code detailed in this chapter were implanted on the Java based jBPM 5 workflow engine. As the BPMN specification does not defined how extensions should be visualised [41], [42], the BPMNdm Visualiser discussed in Section 6.2 was developed to provide the facility of displaying the graphical notations for each extension.

The rest of this section is structured as follows: For each extension described in Table 14:

Step 1: Description of each the BPMNdm extension. A short description is provided of the extension.

Step 2: Display the BPMNdm extension XSD from Chapter 3 The XML schema definition (XSD) for each extension developed in Chapter 3 is presented again.

Step 3: Show the BPMN XML source of a workflow model that will feature the given BPMNdm extension, referencing of the BPMNdm XSD from Step 2 and the extension being defined.

The XSD file specifies the expected structure of the extension. The BPMN workflow file specifies the usage of the extension, and the extension data is provided in the format defined in each XSD file.

Step 4: Discuss any specific implementation details and show the BPMNdm model running in the BPMNdm Visualiser.

Depending on the extension, additional code is required to allow the jBPM workflow engine to interpret the extension.

6.3.1 Cost swimlane

Step 1: description of cost swimlane:

A cost swimlane provides a scalar representation of the cost value given to individual tasks on a BPMN workflow model. Like timelines, a cost swimlane has a visual representation of cost intervals and a cumulative value of the cost occurred at each interval. Cost swimlanes contain `costTasks` which have the attribute cost assigned to each instance. These are depicted in the XSD for cost swimlanes in Figure 72.

Step 2: Cost XSD from Chapter 3:

```
<?xml version="1.0" encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">


<group name="costline">

        <sequence>

                <element name="costInterval" type="double" minOccurs="1"
maxOccurs="unbounded" />

        </sequence>

</group>


<group name="costTask">

        <sequence>

                <element name="cost" type="double" minOccurs="1" maxOccurs="1" />

</sequence>

</group>


</schema>
```

Figure 72: Figure 2: XML schema definition (XSD) for cost swimlane from Chapter 3

The XSD presented in Figure 72 is used in the BPMN model presented in Figure 73. The model depicts a single cost swimlane with two cost interval elements, 10.00 and 15.00. The model has two `costTasks` which contain the values 10.00 and 5.00 respectively. In this example, the unit of cost is minutes. These are associated with each individual cost swimlane interval element.

Step 3: BPMN XML model showing XSD being used:

```
<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:java="http://www.java.com/javaTypes"
xmlns:tns="http://www.jboss.org/drools" xmlns:cl="http://www.ncl.ac.ukcom/myns"
xmlns="http://www.jboss.org/drools"
xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL BPMN20.xsd
http://www.jboss.org/drools drools.xsd http://www.bpsim.org/schemas/1.0 bpsim.xsd"
id="Definition" exporter="org.eclipse.bpmn2.modeler.core" exporterVersion="1.1.2.Final"
expressionLanguage="http://www.mvel.org/2.0" targetNamespace="http://www.jboss.org/drools"
typeLanguage="http://www.java.com/javaTypes">

  <bpmn2:import importType="http://www.w3.org/2001/XMLSchema" location="src/costLine.xsd"
namespace="http://www.ncl.ac.ukcom/myns" />

…

<bpmn2:laneSet id="LaneSet_1" name="Lane Set 1">
      <bpmn2:lane id="Lane_4" name="Clinician">
        <bpmn2:extensionElements>
          <cl:costInterval>10.00</cl:costInterval>
          <cl:costInterval>15.00</cl:costInterval>
        </bpmn2:extensionElements>
        <bpmn2:flowNodeRef>StartEvent_1</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>EndEvent_1</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>ScriptTask_1</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>ScriptTask_3</bpmn2:flowNodeRef>
      </bpmn2:lane>
    </bpmn2:laneSet>
...
<bpmn2:scriptTask id="ScriptTask_1" name="See boss" scriptFormat="http://www.java.com/java">
      <bpmn2:extensionElements>
        <cl:cost>10.00</cl:cost>
      </bpmn2:extensionElements>
      <bpmn2:incoming>SequenceFlow_1</bpmn2:incoming>
      <bpmn2:outgoing>SequenceFlow_6</bpmn2:outgoing>
      <bpmn2:script>System.out.println();</bpmn2:script>
    </bpmn2:scriptTask>
<bpmn2:scriptTask id="ScriptTask_3" name="See receptionist"
scriptFormat="http://www.java.com/java">
      <bpmn2:extensionElements>
        <cl:cost>5.00</cl:cost>
      </bpmn2:extensionElements>
      <bpmn2:incoming>SequenceFlow_6</bpmn2:incoming>
      <bpmn2:outgoing>SequenceFlow_9</bpmn2:outgoing>
      <bpmn2:script>System.out.println();</bpmn2:script>
    </bpmn2:scriptTask>
```

Figure 73: Cost swimlane XSD being used for the cost swimlane "clinician" and tasks "See boss" and "See receptionist".

Figure 74 Shows the BPMN model presented in Figure 73 running in the BPMNdm visualiser, showing the two cost tasks “Consult boss” and “Consult receptionist” rendered in a cost swimlane with the cumulative costs indicated under each task.

Step 4: BPMNdm cost swimlane in BPMNdm Visualiser:

Figure 74 shows a cost swimlane being visualised in the BPMNdm Visualiser. The time intervals 10:00 and 11:00 are plotted on the swimlane, along with the tasks associated with each time interval. Each task also contains the individual cost elements.

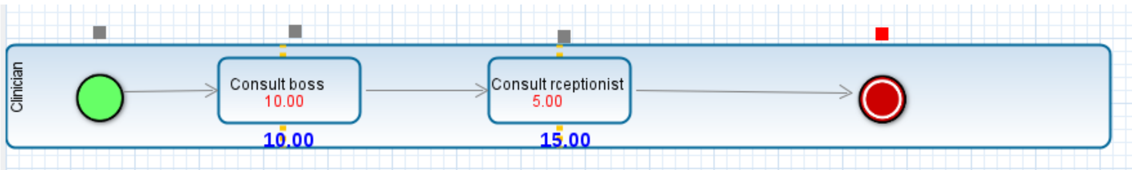

Figure 74: BPMN model displayed in the BPMNdm visualiser, the CostLine and cost tasks.

6.3.2 Timeline

Step 1: Description of a timeline

A Timeline is a swimlane that contains one or more time intervals which are displayed directly on the swimlane. These specially represent when a given task must be run by. In the example of thrombolytic stroke treatment, this would mean that the drug must be administered no later than 4.5 hours after the stroke event. A timeline contains one or more `timeTasks` which have the attributes `startTime`, indicating where on the timeline the task starts and a `duration` which indicates the length of time the task requires. These attributes are of the type `time` which represents a time in the 24-hour format hh:mm for `startTime` represents the `duration` in the format of hh:mm. The runtime engine evaluates these attributes to ensure they are valid. The attributes for the timeline swimlane and `timeTasks` are detailed in Figure 75.

Step 2: Timeline XSD from Chapter 3:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">


<group name="timeline">
        <sequence>
                <element name="time" type="time" minOccurs="1" maxOccurs="unbounded" />
        </sequence>
</group>


<group name="timeTask">
        <sequence>
                <element name="startTime" type="time" minOccurs="1" maxOccurs="1" />
                <element name="duration" type="time" minOccurs="1" maxOccurs="1" />
</sequence>


</group>


</schema>
```

Figure 75: XML schema definition (XSD) for Timeline from Chapter 3

Step 3: BPMN XML model showing XSD being used:

Using the XSD in Figure 75, Figure 76 illustrates a timeline being used in a BPMN model. In the example, two `timeTasks` are provided, “See Patient 1” which starts at 10:00 and lasts for 30 minutes and a second tasks, “See Patient 2” which starts at 11:00 and lasts for 30 minutes. Each individual `timeTasks` corresponds to one time element that is plotted on the timeLine, which is called “Clinician”.

```
<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:java="http://www.java.com/javaTypes"
xmlns:tns="http://www.jboss.org/drools" xmlns:tl="http://www.ncl.ac.ukcom/myns"
xmlns="http://www.jboss.org/drools"
xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL BPMN20.xsd
http://www.jboss.org/drools drools.xsd http://www.bpsim.org/schemas/1.0 bpsim.xsd"
id="Definition" exporter="org.eclipse.bpmn2.modeler.core" exporterVersion="1.1.2.Final"
expressionLanguage="http://www.mvel.org/2.0" targetNamespace="http://www.jboss.org/drools"
typeLanguage="http://www.java.com/javaTypes">

  <bpmn2:import importType="http://www.w3.org/2001/XMLSchema" location="src/timeLine.xsd"
namespace="http://www.ncl.ac.ukcom/myns" /> ...

<bpmn2:laneSet id="LaneSet_1" name="Lane Set 1">

      <bpmn2:lane id="Lane_4" name="Clinician">

        <bpmn2:extensionElements>

          <tl:time>10:00</tl:time>

          <tl:time>11:00</tl:time>

        </bpmn2:extensionElements>

        <bpmn2:flowNodeRef>StartEvent_1</bpmn2:flowNodeRef>

        <bpmn2:flowNodeRef>EndEvent_1</bpmn2:flowNodeRef>

        <bpmn2:flowNodeRef>ScriptTask_1</bpmn2:flowNodeRef>

        <bpmn2:flowNodeRef>ScriptTask_3</bpmn2:flowNodeRef>

      </bpmn2:lane>

    </bpmn2:laneSet>

...

<bpmn2:scriptTask id="ScriptTask_1" name="See Patient 1"
scriptFormat="http://www.java.com/java">

      <bpmn2:extensionElements>

        <tl:startTime>10:00</tl:startTime>

        <tl:duration>00:30</tl:duration>

      </bpmn2:extensionElements>

      <bpmn2:incoming>SequenceFlow_1</bpmn2:incoming>

      <bpmn2:outgoing>SequenceFlow_6</bpmn2:outgoing>

      <bpmn2:script>System.out.println();</bpmn2:script>

    </bpmn2:scriptTask>

<bpmn2:scriptTask id="ScriptTask_3" name="See Patient 2"
scriptFormat="http://www.java.com/java">

      <bpmn2:extensionElements>

        <tl:startTime>11:00</tl:startTime>

        <tl:duration>00:30</tl:duration>

      </bpmn2:extensionElements>

      <bpmn2:incoming>SequenceFlow_6</bpmn2:incoming>

      <bpmn2:outgoing>SequenceFlow_9</bpmn2:outgoing>

      <bpmn2:script>System.out.println();</bpmn2:script>

    </bpmn2:scriptTask>

…
```

Figure 76: imeline XSD being used for the Timeline “clinician” and tasks “See Patient 1” and “See Patient 2”.

Step 4: Using BPMNdm timeline in BPMNdm Visualiser:

Figure 77 shows the BPMN model in Figure 76 being executed in the BPMNdm visualiser, showing the two `timeTasks` being displayed at the time intervals each task occurs.

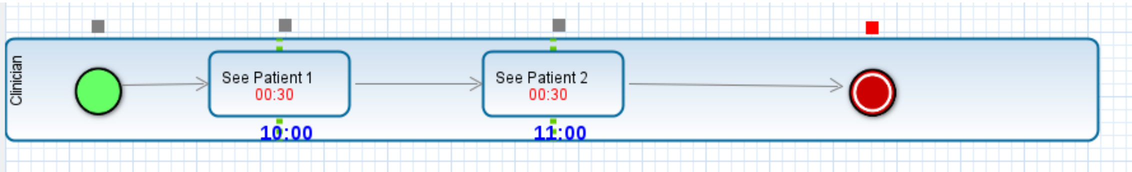

Figure 77: BPMN model displayed in the BPMNdm visualiser, the timeline and time tasks.

6.3.3 Scope and Workflow data

Step 1: Description of scope and workflow data

Russel et al. define workflow data as: "*Data elements are supported while are accessible to all components in each and every case of the workflow and are within the control of the workflow system*" and define Scope data as "*Data elements [that] can be defined which are accessible by a subset of the tasks in a case*" [118].

Each Scope and Workflow data item contains two key attributes, the `owner` swimlane the data item belongs to (or *none* if the item is a global resource that is shared by all participants) and a `isPublic` attribute, which indicate if the item is publicly accessible to all other participants. These attributes are detailed in the Scoped Data XSD in Figure 78.

Step 2: Scoped Data XSD from Chapter 3:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">


       <group name="scopedData">
          <sequence>
             <element name="owner" type="string" minOccurs="1" maxOccurs="1" />
             <element name="isPublic" type="boolean" minOccurs="1" maxOccurs="1" />
          </sequence>
       </group>


</schema>
```

Figure 78: XML schema definition (XSD) for Scoped Data object from Chapter 3

The `scopedData` definition is then imported and used in the BPMN model where applicable:

Step 3: BPMN XML model showing XSD being used:

```
<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:java="http://www.java.com/javaTypes"
xmlns:tns="http://www.jboss.org/drools" xmlns:sc="http://www.ncl.ac.ukcom/myns"
xmlns="http://www.jboss.org/drools"
xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL BPMN20.xsd
http://www.jboss.org/drools drools.xsd http://www.bpsim.org/schemas/1.0 bpsim.xsd"
id="Definition" exporter="org.eclipse.bpmn2.modeler.core" exporterVersion="1.1.2.Final"
expressionLanguage="http://www.mvel.org/2.0" targetNamespace="http://www.jboss.org/drools"
typeLanguage="http://www.java.com/javaTypes">
  <bpmn2:import importType="http://www.w3.org/2001/XMLSchema" location="src/scoped.xsd"
namespace="http://www.ncl.ac.ukcom/myns" />


…


<bpmn2:dataObject id="Patient_outcomes" name="Patient_outcomes"
itemSubjectRef="ItemDefinition_1">
    <bpmn2:extensionElements>
               <sc:owner>Clinician</sc:owner>
               <sc:isPublic>false</sc:isPublic>
    </bpmn2:extensionElements>
</bpmn2:dataObject>


…
```

Figure 79: Scoped XSD being used in the data object "Patient Outcomes"

The example presented in Figure 79 represents a non-public document named “Patient Outcomes” that belongs to a Clinician. A Visual BPMN representation of this design pattern is provided in Figure 80.

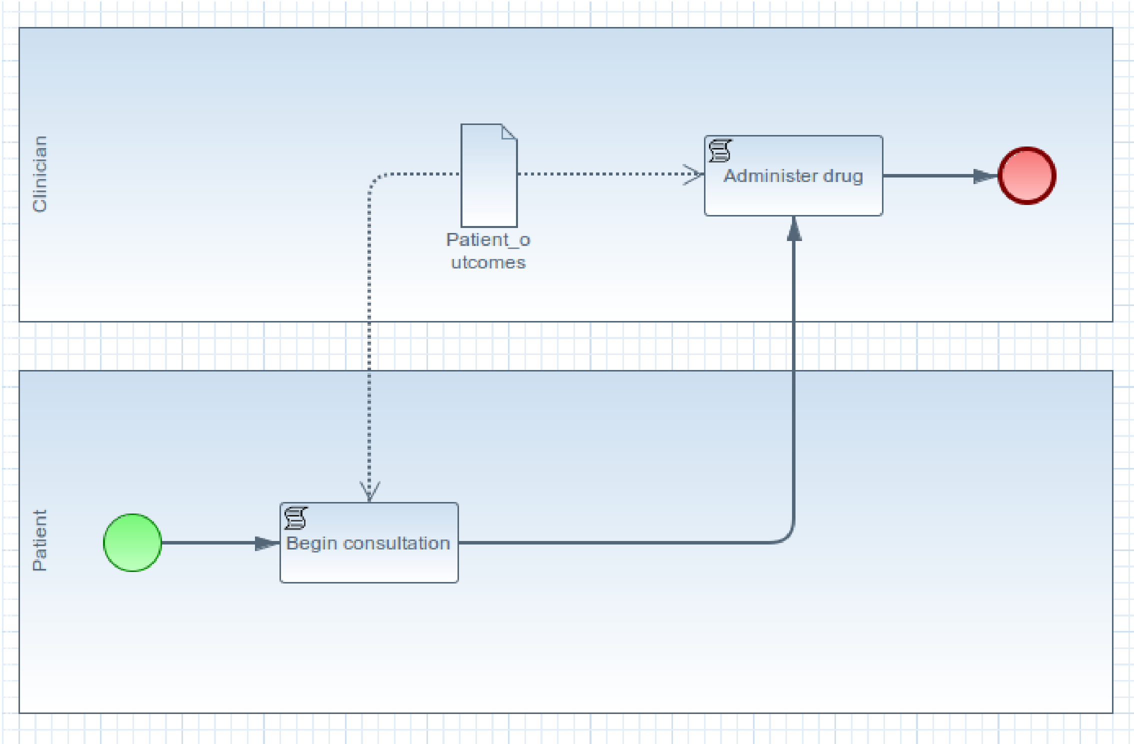

Figure 80: Simplified BPMN model of "Patient Outcomes" document that belongs to the user "Clinician"

Step 4: Using BPMNdm scoped data:

To determine if data objects represent workflow or scoped data, the implementation for jBPM uses two approaches: by infering directly the data's owner by the extended BPMNdm formalism as outlined in Figure 78 and Figure 79 and inferred indirectly by the coordinates of the data object in relation to swimlanes and outgoing/incoming associations with neighbouring tasks. When the jBPM session is started, the BPMN XML file is parsed using the W3C DOM (Document Object Model) Java package [162]. The class searches for the elements *owner* and *public* for each data object. When present, these values are parsed and passed into the corresponding `DrawNode` which represents the on-screen Document element in the BPMNdm visualiser. If owner is set to `_global`, BPMNdm assumes that the document object is *workflow* data rather than *scoped* data. An alternative strategy would be for BPMNdm to infer if a document is global by evaluating the position and size of the document and if it is bound by any swimlanes by looking at each swimales (x,y) coordinates. This was not implemented due to time constraints.

The second method of indirectly inferring the *owner* of a data object uses the coordinates, height and width of the data object to determine if the object is contained within a swimlane, otherwise the *owner* is set to `_global` using the same conventions as above.

During the runtime execution of the model, for each current task in the sequence flow, the code uses the jBPM workflow engine to check for outgoing data flows to data objects. This is inferred as one or more people in the scenario using a document as part of the current activity. Using the decision table described in Table 15, the system logs the contextual information of the activity using the data object, along with which actor the data object belongs to, where applicable. We call this *automatic data referencing.*

Data usage observation case	**Actor is owner?**	**Data is global?**	**Resulting observation**
Case 1	Yes	Yes	Actor is using to their own document, which can be shared with others.
Case 2	Yes	No	Actor is using their own document, which can't be shared with others.
Case 3	No	Yes	Actor is referring to someone else's document, which can be shared.
Case 4	No	No	Erroneous interaction with document, prompt modeller to check model/Actor wishes to use document they don't have access to.

Table 15: Decision table for handling design patterns for Data objects

In order to reduce the number of visual data flows for Workflow data items, these items can be manually specified by the modeller in the desired tasks using the `refDoc(documentName);` method which is embedded in a BPMN script task. A BPMN script task allows for a BPMN task to embed executable code, which is run in the jBPM workflow engine. We call this form of data referencing *manual data referencing.*

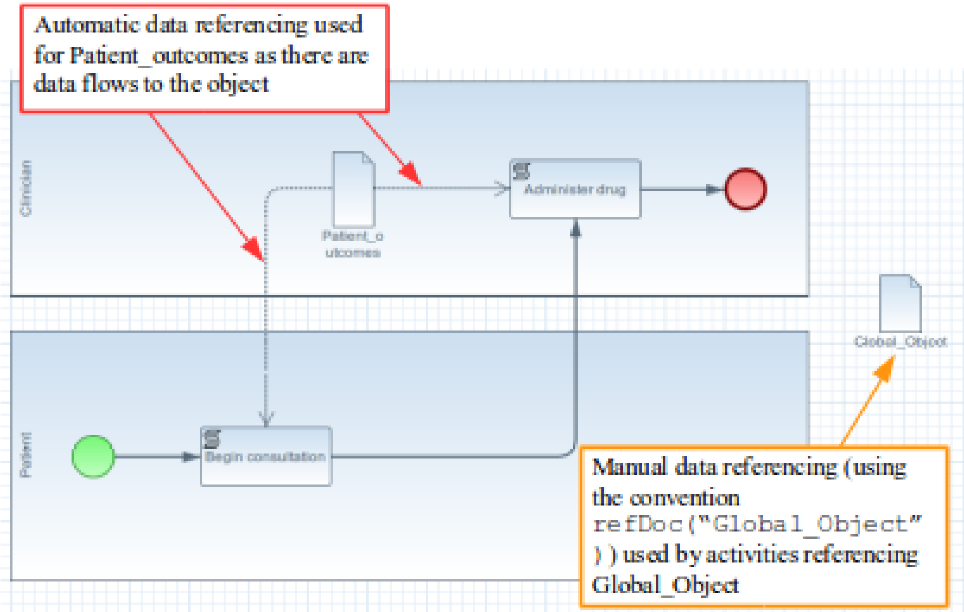

Figure 81: Annotated BPMN diagram showing the semantics of automatic and manual data referencing

The conventions used for both of these methods of data referencing are visualised in Figure 81.

As the scoped data BPMdm extension does not require any additional graphical notations, no additional changes were made to the BPMNdm visualiser. However, using the decision table in Table 15 and the workflow model in Figure 81, the BPMNdm Visualiser was extended to generate tool suggestions for the modeller based on the design of the workflow model. These suggestions are intended to form the basis of design requirements for a shared decision making tool to support the scenario being modelled. This is shown in Figure 82. Each requirement is generated in turn due to the design of the workflow model based on inferring the incoming and outgoing sequence flows, if the tasks are associated with a swimlane and if documents are referenced using the conventions in Table 15 and Figure 81. The suggestions are summarised as:

1. Data referencing, Patient referencing document belong to clinician.

2. Complex document, may need high resolution screen in order to fit all information on thw screen without scrolling. This is due to the patient outcomes document having a complex variable set to true. This is triggered by the data referencing activity above.

3. Task share between clinician and patient. This is inferred as the sequence flow traverses from the patient to the clinician. Also, both actors are referring to the same document.

4. Personal data reference, where the clinician uses a document they own (patient outcomes).

5. Complex document, may need high resolution screen. This is due to the patient outcomes document having a complex variable set to true. This is triggered for a second time by the data referencing activity above.

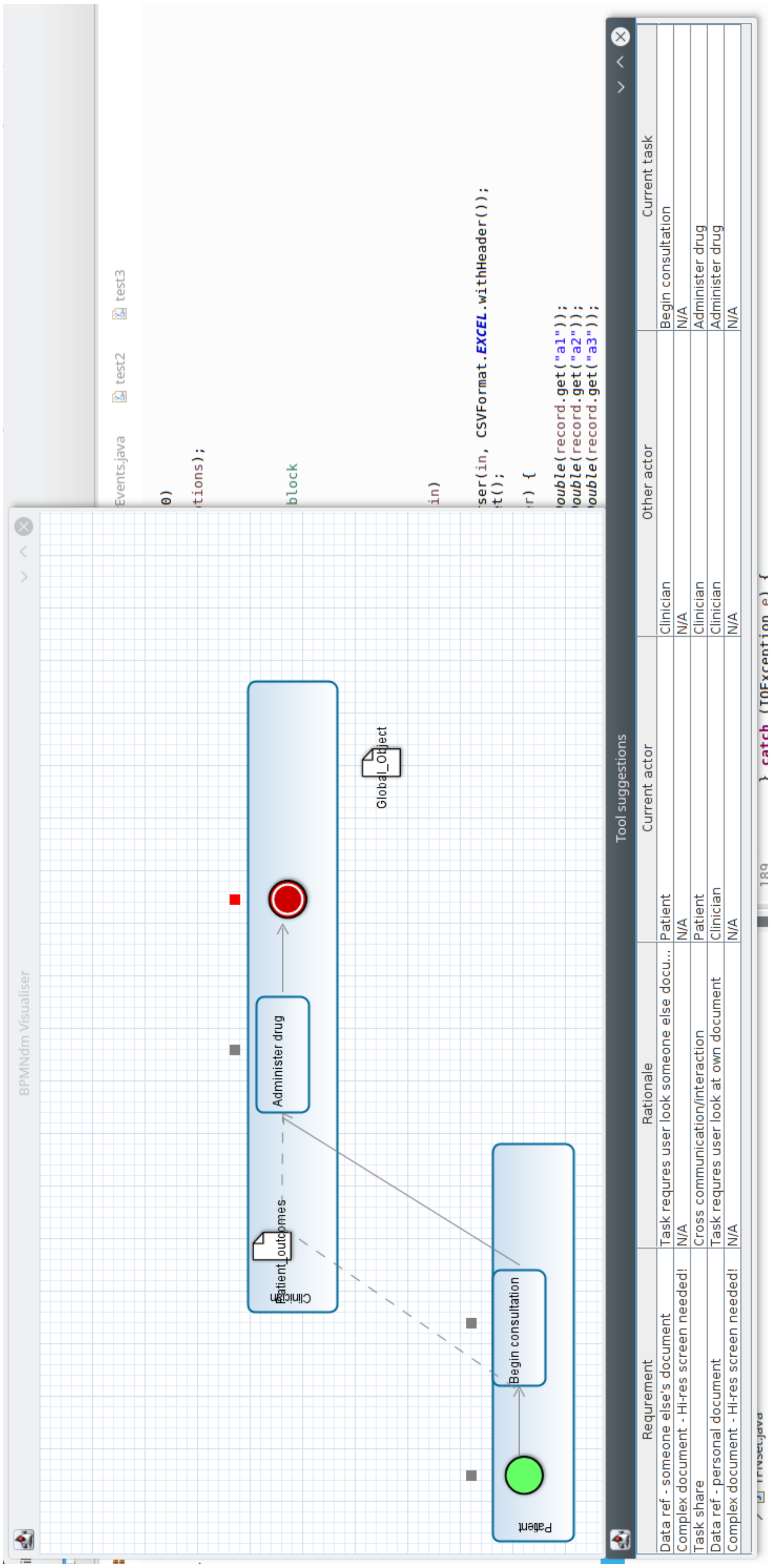

Figure 82: BPMNdm Visualiser and Tool Suggestions window

6.3.4 Weighted paths

Step 1: Description of weighted path

A weighted path is a visual representation of a weighted value assigned to one or more sequence flows in the BPMNdm model. Weighted paths are provided to show preferred sequence flows in the workflow model, for example if a certain sequence flow represented the preferred pathway for treatment. The XSD for a weighted path contains a single attribute that represents the weight of the specified sequence flow. This is shown in Figure 83.

Step 2: Weighted path XSD from Chapter 3:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:myns="http://www.ncl.ac.uk/myns"
elementFormDefault="qualified" targetNamespace="http://www.ncl.ac.uk/myns">


<group name="weightedPath">
        <sequence>
        <element name="weight" type="double" minOccurs="1" maxOccurs="1" />
        </sequence>
</group>


</schema>
```

Figure 83: XML schema definition (XSD) for Weighted Paths from Chapter 3

To illustrate the usage of weighted paths, the following sample BPMN model was constructed, describing a fictitious scnario where a clinician decides the type of treatment for cancer; either chemotherapy or radiotherapy as shown in Figure 84.

The process starts with the clinician evaluating the treatment options, followed by them choosing an option. The clinician will then administer the desired option and the process ends.

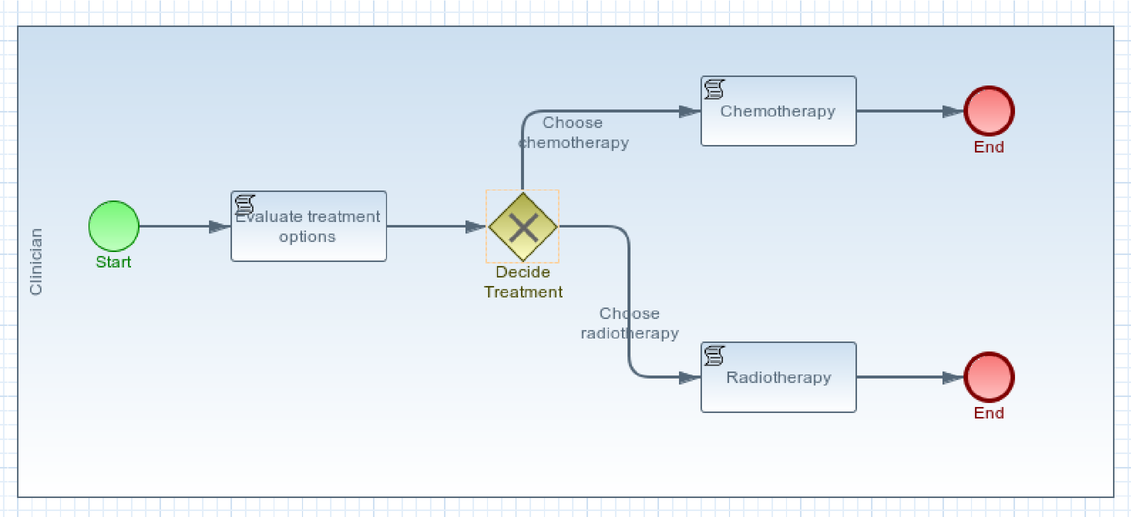

Figure 84: BPMN model of a clinician making a choice for treatment (chemotherapy or radiotherapy)

For this scenario, it is assumed that the existing treatment protocols specify that radiotherapy is preferred over chemotherapy, as radiotherapy is generally more efficacious. The modeller infers this as a preference score of **0.6 for radiotherapy** and **0.4 for chemotherapy**.

Step 3: BPMN XML model showing XSD being used:

The XSD in Figure 83 is then used in the BPMN model for the elements that represent the outgoing sequence flows for each treatment from the gateway "Decide Treatment" in Figure 84. The XML code in Figure 84 shows the usage of the weighted path XSD, with the two weights being specified for each outgoing sequence flow associated with radiotherapy and chemotherapy.

```
<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:java="http://www.java.com/javaTypes"
xmlns:tns="http://www.jboss.org/drools" xmlns:wp="http://www.ncl.ac.ukcom/myns"
xmlns="http://www.jboss.org/drools"
xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL BPMN20.xsd
http://www.jboss.org/drools drools.xsd http://www.bpsim.org/schemas/1.0 bpsim.xsd"
id="Definition" exporter="org.eclipse.bpmn2.modeler.core" exporterVersion="1.1.2.Final"
expressionLanguage="http://www.mvel.org/2.0" targetNamespace="http://www.jboss.org/drools"
typeLanguage="http://www.java.com/javaTypes">
  <bpmn2:import importType="http://www.w3.org/2001/XMLSchema" location="src/weightedPath.xsd"
namespace="http://www.ncl.ac.ukcom/myns" />


…


<bpmn2:sequenceFlow id="SequenceFlow_4" tns:priority="1" name="Choose chemotherapy"
sourceRef="ExclusiveGateway_1" targetRef="ScriptTask_4">
      <bpmn2:extensionElements>
        <wp:weight>0.4</wp:weight>
      </bpmn2:extensionElements>
…
</bpmn2:sequenceFlow>


…
 <bpmn2:sequenceFlow id="SequenceFlow_5" tns:priority="1" name="Choose radiotherapy"
sourceRef="ExclusiveGateway_1" targetRef="ScriptTask_5">
      <bpmn2:extensionElements>
        <wp:weight>0.6</wp:weight>
      </bpmn2:extensionElements>
…
</bpmn2:sequenceFlow>


...
```

Figure 85: Weighted Paths XSD being used for the paths "Choose chemotherapy" and "Choose radiotherapy".

The constraints used to determine the outgoing sequence flow of the gateway can be determined at design-time or at runtime, depending on the requirements of the model. If the constraints are determined at runtime, the modeller can use the method `actor.decide(currentNode);` which is a method of the `Actor` class, for which an instance is bound to the lane *Clinician*. We discuss the development of the `Actor` class in Section 6.4. The implementation of `actor.decide(currentNode);` is deferred for the modeller to implement, though a sample implantation for this thesis is detailed in Chapter 6, though the `currentNode` parameter includes a list of outgoing sequence flows and their corresponding weights. Alternatively, if the constraints are defined at design-time, the modeller can use standard jBPM practices of defining gateway constraints, such as using Java, Rule or Mvel syntaxes [159]. In the example provided in Figure 84, chemotherapy is arbiteraly chosen.

Regardless of if the constraints are generated at runtime or at design-time, in cases where the chosen sequence flow is not the highest weighted path, the BPMNdm system will warn the user of this.

Step 4: Using BPMNdm weighted path in BPMNdm Visualiser:

When the model is run, the BPMNdm visualiser displays the model, the actual sequence flow and for each weighted path, the actual weight is printed next to the corresponding sequence flow and the visual thickness of the sequence flow is adjusted proportionally to the weight as shown in Figure 86.

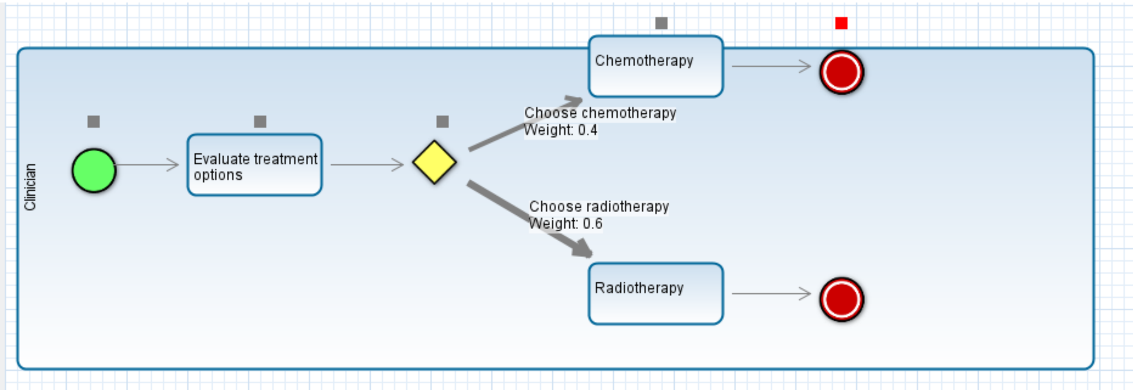

Figure 86: BPMN model displayed in the BPMNdm visualiser, showing the weighted paths for each treatment option, as well as the actual sequence flow of the model.

6.4 Person-to-person interaction rules implementation

Chapter 4 described the development of three Drools rules that encapsulate the person-to-person interactions identified in the ethnographic data. The three rules were defined in Section 4.5 as:

- **Interjection:** where a person adds more information (which could be irrelevant depending on the situation and type of information being shared) to the current item being discussed by another person.
- **Intervention:** where a senior person (such as a chair person, or other person) stops the current activity for a pre-defined reason (such as the current activity taking too long).

- **Question:** where a person asks the current person speaking to clarify or further explain the current item being discussed.

Drools rules can be used in BPMN workflow models and executed using the jBPM workflow engine. This section begins with a description of how to load Drools rules into the jBPM workflow engine, followed by an overview of extra Java classes developed to encapsulate the activities performed by decision makers, in the form of an `Actor` class representing scenario participants and a `Subject` class encapsulating the subjects actors discuss during a scenario.

6.4.1 Pre-requites and assumptions

For each rule developed, it is assumed that the interaction represented will be carried out between two people only, this was done to simplify the resulting syntax. When modelling more complex interactions, each rule and associated BPMN workflow pattern and used as a re-usable sub-process called a "*Call Activity" [31]*. The sub process is a secondary BPMN workflow that is encapsulated by the original BPMN model. This allows the modeller to re-use the workflow specific sequences of tasks, gateways and objects that constitutes a workflow pattern and be run as often as needed to simulate interactions between multiple people with the use of arbitrary cycles when necessary. A second advantage of this approach is that it solves issues of concurrency as the jBPM engine can be set to pause the execution of the calling process until the sub-process is completed.

To use each rule, the jBPM Project that will run the BPMN model and Drools rules requires the following Java code:

1. Load in all BPMN and Drools files for the model

A common way [122] is to write a main class that uses both BPMN files and Drools rules in a workflow model is to extend the example provided in Figure 69:

```
public class Main {

private static final long serialVersionUID = 1L;
public static KnowledgeBase kbase;

public static void main(String[] args) throws Exception {


kbase = readKnowledgeBase();
StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();

DrawEventListener del = new DrawEventListener("test.bpmn");
ksession.addEventListener(del);

WorkflowProcessInstance test = (WorkflowProcessInstance)
ksession.startProcess("com.sample.bpmn");
ksession.fireAllRules();

}

private static StatefulKnowledgeSession restoreSession(
      StatefulKnowledgeSession ksession, boolean b) {
      return ksession;
}


private static KnowledgeBase readKnowledgeBase() throws Exception {
KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();
      kbuilder.add(ResourceFactory.newClassPathResource("test.bpmn"),
ResourceType.BPMN2);
      kbuilder.add(ResourceFactory.newClassPathResource("RuleFile.drl"),
ResourceType.DRL);
            return kbuilder.newKnowledgeBase();
}

}
```

Figure 87: UML diagram for Main method of a jBPMN project using BPMN and Drools files

The `kbase` variable of type `KnowledgeBase` shown Figure 87 represents the entire model that is being executed by the jBPMN runtime. The `main()` method calls the private `readKnowledgeBase()` method that inserts one or more BPMN models and Drools rules as specified by the modeller into the *`kbase`* object. This is then returned to the main method where any non BPMN or Drools specific elements are added, such as event listeners or other Java objects as defined by the modeller. The model is then run. The method `fireAllRules()` ensures that all rules are executed.

2. Provide an `Actor` class to represent participants in the modelled scenario.

The `Actor` class is a standard Java class that represents the people participating in the scenario. The methods for the `Actor` class are defined based on the requirements of each rule defined in sequence.

3. Provide a `Subject` class

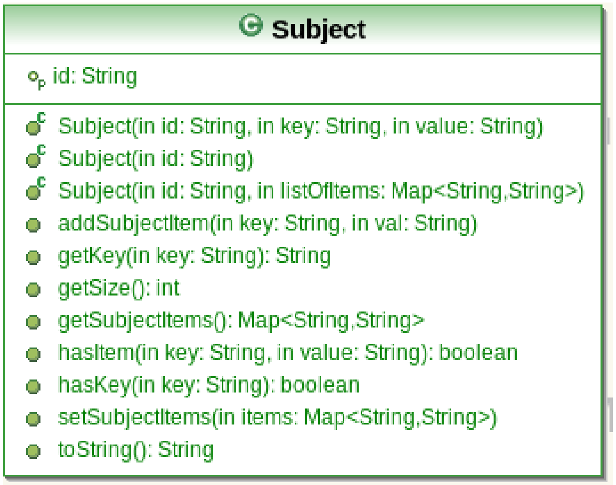

Figure 88: UML class diagram for `Subject` class

A `Subject` class represents each subject being discussed in each scenario. Each `Subject` instance has an identification number and a `Map` storing `String` keys and values, representing individual pieces of information about the subject. A UML class diagram for this class is provided in Figure 88.

The rest of this section is structured as follows:

For each rule developed in Chapter 4, the flowing steps are used to show how the rule can be used in a BPMN workflow model:

Step 1: Provide the Drools rule developed in Chapter 4.

Step 2: Provide a discussion of additional code necessary to make the rule work in a sample workflow model.

For this step, additional extensions to the Actor, Subject and Main class shown in Figure 87 may be necessary. This step will explain what additions and changes are necessary. A BPMN workflow model will also be illustrated that is used with the rule.

Step 3: Show the rule and workflow model being executed.

All of the rules generate text output witch is displayed on the jBPM console.

6.4.2 Interjection theme

Step 1: Rule file from Chapter 4:

The Drools rule in Figure 89 is taken from Chapter 3 and shows the necessary classes as well as the logic of the rule.

```
package uk.ac.ncl.cs

import uk.ac.ncl.cs.Subject
import uk.ac.ncl.cs.Actor


rule "Interjection"
dialect "java"
ruleflow-group "SubjectInterjection"

    when
        //conditions
        sub : Subject() //Subject in question
        a1 : Actor( Turn, Paused, Activity == "Talking" ) //Speaker
        a2 : Actor( a2.id != a1.id, !Turn, hasAdditionalInfomation(sub) )
//Other person

    then
        //actions
        System.out.println(a2.getId() + " has additional infomation, will
interject.");
        kcontext.getKnowledgeRuntime().setGlobal("willInterject", true);
end
```

Figure 89: Drools rule for Interjection theme

The syntax of the Drools rule in Figure 89 specifies that when the model instance has a given *Subject* (`Sub`) that is being discussed, an *Actor* (`a1`) who has the current turn, has paused and their current activity is "Talking", and a second *Actor* (`a2`) who is not `a1` and it is not currently their turn, has additional information on the subject being discussed. If these conditions evaluate to true, this will set the global variable "`willInterject`" to `true`, signifying that `a2` will interject.

Step 2: Necessary code changes and sample BPMN workflow model:

In order to support the Intervention rule, the Actor class requires the following additional methods:

1) **`boolean hasAdditionalInfomation(Subject other)`**

This method cross references the subjects being discussed by the other `Actor`, and returns `true` if this actor has additional information on the given subject, or `false` otherwise. For a scenario, each actor will have a list of facts associated with each subject. The method will return `false` if one or more of these facts has not been presented by another actor during the scnario.

2) **boolean `isTurn()` and setter `setTurn()`**

   This method returns true if it is the actor's current turn. The corresponding setter method `setTurn()` is called in BPMN script tasks that are associated with the actor.

3) **String `getActivity()` and setter `setActivity(String activity)`**

   The `getActivity()` method returns the current activity the actor is doing. This is set in the corresponding BPMN script task by calling the corresponding `setActivity("Talking")` method.

4) **boolean `isPaused()` and corresponding setter `setPaused(boolean value)`**

   The `isPaused()` method indicates if the actor has paused or not. This is semantically equivalent of a person pausing, while it is still their turn. The setter method is called in the corresponding BPMN script task.

The following example BPMN model shows the design pattern and necessary Java method calls to simulate an interjection Firstly, the BPMN workflow model is presented showing a person discussing an item with another person listening. Secondly, the implementation code necessary to trigger an interjection when the second person has additional information is shown. Thirdly, the code necessary to represent the information stored by each person is shown which is necessary to trigger the interjection event..

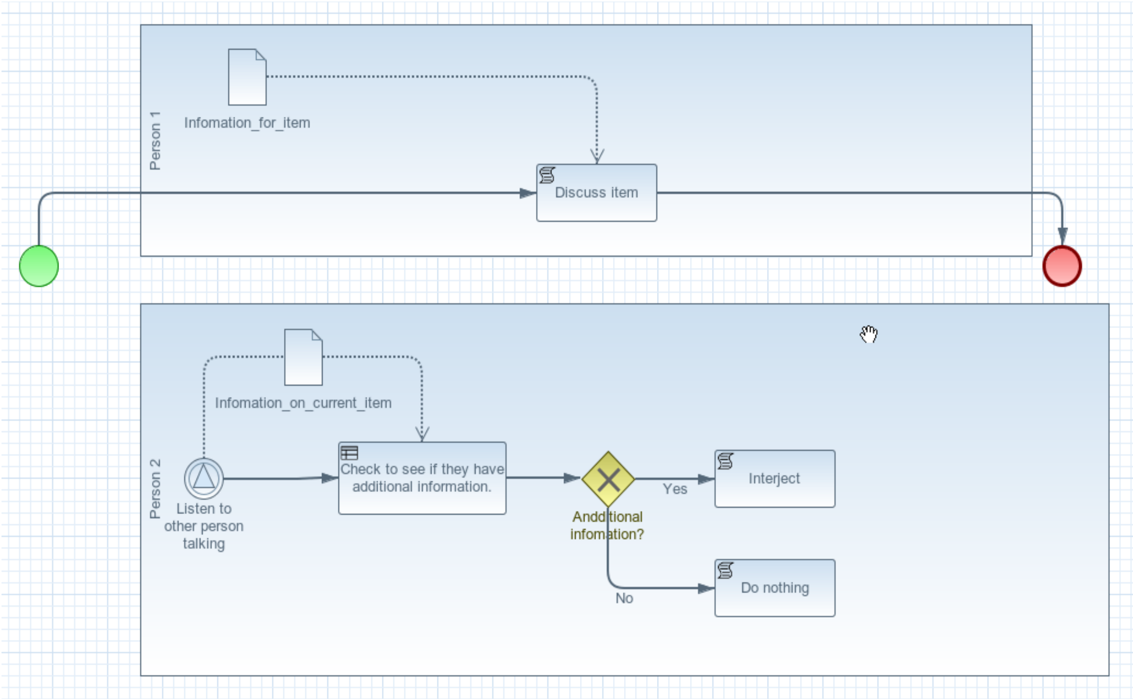

Figure 90: BPMN workflow model showing the design pattern for encapsulating interjections

The workflow model in Figure 90 shows two swimlanes corresponding to two Actors (Person 1 and Person 2). Person 1 performs two tasks where they will discuss two items, which are separate Subject instances. Person 2 listens to the item being discussed, as signified by the intermediate catch event. Next the person performs the business rule task "Check to see if they have additional information", which invokes the Interjection rule detailed in Figure 89. The exclusive gateway "Additional information?" checks the boolean value `willInterject`, and will perform an interjection, providing the additional information they have on the subject. The sequence flow of Person 2 is ad-hoc as there are no end events associated with their tasks. The rationale behind this is that the interjection does not end this process, rather adds to the information being shared by Person 1.

The following code samples detail the Java code necessary for each Script task to invoke the Interjection workflow pattern:

```
//Discuss first item Script Task
person1.setTurn();
person1.setActivity("Talking");
currentSubject = person1.getSubjectById(currentSubjectId);
kcontext.getKnowledgeRuntime().setGlobal("currentSubject",
currentSubject);
System.out.println("Actor " + person1 + " is discussing "  +
currentSubject);
actorList.updateSubjectForOtherActors(person1,currentSubject);
person1.setPaused(true);
kcontext.getKnowledgeRuntime().insert(person1);
kcontext.getKnowledgeRuntime().insert(person2);
kcontext.getKnowledgeRuntime().insert(currentSubject);
kcontext.getKnowledgeRuntime().signalEvent("Listen_int",null,kcontext.getP
rocessInstance().getId());
```

Figure 91: Code listing for "Discuss first item" script task.

The code listing in Figure 91 provides the necessary method calls to trigger the interjection process of Person 2. Firstly, the modeller sets the current turn to Person 1, which is the instance of Actor associated with the Person 1 BPMN lane. The global variable for the model `currentSubject` is then set to the first item Person 1 has on their list of subjects. For each actor associated with this model, their individual list of subjects are updated to include the new information being shared by Person 1, symbolised by the call `actorList.updateSubjectForOtherActors()` method. Following this, the two actors and the current subject being discussed are inserted into the jBPM knowledge runtime in order for them to be associated with the Interjection drools rule. The script finally signals the “`Listen_int`” method that then triggers the “Listen to other person talking” intermediate catch event in the lane of “Person 2”.

The Business Rule task “Check to see if they have additional information” is then executed which runs the Drools rule in Figure 103. If the “`willInterject`” value is set to true by the rule, Person 2 will perform an interjection, which is simulated by the following code:

```
//Interject script task
System.out.println("Interjecting");
```

```
actorList.updateSubjectForOtherActors(person2,person2.getSubjectById(curre
ntSubjectId));
kcontext.getKnowledgeRuntime().setGlobal("willInterject", false);
```

Figure 92: Code listing for Interject script task.

Interjection simulated by the code in Figure 92 firstly prints the current activity to the console then updates the other Actors used in the model of the additional information they have. Finally, "`willInterject`" is set to `false`.

Example use of Drools rule and BPMN workflow model:

In this example, three individual subjects and two actors are created. Actor 1 (Ben) is associated with the first subject, which is regarding a robbery committed by "Dave" and the third subject: Shoplifting committed by "Jenny".

Actor 2 (Joe) is also associated with the same type of subject, but has the additional information of when the crime was committed. The code necessary to create the subjects and actors is provided in Figure 93:

```
//In method public static void main(String[] args)
TurnInitiater ti = new TurnInitiater();

ActorList al = new ActorList();

Map<String, Object> params = new HashMap<String, Object>();

Subject sub = new Subject("01", "Dave", "Robbery");

Subject sub2 = new Subject("01", "Dave", "Robbery");
sub2.addSubjectItem("Date", "15-10-15");
Subject sub3 = new Subject("02", "Jenny", "Shoplifting");

Actor person1 = new Actor("01", "Ben", "Participant", ti);
Actor person2 = new Actor("02", "Joe", "Participant", ti);

al.add(person1);
al.add(person2);

person1.addNewSubject(sub);
person2.addNewSubject(sub2);
person2.addNewSubject(sub3);
```

Figure 93: Code sample of the creation of two actors and three subject objects.

The actors, subjects and necessary variables are inserted into the model then the model is run:

```
kbase = readKnowledgeBase();
            StatefulKnowledgeSession ksession =
kbase.newStatefulKnowledgeSession();

ksession.setGlobal("willInterject", false);
ksession.setGlobal("person1", person1);
ksession.setGlobal("person2", person2);
ksession.setGlobal("actorList", al);
ksession.setGlobal("currentSubject", null);
```

```
ksession.setGlobal("currentSubjectId", "01");
ksession.addEventListener(new SimpleProcessEventListener());

WorkflowProcessInstance test = (WorkflowProcessInstance) ksession
                               .startProcess("com.sample.bpmn", params);
```

Figure 94: Code sample showing global variables being set for process and the process being run

The variables described in Figures 91,92,93 and 94 are inserted into the process as global variables, as they will be accessed by the BPMN sub-process shown in Figure 90 that uses the rule.

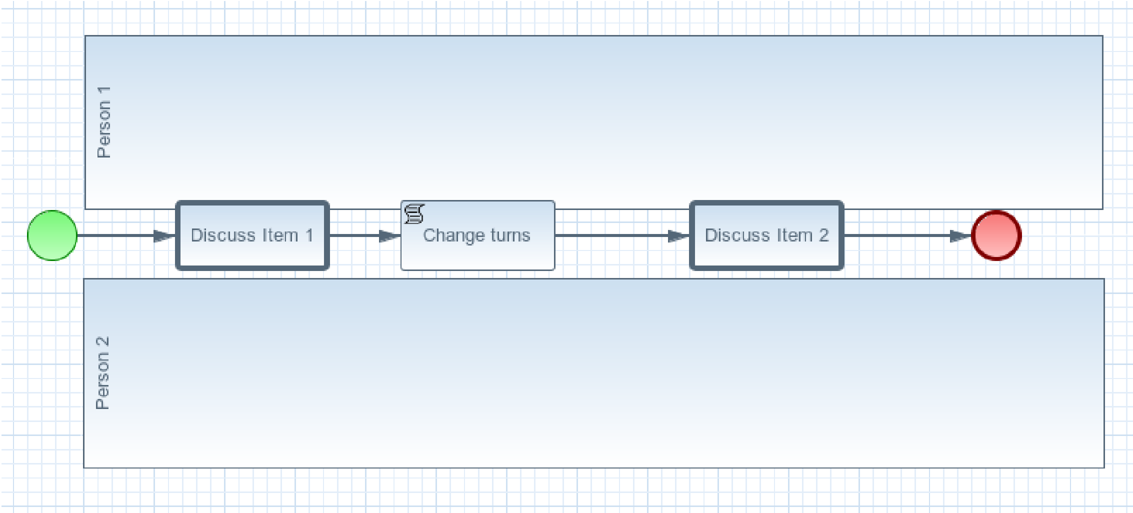

Figure 95: BPMN super process showing the Interjection call activities "Discuss Item 1" and "Discuss Item 2"

The interjection workflow model is used in a simple super-process that comprises of two interjection call activities with an intermediate script task that changes the Actors over so each one simulates taking a turn in sharing information. This is shown in Figure 95.

Step 3: Running the BPMN workflow model and Drools rule:

When the model is run, during the “Discuss Item 1” task, Actor 2 performs an interjection since Actor 1 had paused and Actor 2 has additional information that was not shared. This is shown in Figure 96.

```
Actor ID: 01, Name: Ben is discussing 01
02 has additional information, will interject.
Interjecting
```

```
End
```

Figure 96: Terminal output showing an interjecting being triggered.

When "Discuss Item 2" task is run, Actor 2 is now discussing the second subject, which Actor 1 does not have any additional information on, therefore no interjection is triggered. This is shown in Figure 97.

```
Actor 01 is unsetting turn, it is no one's turn now
Setting initial turn for 02
Actor ID: 02, Name: Joe is discussing 02
```

Figure 97: Terminal output showing no interjection being performed

6.4.3 Intervention theme

Step 1: Rule file from Chapter 4:

The following figure shows the final Drools rule for the intervention theme developed in Chapter 4:

```
package uk.ac.ncl.cs
```

```
import org.drools.runtime.StatefulKnowledgeSession;
import uk.ac.ncl.cs.Actor

rule "Intervention"
dialect "java"
ruleflow-group "QIntervention"
    when

     a1: Actor( Role == "Chair", !Turn )
     a2: Actor( Role != "Chair", Turn, a2.TimeTaken > 5)
    then
        //actions
        a1.setTurn();
        System.err.println("The participant has spent too long talking");
        kcontext.getKnowledgeRuntime().setGlobal("chairIntervention",
true);

end
```

Figure 98: Drools rule for Intervention theme from Chapter 4

Step 2: Necessary code changes and sample BPMN workflow model:

The Drools rule for the Intervention in Figure 98. The conditions are that when an Actor (`a1`) is the Chair and it is not their turn, and if a second actor who is not chair, and it is their turn, and they have taken over 5 minutes to perform their current activity, the Chair `a1` takes over by calling the `setTurn()` method that yields control from a2. The global variable "`chairIntervention`" is set to to true.

The BPMN model that is used to define the control flow for dealing with the intervention is provided in Figure 99.

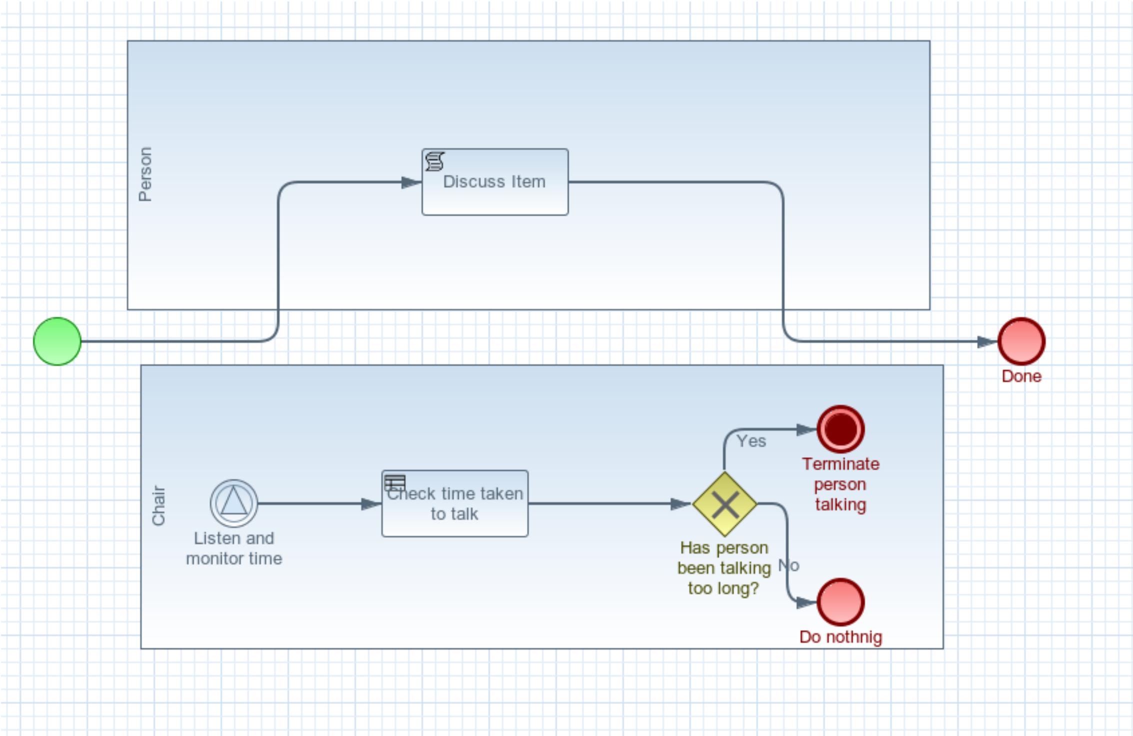

Figure 99: BPMN Workflow model showing design pattern for the design pattern for encapsulating interventions

The workflow model for interventions includes a “Person” performing the script task “Discuss item”. The other person in this model is a meeting “Chair”. When the “Discuss item” task is executed, this triggers the “Listen and monitor time” event. This calls the “Check time taken to talk” rule task which is associated with the Drools rule in Figure 98. If the other person has taken too long to talk, then the Chair will terminate the current process, or do nothing otherwise.

The code necessary for the “Discuss item” task is provided in Figure 100:

```
a2.setTurn();
a2.incrimentTimeTaken(60);
kcontext.getKnowledgeRuntime().insert(a1);
kcontext.getKnowledgeRuntime().insert(a2);
kcontext.getKnowledgeRuntime().signalEvent("Listen_int",null,kcontext.getP
rocessInstance().getId());
```

Figure 100: Code listing for "Discuss item" task

The first action is Actor 2 (associated to the “Person”) is set as the current turn. Next, the time taken to perform the activity is set to 60 seconds. Next, the two actors and the current subjecting being discussed are inserted into the jBPM knowledge runtime in order for them to be associated with the Interjection drools rule. The script finally signals the "`Listen_int`" method that then triggers the "Listen and monitor time" intermediate catch event in the lane of "Chair". This triggers the execution of the Intervention rule, which sets the variable "`chairIntervention`" to be set to true if the other person has been talking for more than 5 minutes, or false otherwise. The exclusive way evaluates the value of `chairIntervention` and terminates the current instance of the model if it is true.

Example use of Drools rule and BPMN workflow model:

Similar to the Interjection workflow model, two actors are defined before the model is run. In this case, a person is created who has the role as chair and the other as a "Participant":

```
//In method public static void main(String[] args)
TurnInitiater ti = new TurnInitiater();

ActorList al = new ActorList();

Map<String, Object> params = new HashMap<String, Object>();

Actor person1 = new Actor("01", "Ben", "Chair", ti);
Actor person2 = new Actor("02", "Joe", "Participant", ti);

al.add(person1);
al.add(person2);

kbase = readKnowledgeBase();
            StatefulKnowledgeSession ksession =
kbase.newStatefulKnowledgeSession();

ksession.setGlobal("chairIntervention", false);
ksession.setGlobal("person1", person1);
ksession.setGlobal("person2", person2);
ksession.setGlobal("actorList", al);
ksession.addEventListener(new SimpleProcessEventListener());

WorkflowProcessInstance test = (WorkflowProcessInstance) ksession
                        .startProcess("com.sample.bpmn", params);
```

Figure 101: Code sample of the creation of two actors and running the model.

Step 3: Running the BPMN workflow model and Drools rule:

The code listed in Figure 101 is similar to Figures 89 and 98 with the exception that no subjects are set for this example and the first Actor (person 1) is set as the meeting Chair. Due to the Person 2 taking 60 seconds to perform the activity show in in Figure 100, when the model is run, the model and code trigger the intervention and subsequent termination of the process, as shown in Figure 102.

```
Setting initial turn for 02
Chair 01 is forcing  Actor 02 to give up turn.
The participant has spent too long talking
End
```

Figure 102: Terminal output showing the intervention being performed

6.4.4 Question theme

Step 1: Rule file from Chapter 4:

The following figure shows the final Drools rule developed for the question theme developed in Chapter 4:

```
package uk.ac.ncl.cs

import uk.ac.ncl.cs.Subject
import uk.ac.ncl.cs.Actor


rule "Question"
dialect "java"

    when
        //conditions
        sub : Subject() //Subject in question
        a1 : Actor( Turn ) //Speaker
        a2 : Actor( id != a1.id, !hasSubjectById(sub.id), !Turn, canTalk )
//Other person

    then
        //actions
        System.out.println(a2.getId() + " does not have information,
firing question task.");
        a2.setToAskQuestion(true);
end
```

Figure 103: Drools rule for Question theme from Chapter 4

Step 2: Necessary code changes and sample BPMN workflow model:

The syntax of the Drools rule in Figure 103 specifies that when the model instance has a given *Subject* (`Sub`) that is being discussed, an *Actor* (`a1`) who has the current turn, and a second *Actor* (a2) who is not a1, does not have information on what is being discussed it is not their turn however at the moment of the rule firing, they are able to talk. If these conditions evaluate to `true`, this will set the field `toAskAQuestion` to `true` for `a2`, which will be evaluated at a later point in the workflow model.

In order to support the Intervention rule, the Actor class requires the following additional methods:

1. **`hasSubjectById(subjectId)`**

   This method returns true if the Actor has the subject by the given subject id. This is defined as "`UnderstandSubject`" in Chapter 4. Since a person's understanding of a given subject is subjective, this was simplified to cross referencing the subjects a user has against what the speaker has provided. Chapter 6 describes the encapsulation of subjectivity with Triangular fuzzy numbers, that could be used to define a `understandSubject` method.

2. **`setToAskQuestion(value)` setter and `needToAskQuestion()` getter**

   This method indicates if the simulated Actor wishes to asks a question and the condition will be evaluated at a future point when the workflow model determines how to handle the condition.

3. **`canTalk()` method**

   This method returns `true` if at the time of the rule firing, the selected actor is able to talk, which is dependant on the other actor's `isPaused()` method returning `true`.

The following example BPMN model shows the design pattern and necessary Java method calls to simulate a question being asked:

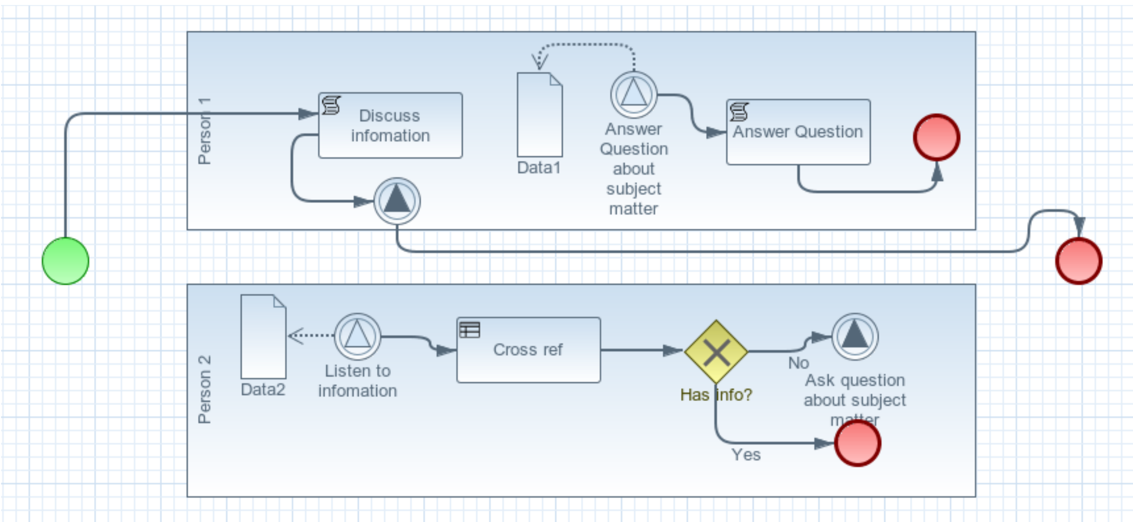

Figure 104: BPMN Workflow model showing design pattern for the design pattern for encapsulating questions

Figure 104 shows two swimlanes, Person 1 and Person 2. Person 1 is the one who is sharing information on the current Subject. This causes an intermediate throwing event to trigger the "Listen to information" catch event on the Person 2 swimlane. This triggers the "Cross ref" rule task which is associated with the Drools rule provided in Figure 103. Next, at the "Has Info?" exclusive gateway, if the actor method `needToAskQuestion()` method associated with Person 2 lane evaluates to true, this triggers the "Ask question about subject matter" intermediate throw event, or ends. If the event is thrown, it is caught by the "Answer question about subject matter" intermediate catch event on the "Person 1" lane, and the "Answer question" task is executed before the process ends.

Example use of Drools rule and BPMN workflow model:

Like previous examples, two Actor instances and one Subject instance is created. The subject is associated with one actor. This is shown in Figure 105:

```
//In method public static void main(String[] args)
TurnInitiater ti = new TurnInitiater();

ActorList al = new ActorList();

Map<String, Object> params = new HashMap<String, Object>();

Subject sub = new Subject("01", "Dave", "Robbery");

Actor person1 = new Actor("01", "Ben", "Participant", ti);
Actor person2 = new Actor("02", "Joe", "Participant", ti);
person1.addNewSubject(sub);

al.add(person1);
al.add(person2);

kbase = readKnowledgeBase();
            StatefulKnowledgeSession ksession =
kbase.newStatefulKnowledgeSession();

ksession.setGlobal("person1", person1);
ksession.setGlobal("person2", person2);
ksession.setGlobal("actorList", al);
```

```
ksession.setGlobal("currentSubjectId", "01");
ksession.addEventListener(new SimpleProcessEventListener());

WorkflowProcessInstance test = (WorkflowProcessInstance) ksession
                        .startProcess("com.sample.bpmn", params);
```

Figure 105: Code sample of the creation of two actors and running the to simulate a question model.

In this example, a question will be triggered by simulating a piece of information not being disclosed (in this case Subject 1 - "Dave – Robbery"). This is achieved in the "Discuss information" Script Task when the modeller purposely excludes the line "`actorList.updateSubjectForOtherActors(a1,currentSubject);`", which does not update the list of known subjects for the other actors in the model. The code necessary for the script task is provided in Figure 106:

```
a1.setTurn();
a1.setActivity("Talking");
currentSubject = a1.getSubjectById(currentSubjectId);
kcontext.getKnowledgeRuntime().setGlobal("currentSubject",
currentSubject);
System.out.println("Actor " + a1 + " is discussing "  + currentSubject);
//actorList.updateSubjectForOtherActors(a1,currentSubject);
kcontext.getKnowledgeRuntime().insert(a1);
kcontext.getKnowledgeRuntime().insert(a2);
kcontext.getKnowledgeRuntime().insert(currentSubject);
```

Figure 106: Code listing for "Discuss information" task.

Figure 107: UML class diagram of complete Actor class

The code listing in Figure 106 shows that the Actor a1 (Ben) is set for the current turn and their activity is set to “Talking”. The current subject the actor is discussing is assigned to the variable “`currentSubject`” and this is set as a global variable. The last three lines insert the two actors and current subject into the Drools runtime ready for the Drools rule to be executed. Figure 107 shows the complete UML diagram of the Actor class with all of the methods developed for this chapter

After the task “Discuss information” is triggered, the next node triggered is an intermediate throw event which is caught by the intermediate catch event on the “Person 2” swimlane (associated with the Actor “Joe”). Next, the “Cross ref” rule task executes the Drools rule in Figure 103. If the actor does not have the subject ID being discussed, this will set the field `needsToAskQuestion` to `true`. This causes the “Has info?” exclusive gateway to direct the sequence flow to the “Ask question about subject matter” intermediate event, which is caught on the Person 1 lane, stimulating the question being answered. The trace of this process and rule running is provided in Figure 108.

Step 3: Running the BPMN workflow model and Drools rule:

The output shown in Figure 108 shows the initial turn being set to “Ben”, with the actor discussing the first subject item. The second actor, “Joe” does not have the subject item, therefore fires a question.

```
Setting initial turn for 01
Actor ID: 01, Name: Ben is discussing 01
Checking...
02 does not have information, firing question task.
ID: 01, Name: Ben is giving answer to question
```

Figure 108: Code listing for running the process model simulating a question

6.5 Discussion

This chapter documents the implementation of the four BPMNdm formalism extensions defined in Chapter 3 and the three person-to-person interaction rules defined in Chapter 4.

For the BPMNdm formalism extension, the extension XSD is provided followed by an example of incorporating the XSD in a BPMN XML file as well as Java code specific to the jBPM 5 workflow engine that is necessary to run a BPMN model using the extension. This section also described the requirements of an Actor class which would store the tasks each simulated actor would undertake, as well as any documents associated with the instance of the Actor class. While each extension defined in Chapter 3 has been implemented, the need for a custom BPMN visualiser to visualise the extension in a BPMN workflow model is a drawback to using BPMN, as the formalism does not include provision for visualising extension elements. This requires the modeller to provide their own visualising code, or to use the generic visual components provided by the workflow engine. To overcome this issue with our extensions, we have designed and demonstrated the BPMNdm Visualiser, which visualises our extended BPMN elements, shows the real-time execution of the workflow model, indicates the previous sequence flow and the current node on the diagram.

A disadvantage of providing additional visualising code is the additional time and skills required for the modeller to provide additional code. The disadvantage of using the built in visual components is the loss of visual representation of the BPMN extension, thus undermining the principle of visual the visual affordance of the BPMN symbols in terms of producing visual workflow models for non-technical users.

We have outlined how to use the BPMNdm extensions in BPMN workflow models using the following steps:

Step 1: Description of each the BPMNdm extension.

Step 2: Display the BPMNdm extension XSD from Chapter 3.

Step 3: Show the BPMN XML source of a workflow model that will feature the given BPMNdm extension, referencing of the BPMNdm XSD from Step 2 and the extension being defined.

Step 4: Discuss any specific implementation details and show the BPMNdm model running in the BPMNdm Visualiser.

The rest of this chapter described the technical implementation of the three Drools rules defined in Chapter 4. For each rule, a sample workflow model was constructed and the necessary Java code was provided in order to provide the logical implementation to allow the model to be executed in jBPM 5. Further extensions to the Actor class were defined, such as methods for interacting with a Subject object, which is a code representation of the current item being discussed in a meeting scenario.

We have structured the specific steps necessary to use the Drools rules developed in the following way:

Step 1: Provide the Drools rule developed in Chapter 4.

Step 2: Provide a discussion of additional code necessary to make the rule work in a sample workflow model.

Step 3: Show the rule and workflow model being executed.

While the person-to-person interaction rules were developed based on the MAPPA scenario detailed in Chapter 4, the implementation of the rules are generic. The conditions for each rule are based on the evaluation of string rather than object parameters and can be further customised for a specific scenario. This design pattern is based on the principle conventions of Drools rules [121], where the logic of a rule is separated from the semantics of the workflow model, or the technical details of any code to run the model in the workflow engine. The example BPMN workflow models provided in this chapter represent two actors participating in the scenario, however it would be possible to simulate three or more actors participating in the scenario. To achieve this, the modeller would need to replace the one or both of the existing Actor class instances and run the model again. The advantage of this approach is that different Actors, with their own individual characteristics, can be used each time the model is run. A disadvantage of this is if the scenario is modelling simultaneous interaction between three or more actors, all of which are performing different tasks. In this case, a more complex BPMN model with additional swimlanes showing the individual tasks being performed would be necessary.

Chapter 7 - Conclusion

7.1 Overview of thesis objectives

Objective 1 of this thesis was to appraise a number of scenarios. This objective was satisfied with the description of the thrombolytic stroke treatment scenario in Chapter 3. This scenario was supplemented by three further scenarios; defining a security policy, MSc project selection and cardiovascular risk reduction. In Chapter 4, the MAPPA process was documented.

Objective 2 was to find and utilise a modelling notation that allowed for adaptations to support specific decision making activities that had been identified for Objective 1. This was fulfilled by extending the BPMN formalism to encapsulate decision-making activities by appraising thrombolytic stroke treatment from the perspective of the patient, security policy form the perspective of the policy makers, MSc project selection from the perspective of the student, and cardiovascular risk reduction from the perspective of the general practitioner. These are detailed in Chapter 3.

Objective 3 was to identify a suitable workflow engine to run BPMN models as well as provide support for custom made components and extensions. jBPM 5 was chosen to fulfil this objective as it supported a subset of the BPMN 2.0 formalism and allowed the modeller to provide Java based extensions to enhance the engines' functionality.

Objective 4 was to to devise a methodology for defining interactions between participants in a decision making scenario. This was fulfilled by the development of three Drools rules encapsulating person-to-person interactions as described in Chapter 4, which were developed by analysing the ethnographic data collected by observing MAPPA meetings.

Objective 5 was to adopt a methodology for conceptualising and encapsulating subjectivity. This fulfilled by adopting triangular fuzzy numbers (TFNs) as an approach for encapsulating subjective information. This is discussed in Chapter 5.

7.2 Results

In this thesis, we presented four primary outputs from the research and development undertaken. In Chapter 3, we presented the development of four extensions to the BPMN formalism, Scoped/Workflow data, Weighted Paths, Timelines and Cost lines. These BPMN extensions were defined by analysing four decision-making scenarios and the extended formalism was termed BPMN decision making, or BPMNdm. The first contribution of this theses was the creation of a methodology for transforming the requirements derived from the analysis into BPMN extensions by modifying the methodology by Stroppi et al. [42] that is formed of the following steps:

Step 1: Construct BPMN workflow models for decision making scenarios using the existing BPMN 2.0 formalism and identify desired decision making activities

Step 2: For each decision making activity, build a visual prototype of how the decision making activity should be visualised by using existing BPMN components and visualising new components where necessary.

Step 3: For each visual prototype, construct a UML class diagram, indicating the extension element and its relationship to the existing BPMN elements in the formalism. This visualisation is the metamodel for the extension.

Step 4: Implement the extension as a XSD file that can be used in BPMN workflow models.

The second contribution of this thesis is the development of a set of rules to encapsulate subjective interactions between human actors in a meeting environment. This is detailed in Chapter 4 which presented the development of person-to-person interaction rules intended to support the development of workflow models for modelling risk communication scenarios. The development of these rules was informed by identifying themes by using thematic content analysis on a series of meeting observation notes from eight meetings as part of the MAPPA case study. The themes generated were transformed into Drools rules using First Order Logic syntax as an intermediate step, as described by Kaiser et. Al [123]. Three rules were produced: Question, Interjection and Interventions. The rules were designed to be applicable to other decision making scenarios, therefore any components that required specific implementation relevant to the scenario being modelled, were intended to be delegated to the modeller to implement. The rules are intended to be used as part of structured design patterns for the larger workflow model. For example, each rule and surrounding workflow pattern is based on the participation between two actors. To simulate more than two actors participating, the surrounding work pattern is represented as a reusable process, and a high-level workflow model is delegated to symbolising multiple actors that allow the work pattern to represent the interaction of more than two actors.

This aggregation of the interaction rule and surrounding workflow pattern allows for the modeller to re-use the workflow pattern when needed, however it is dependent on the modeller completing the necessary implementation, such the definition of task items, any documents, conversation topics etc. The methodology developed for the rules was as follows:

Pre step: Perform thematic content analysis on the data to identify relevant themes.

Step 1: For each them, map the theme to an analogous BPMN element.

Step 2: For each them that does not map well to a BPMN element, show the thematic map of the theme identifying the key attributes that constitute the theme.

Step 3: Using Kaiser et. al's definitions, map the theme to **Events, Conditions** and **Actions**, expressing these mappings using FOL syntax.

Step 4: Using the FOL formula produced in Step 3, produce a Drools rule by mapping the Events and Conditions to the *when* section of the rule that defines the logic of the rule. Actions are mapped to the *then* block of the rule, which define the action the rule should perform. As we are using these rules as part of a BPMN workflow model, the actual actions are defined in the sequence flow of the workflow model, that is determined by the rule.

Chapter 5 presented the use of TFNs as a method of encapsulating subjectivity by aggregation a set of subjective responses to a scale as an interval of confidence. The third contribution of this theses is a worked example was provided using participant data collected as part of the wider research on the thrombolytic stroke treatment scenario described in Chapter 3. The example shows the aggregation of patient and family member preferences to Shared Decision making into a TFN. By varying the α value on the TFN, this is assumed to indicate the decision makers' confidence in their choices and by varying λ, indicating the decision makers' attitude towards risk [134].

Chapter 6 presented technical implementations for BPMNdm developed in Chapter 3 and the Drools rules developed in Chapter 4.

Each contribution in intended to be presented so that each methodology developed is repeatable by breaking down each process in to steps, for other system designers to adapt for their own requirements.

7.3 Future work

The principle limitation for each component of this thesis is the absence of user evaluations of the tools and methodologies developed. While the extensions to the BPMNdm formalism were defined by evaluating existing risk-communication scenarios and following a structured methodology for formalising and implementing each extension, a full evaluation with stakeholders using workflow models with the extensions was not undertaken. However, for each extension, it was intended to follow the design principles for visualising BPNM elements, such as using graphical notations and visual scales, which usability principles are well understood in the scientific community [163].

For the person-to-person interaction rules described in Chapter 4, the limitation is that while they are developed using the ethnographic data obtained from the MAPPA scenario, the components of each rule are designed to be generic aggregations of the activities identified in each MAPPA meeting, rather than being tailored specifically to these activities. In order to use each rule for a specific scenario, the modeller needs to specify additional details, such as the activities undertaken, the data each participant will use to make decisions and to finally construct a workflow model that replicates the decision making activity that uses the rules.

For the use of TFNs as discussed in Chapter 5 to encapsulate subjectivity, the choice of using triangular fuzzy numbers is controversial, specifically in the case of risk communication and decision making, how a set of responses to subject criteria are aggregated into a single TFN. Furthermore, given a set of subjective assessments made by decision makers, to produce a crisp number from the fuzzy set requires the modeller to make assumptions on the confidence and attitude to risk of the decision makers. In such circumstances, it is imperative that these assumptions are verified with the stakeholders to ensure that these assumptions are reasonable. Extending an existing decision making algorithm such as TOPSIS to incorporate TFNs and their representation of subjective data has been discussed by Yeh and Chang [134], however a primary drawback of TOPSIS is the assumption that decision makers perform pairwise comparisons of each alternative, which can be time consuming when there are multiple assessment criteria. Additionally, TOPSIS assumes that each criteria is weighted by the decision makers, which may not be done in the real-life scenario. This limits the usefulness of TOPSIS in these cases, as it is not analogous to the decision making activities performed by the decision makers.

The work described in this thesis draws research conducted in three broad subject areas; workflow modelling involving risk-communication, and theories for encapsulating subjective interactions between actors. Each research topic has additional research avenues that were not studied for this thesis. For example, a summary was provided in Chapter 5 for mapping BPMN and jBPM to fulfil the requirements for a BDI software agent. Due to time constraints, the full functionality was not implemented, indeed this is one area for further study.

A new case study involving risk communication with the aim of producing a risk communication tool would be one ideal candidate for further study. The case study should involve ethnographic studies to generate data about the problem domain, with the construction of BPMNdm diagrams and the usage and further development of the Drools rules presented in Chapter 4. The resulting workflow models and rules could then be evaluated by the client, and once validated, used to help generate tool requirements or simulate specific scenarios that the client was interested in evaluating.

7.3.1 Detailed research questions

This research has motivated a number of research themes arising from the future work and the limitations of the work presented in this PhD:

Decision making algorithms that do not assume pairwise comparisons

The TOPSIS algorithm is one of a number of algorithms available to simulate decision making activities. Alternatives include the VIKOR method and ELECTRE [164]. However, these algorithms still assume that decision makers use pairwise comparisons when appraising options. While the pairwise comparison process is regarded as being analogous to the cognitive process of making decisions [134], [156], the relative effort required for decision makers to perform pairwise comparisons when accessing options against multiple criteria is an intensive task [134]. Furthermore, it can not be assumed that every type of decision-making activity used pairwise comparisons. While TFNs offer a method for accounting for uncertainty and subjectivity, the lack of a Multi Criteria Decision Making (MCDM) algorithm that does not assume pairwise comparisons are undertaken limits the usage of these algorithms to scenarios that the investigator has overseen. Therefore, it can not be assumed that pairwise comparisons are a realistic reflection of the actual comparisons made in the problem domain.

Formalised definition for the visualisation for extended BPMN elements

As discussed in Chapter 3, the BPMN specification does not specify how formalism extensions should be visualised [42]. While the BPMN formalism is documented and modellers can included their own extensions using the XML Schema Definition (XSD) documents to provide the logical and structure components of their extension, the BPMN standard does not specify how these extensions should be visualised. Therefore, modellers are free to choose how their extended elements should be visualised, or choose not to visualise them.

A formalised standard, similar to a Human Interface Guideline document developed for GUIs would allow modellers to create their own visualisations for their extensions, while conforming to the visual style of the existing BPMN elements. A formalised standard would enable existing BPMN editors to simplify the creation of new extended BPMN elements.

References

[1] A. Edwards, E. Matthews, R. Pill, and M. Bloor, ‘Communication about risk: the responses of primary care professionals to standardizing the’language of risk’and communication tools.’, *Fam. Pract.*, vol. 15, no. 4, pp. 301–307, 1998.
[2] D. Flynn *et al.*, ‘Engaging patients in health care decisions in the emergency department through shared decision-making: a systematic review’, *Acad. Emerg. Med. Off. J. Soc. Acad. Emerg. Med.*, vol. 19, no. 8, pp. 959–967, Aug. 2012.
[3] ‘10 dangers of the medieval period’, *History Extra*. [Online]. Available: http://www.historyextra.com/feature/medieval/10-dangers-medieval-period. [Accessed: 25-May-2016].
[4] M. A. Riva, M. Benedetti, and G. Cesana, ‘Pandemic Fear and Literature: Observations from Jack London’s The Scarlet Plague’, *Emerg. Infect. Dis.*, vol. 20, no. 10, pp. 1753–1757, Oct. 2014.
[5] ‘Worldwide data | World Cancer Research Fund International’. [Online]. Available: http://www.wcrf.org/int/cancer-facts-figures/worldwide-data. [Accessed: 25-May-2016].
[6] A. S. Ahmad, N. Ormiston-Smith, and P. D. Sasieni, ‘Trends in the lifetime risk of developing cancer in Great Britain: comparison of risk for those born from 1930 to 1960’, *Br. J. Cancer*, vol. 112, no. 5, pp. 943–947, Mar. 2015.
[7] E. U. Weber, A.-R. Blais, and N. E. Betz, ‘A domain-specific risk-attitude scale: Measuring risk perceptions and risk behaviors’, *J. Behav. Decis. Mak.*, vol. 15, no. 4, pp. 263–290, 2002.
[8] B. Fischhoff, *Communicating Risks and Benefits: An Evidence Based User’s Guide: An Evidence Based User’s Guide*. Government Printing Office, 2012.
[9] Hugh Beyer, *Contextual design : defining customer-centered systems*. San Francisco, Calif: Morgan Kaufmann, 1998.
[10] D. Flynn, G. A. Ford, L. Stobbart, H. Rodgers, M. J. Murtagh, and R. G. Thomson, ‘A review of decision support, risk communication and patient information tools for thrombolytic treatment in acute stroke: lessons for tool developers’, *BMC Health Serv. Res.*, vol. 13, no. 1, p. 225, Jun. 2013.
[11] D. Flynn *et al.*, ‘Development of a computerised decision aid for thrombolysis in acute stroke care’, *BMC Med. Inform. Decis. Mak.*, vol. 15, no. 1, p. 6, Feb. 2015.
[12] G. Loewenstein, ‘Hot-cold empathy gaps and medical decision making.’, *Health Psychol.*, vol. 24, no. 4S, p. S49, 2005.
[13] R. Y. Hirokawa and M. S. Poole, *Communication and group decision making*. SAGE Publications, 1996.
[14] V. M. Papadakis, S. Lioukas, and D. Chambers, ‘Strategic decision-making processes: the role of management and context’, *Strateg. Manag. J.*, vol. 19, no. 2, pp. 115–147, 1998.

[15] M. A. Korsgaard, D. M. Schweiger, and H. J. Sapienza, ‘Building commitment, attachment, and trust in strategic decision-making teams: The role of procedural justice’, *Acad. Manage. J.*, vol. 38, no. 1, pp. 60–84, 1995.

[16] H Qudrat-Ullah, J. M. Spector, and P. I Davidsen, *Complex decision making: theory and practice*. Berlin ; New York: Springer, 2008.

[17] C. A. Klöckner and E. Matthies, ‘How habits interfere with norm-directed behaviour: A normative decision-making model for travel mode choice’, *J. Environ. Psychol.*, vol. 24, no. 3, pp. 319–327, Sep. 2004.

[18] J. W. Payne, J. R. Bettman, and M. F. Luce, ‘Chapter 5 - Behavioral Decision Research: An Overview A2 - Birnbaum, Michael H.’, in *Measurement, Judgment and Decision Making*, San Diego: Academic Press, 1998, pp. 303–359.

[19] D. E. Bell and H. Raiffa, *Decision Making: Descriptive, Normative, and Prescriptive Interactions*. Cambridge University Press, 1988.

[20] N. Suhonen, ‘Normative and descriptive theories of decision making under risk: A short review’, *Joensuu Finl. Univ. East. Finl.*, 2007.

[21] H. Kunreuther *et al.*, ‘High stakes decision making: Normative, descriptive and prescriptive considerations’, *Mark. Lett.*, vol. 13, no. 3, pp. 259–268, 2002.

[22] M. Peterson, ‘An introduction to decision theory’, *Camb. Books*, 1993.

[23] ‘Ovid: Group Decision Making and Normative Versus Informational Influence: Effects of Type of Issue and Assigned Decision Rule.’ [Online]. Available: http://ovidsp.tx.ovid.com/sp-3.20.0b/ovidweb.cgi?QS2=434f4e1a73d37e8c6dcfcba20050f7039a7cdbca030f04c9e34f41eb0e4de616f27f45882563dab105774e574c8ea936f593bb12f57b0af7c6ac553d96ddcf9c7b2b6d8c56687d0767954a81ebedf206bda191b647729ff62c6d2816bac3c5e6f6f1a8dce098f9631c6ee7458fff55865717b449290b97752c1a3fa2419b718f13c33abbca38b8d0907ff93d05dad6120a5b4135ca5975a4b64d2d49a6b6caeabfc95da8eb71b70abfa3bf8d62d9c57210e2e66bd74725820dc8273751252e5dc215c129f61f9e00203e64630219d3333f01e7309864256ee4a5c080184e60dce047f4fce1a56b01be86c8586726e9712aa7fb45cce9a7d3270e57fa7d5b90ff55311debdf1697b4389b7291440c79ca3a6fa9c6ab3d0326. [Accessed: 21-Jun-2016].

[24] J. Baron, *Normative Models ofJudgment and Decision Making*. The Blackwell handbook of judgment and decision making, 2004.

[25] D. J. Koehler and N. Harvey, *Blackwell handbook of judgment and decision making*. John Wiley & Sons, 2008.

[26] W. M. van der Aalst, ‘The application of Petri nets to workflow management’, *J. Circuits Syst. Comput.*, vol. 8, no. 01, pp. 21–66, 1998.

[27] S. A. White, ‘Process modeling notations and workflow patterns’, *Work. Handb.*, vol. 2004, pp. 265–294, 2004.

[28] B. Kiepuszewski, A. H. ter Hofstede, and W. M. van der Aalst, ‘Fundamentals of control flow in workflows’, *Acta Inform.*, vol. 39, no. 3, pp. 143–209, 2003.

[29] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, ‘Workflow Patterns’, *Distrib. Parallel Databases*, vol. 14, no. 1, pp. 5–51, Jul. 2003.

[30] ‘Workflow Patterns | Patterns | Control’. [Online]. Available: http://www.workflowpatterns.com/patterns/control/. [Accessed: 03-Aug-2015].

[31] M. Salatino, *JBPM 5 Developer Guide*. Packt Publishing Ltd, 2012.

[32] A. Basu and R. W. Blanning, ‘A Formal Approach to Workflow Analysis’, *Inf. Syst. Res.*, vol. 11, no. 1, p. 17, Mar. 2000.

[33] P. J. Kammer, G. A. Bolcer, R. N. Taylor, A. S. Hitomi, and M. Bergman, 'Techniques for supporting dynamic and adaptive workflow', *Comput. Support. Coop. Work CSCW*, vol. 9, no. 3–4, pp. 269–292, 2000.

[34] J. C. Recker and A. Dreiling, 'Does it matter which process modelling language we teach or use? an experimental study on understanding process modelling languages without formal education', 2007.

[35] S. A. White, 'Introduction to BPMN', *IBM Coop.*, pp. 2008–029, 2004.

[36] M. Chinosi and A. Trombetta, 'BPMN: An introduction to the standard', *Comput. Stand. Interfaces*, vol. 34, no. 1, pp. 124–134, Jan. 2012.

[37] W. M. van der Aalst and A. H. Ter Hofstede, 'YAWL: yet another workflow language', *Inf. Syst.*, vol. 30, no. 4, pp. 245–275, 2005.

[38] W. M. van der Aalst, L. Aldred, M. Dumas, and A. H. ter Hofstede, 'Design and implementation of the YAWL system', in *CAiSE*, 2004, vol. 3084, pp. 142–159.

[39] 'Chapter 6. Core Engine: BPMN 2.0'. [Online]. Available: https://docs.jboss.org/jbpm/v5.1/userguide/ch06.html. [Accessed: 21-Jun-2016].

[40] A. Rozinat, M. Wynn, W. van der Aalst, A. ter Hofstede, and C. Fidge, 'Workflow Simulation for Operational Decision Support Using Design, Historic and State Information', in *Business Process Management*, 2008, pp. 196–211.

[41] The Object Management Group, *BPMN 2.0 specification document*. 2010.

[42] L. J. R. Stroppi, O. Chiotti, and P. D. Villarreal, 'Extending BPMN 2.0: Method and Tool Support', in *Business Process Model and Notation*, Springer, 2011, pp. 59–73.

[43] P. Cotofrei and K. Stoffel, 'Fuzzy Extended BPMN for Modelling Crime Analysis Processes'.

[44] D. Gagne and A. Trudel, 'Time-BPMN', in *2009 IEEE Conference on Commerce and Enterprise Computing*, 2009, pp. 361–367.

[45] M. Brambilla, P. Fraternali, and C. Vaca, 'BPMN and Design Patterns for Engineering Social BPM Solutions'.

[46] K. Saeedi, L. Zhao, and P. R. F. Sampaio, 'Extending BPMN for Supporting Customer-Facing Service Quality Requirements', in *2010 IEEE International Conference on Web Services (ICWS)*, 2010, pp. 616–623.

[47] B. Brodt, 'BPMN2 Modeler - Documentation'. [Online]. Available: https://www.eclipse.org/bpmn2-modeler/documentation.php. [Accessed: 27-Apr-2016].

[48] N. Mulyar, W. M. van der Aalst, and N. Russell, 'Process flexibility patterns', *Tech. Univ. Eindh.*, 2008.

[49] H. A. Reijers, N. Russell, S. Geer, and G. A. M. Krekels, 'Workflow for healthcare: A methodology for realizing flexible medical treatment processes', in *Business Process Management Workshops*, 2010, pp. 593–604.

[50] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic*, vol. 4. Prentice Hall New Jersey, 1995.

[51] R. Layard, 'Measuring subjective well-being', *Science*, vol. 327, no. 5965, pp. 534–535, 2010.

[52] L. Tinkler and S. Hicks, 'Measuring subjective well-being', *Lond. Off. Natl. Stat.*, 2011.

[53] D. Kahneman and A. B. Krueger, 'Developments in the Measurement of Subjective Well-Being', *J. Econ. Perspect.*, vol. 20, no. 1, pp. 3–24, Jan. 2006.

[54] A. Bandura, *Self-efficacy: The exercise of control*. Worth Publishers, 1997.

[55] E. Bonabeau, 'Agent-Based Modeling: Methods and Techniques for Simulating Human Systems', *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 10, pp. 7280–7287, 2002.

[56] M. Klein and C. Dellarocas, 'A knowledge-based approach to handling exceptions in workflow systems', *Comput. Support. Coop. Work CSCW*, vol. 9, no. 3–4, pp. 399–412, 2000.

[57] S. Marsella and J. Gratch, 'Modeling coping behavior in virtual humans: don't worry, be happy', in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003, pp. 313–320.

[58] J. Gratch and S. Marsella, 'A domain-independent framework for modeling emotion', *Cogn. Syst. Res.*, vol. 5, no. 4, pp. 269–306, 2004.

[59] T. J. Meine *et al.*, 'Association of intravenous morphine use and outcomes in acute coronary syndromes: results from the CRUSADE Quality Improvement Initiative', *Am. Heart J.*, vol. 149, no. 6, pp. 1043–1049, 2005.

[60] S. Maalal and M. Addou, 'A new approach of designing Multi-Agent Systems', *ArXiv Prepr. ArXiv12041581*, 2012.

[61] A. S. Rao, M. P. Georgeff, and others, 'BDI agents: From theory to practice.', in *ICMAS*, 1995, vol. 95, pp. 312–319.

[62] H. Endert, T. Küster, B. Hirsch, and S. Albayrak, 'Mapping BPMN to agents: An analysis', *Agents Web-Serv. Ontol. Integr. Methodol.*, pp. 43–58, 2007.

[63] B. S. Onggo and O. Karpat, 'Agent-based conceptual model representation using BPMN', in *Proceedings of the Winter Simulation Conference*, 2011, pp. 671–682.

[64] M. Goldstein and others, 'Subjective Bayesian analysis: principles and practice', *Bayesian Anal.*, vol. 1, no. 3, pp. 403–420, 2006.

[65] H. Pan and D. McMichael, 'Fuzzy causal probabilistic networks-a new ideal and practical inference engine', in *Proc. 1st International Conference on Multisource-Multisensor Information Fusion*, 1998, pp. 6–8.

[66] E. W. Weisstein, 'Conditional Probability'. [Online]. Available: http://mathworld.wolfram.com/ConditionalProbability.html. [Accessed: 15-Apr-2016].

[67] D. Dubois and H. Prade, 'When upper probabilities are possibility measures', *Fuzzy Sets Syst.*, vol. 49, no. 1, pp. 65–74, Jul. 1992.

[68] M. Inuiguchi and T. Tanino, 'Necessity measures and fuzzy rough sets defined by certainty qualifications', in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, 2001, vol. 4, pp. 1940–1945 vol.4.

[69] A. Kaufmann, M. M. Gupta, and A. Kaufmann, *Introduction to fuzzy arithmetic: theory and applications*. Van Nostrand Reinhold Company New York, 1985.

[70] X. Wang, D. Ruan, and E. E. Kerre, *Mathematics of fuzziness—basic issues*, vol. 245. Springer Science & Business Media, 2009.

[71] T. Allahviranloo and R. Saneifard, 'Defuzzification method for ranking fuzzy numbers based on center of gravity', *Iran. J. Fuzzy Syst.*, vol. 9, no. 6, pp. 57–67, 2012.

[72] U. Bodenhofer, 'Fuzzy Orderings of Fuzzy Sets'.

[73] Y. Yuan, 'Criteria for evaluating fuzzy ranking methods', *Fuzzy Sets Syst.*, vol. 43, no. 2, pp. 139–157, Sep. 1991.

[74] M. J. Palenchar and R. L. Heath, 'Strategic risk communication: Adding value to society', *Public Relat. Rev.*, vol. 33, no. 2, pp. 120–129, Jun. 2007.

[75] D. Petrova, R. Garcia-Retamero, A. Catena, and J. van der Pligt, 'To screen or not to screen: What factors influence complex screening decisions?', *J. Exp. Psychol. Appl.*, vol. 22, no. 2, pp. 247–260, Jun. 2016.

[76] D. Sobolev and N. Harvey, 'Assessing Risk in Graphically Presented Financial Series', *Risk Anal.*, vol. 36, no. 12, pp. 2216–2232, Dec. 2016.

[77] D. Petrova and R. Garcia-Retamero, ‘Commentary: Risky decision-making is associated with residential choice in healthy older adults’, *Front. Psychol.*, vol. 7, Aug. 2016.
[78] A. Bandura, ‘Self-efficacy: toward a unifying theory of behavioral change.’, *Psychol. Rev.*, vol. 84, no. 2, p. 191, 1977.
[79] M. S. Eastin and R. LaRose, ‘Internet Self-Efficacy and the Psychology of the Digital Divide’, *J. Comput. Commun.*, vol. 6, no. 1, pp. 0–0, Sep. 2000.
[80] J. Lam and M. Lee, ‘Bridging the Digital Divide-The Roles of Internet Self-Efficacy towards Learning Computer and the Internet among Elderly in Hong Kong, China’, 2005.
[81] D. Grembowski *et al.*, ‘Self-Efficacy and Health Behavior Among Older Adults’, *J. Health Soc. Behav.*, vol. 34, no. 2, pp. 89–104, Jun. 1993.
[82] L. Trevena, E. Peters, J. King, M. Galesic, and E. Ozanne, ‘2012 UPDATED CHAPTER C: PRESENTING PROBABILITIES SECTION 1: AUTHORS/AFFILIATIONS’.
[83] S. T. Hawley, B. Zikmund-Fisher, P. Ubel, A. Jancovic, T. Lucas, and A. Fagerlin, ‘The impact of the format of graphical presentation on health-related knowledge and treatment choices’, *Patient Educ. Couns.*, vol. 73, no. 3, pp. 448–455, Dec. 2008.
[84] J. S. Ancker, Y. Senathirajah, R. Kukafka, and J. B. Starren, ‘Design Features of Graphs in Health Risk Communication: A Systematic Review’, *J. Am. Med. Inform. Assoc.*, vol. 13, no. 6, pp. 608–618, Nov. 2006.
[85] I. M. Lipkus and J. G. Hollands, ‘The Visual Communication of Risk’, *JNCI Monogr.*, vol. 1999, no. 25, pp. 149–163, Jan. 1999.
[86] E. R. Stone, W. Bruine de Bruin, A. M. Wilkins, E. M. Boker, and J. MacDonald Gibson, ‘Designing Graphs to Communicate Risks: Understanding How the Choice of Graphical Format Influences Decision Making’, *Risk Anal.*, vol. 37, no. 4, pp. 612–628, Apr. 2017.
[87] L. J. Trevena, A. Barratt, P. Butow, and P. Caldwell, ‘A systematic review on communicating with patients about evidence’, *J. Eval. Clin. Pract.*, vol. 12, no. 1, pp. 13–23, Jan. 2006.
[88] A. Gupta, *Risk Management and Simulation*. CRC Press, 2016.
[89] R. K. Hanson and M. T. Bussiere, ‘Predicting relapse: a meta-analysis of sexual offender recidivism studies.’, *J. Consult. Clin. Psychol.*, vol. 66, no. 2, p. 348, 1998.
[90] N. L\a angström and M. Grann, ‘Risk for criminal recidivism among young sex offenders’, *J. Interpers. Violence*, vol. 15, no. 8, pp. 855–871, 2000.
[91] P. Krause, J. Fox, P. Judson, and M. Patel, ‘Qualitative risk assessment fulfils a need’, in *Applications of Uncertainty Formalisms*, Springer, 1998, pp. 138–156.
[92] L. T. Nowell, ‘Graphical encoding for information visualization: using icon color, shape, and size to convey nominal and quantitative data’, 1997.
[93] ‘SAGE: Qualitative Research Practice: A Guide for Social Science Students and Researchers: Jane Ritchie: 9780761971108’. [Online]. Available: http://www.sagepub.in/textbooks/Book211399. [Accessed: 27-Apr-2016].
[94] L. Wilkinson and M. Friendly, ‘The history of the cluster heat map’, *Am. Stat.*, 2012.
[95] ‘Risk Heat Maps - CGMA’. [Online]. Available: http://www.cgma.org/Resources/Tools/essential-tools/Pages/risk-heat-maps.aspx?TestCookiesEnabled=redirect. [Accessed: 27-Apr-2016].
[96] L. J. R. Stroppi, O. Chiotti, and P. D. Villarreal, ‘Extending BPMN 2.0: Method and Tool Support’, in *Business Process Model and Notation*, Springer, 2011, pp. 59–73.

[97] J. M. Wardlaw, V. Murray, E. Berge, and G. J. del Zoppo, 'Thrombolysis for acute ischaemic stroke', in *Cochrane Database of Systematic Reviews*, John Wiley & Sons, Ltd, 2014.
[98] J. M. Wardlaw *et al.*, 'Recombinant tissue plasminogen activator for acute ischaemic stroke: an updated systematic review and meta-analysis', *Lancet Lond. Engl.*, vol. 379, no. 9834, pp. 2364–2372, Jun. 2012.
[99] M. G. Lansberg, G. W. Albers, and C. A. C. Wijman, 'Symptomatic intracerebral hemorrhage following thrombolytic therapy for acute ischemic stroke: a review of the risk factors', *Cerebrovasc. Dis. Basel Switz.*, vol. 24, no. 1, pp. 1–10, 2007.
[100] 'PLOS ONE: Situationally-Sensitive Knowledge Translation and Relational Decision Making in Hyperacute Stroke: A Qualitative Study'. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0037066. [Accessed: 13-May-2016].
[101] W. Hacke *et al.*, 'Thrombolysis with alteplase 3 to 4.5 hours after acute ischemic stroke', *N. Engl. J. Med.*, vol. 359, no. 13, pp. 1317–1329, Sep. 2008.
[102] K. R. Lees *et al.*, 'Time to treatment with intravenous alteplase and outcome in stroke: an updated pooled analysis of ECASS, ATLANTIS, NINDS, and EPITHET trials', *Lancet Lond. Engl.*, vol. 375, no. 9727, pp. 1695–1703, May 2010.
[103] N. Wahlgren *et al.*, 'Thrombolysis with alteplase for acute ischaemic stroke in the Safe Implementation of Thrombolysis in Stroke-Monitoring Study (SITS-MOST): an observational study', *Lancet Lond. Engl.*, vol. 369, no. 9558, pp. 275–282, Jan. 2007.
[104] G. W. Albers and J.-M. Olivot, 'Intravenous alteplase for ischaemic stroke', *The Lancet*, vol. 369, no. 9558, pp. 249–250, Jan. 2007.
[105] M. Dirks *et al.*, 'Intravenous thrombolysis in acute ischaemic stroke: from trial exclusion criteria to clinical contraindications. An international Delphi study', *J. Neurol. Neurosurg. Psychiatry*, vol. 78, no. 7, pp. 685–689, Jul. 2007.
[106] J. M. Wardlaw, V. Murray, E. Berge, and G. J. Del Zoppo, 'Thrombolysis for acute ischaemic stroke', *Cochrane Database Syst. Rev.*, no. 4, p. CD000213, 2009.
[107] V. A. Entwistle and I. S. Watt, 'Patient involvement in treatment decision-making: the case for a broader conceptual framework', *Patient Educ. Couns.*, vol. 63, no. 3, pp. 268–278, Nov. 2006.
[108] '"Two per cent isn't a lot, but when it comes to death it seems quite a lot anyway": patients' perception of risk and willingness to accept risks as... - PubMed - NCBI'. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19103942. [Accessed: 16-May-2016].
[109] 'Thrombolytic treatment for stroke: patient preferences for treatment, information, and involvement. - PubMed - NCBI'. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19110139. [Accessed: 16-May-2016].
[110] 'Patient preferences for stroke outcomes.' [Online]. Available: http://stroke.ahajournals.org/content/25/9/1721. [Accessed: 16-May-2016].
[111] V. M. Montori, A. Gafni, and C. Charles, 'A shared treatment decision-making approach between patients with chronic conditions and their clinicians: the case of diabetes', *Health Expect.*, vol. 9, no. 1, pp. 25–36, 2006.
[112] J. Hamann, R. Cohen, S. Leucht, R. Busch, and W. Kissling, 'Shared decision making and long-term outcome in schizophrenia treatment', *J. Clin. Psychiatry*, vol. 68, no. 7, pp. 992–997, Jul. 2007.
[113] D. A. D. Brun and D. D. Flynn, 'Co-production of a Theory-Based Film about Shared Decision Making in Primary Care for Depression', *http://eprints.ncl.ac.uk*, 2015.

[114] D. Flynn *et al.*, *A theory-based educational film about shared decision making in primary care for depression*. Newcastle upon Tyne: Newcastle University, Digital Media Services, 2015.
[115] 'About us'. [Online]. Available: http://www.emishealth.com/about-us/. [Accessed: 27-Aug-2015].
[116] *BPMN*. .
[117] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, 'Workflow Patterns', *Distrib. Parallel Databases*, vol. 14, no. 1, pp. 5–51, Jul. 2003.
[118] N. Russell, A. H. M. Ter Hofstede, D. Edmond, and W. M. P. van der Aalst, 'Workflow data patterns', QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane, 2004.
[119] *Workflow Patterns \textbar Patterns \textbar Control*. .
[120] N. Mulyar, W. M. van der Aalst, and N. Russell, 'Process flexibility patterns', *Tech. Univ. Eindh.*, 2008.
[121] M. Bali, *Drools JBoss Rules 5. X Developer's Guide*. Packt Publishing Ltd, 2013.
[122] 'Drools example jBPM 5'. [Online]. Available: http://www.mastertheboss.com/jboss-jbpm/drools/drools-example-jbpm-5. [Accessed: 30-Mar-2016].
[123] R. Kaiser, W. Weiss, M. Falelakis, S. Michalakopoulos, and M. F. Ursu, 'A rule-based virtual director enhancing group communication', in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, 2012, pp. 187–192.
[124] John D. Brewer, *Ethnography*. Philadelphia, PA: Open University Press, 2000.
[125] D. L. Jorgensen, *Participant Observation: A Methodology for Human Studies*. SAGE, 1989.
[126] V. Braun and V. Clarke, 'Using thematic analysis in psychology', *Qual. Res. Psychol.*, vol. 3, no. 2, pp. 77–101, 2006.
[127] R. Anderson, 'Thematic content analysis (TCA)', *Descr. Present. Qual. DataOnline Available Httpwww Wellknowingconsulting OrgpublicationspdfsThematicContentAnalysis Pdf Accessed 8 Dec. 2011*, 2007.
[128] M. Fitting, 'First-Order Logic', in *First-Order Logic and Automated Theorem Proving*, Springer, 1990, pp. 97–125.
[129] *Multi agency public protection arrangements*. United Kingdom: National MAPPA Team National Offender Management Service Offender Management and Public Protection Group, Ministry of Justice, 2012.
[130] P. PROTECTION, 'FORCE PROCEDURES'.
[131] E. Participation, 'Data Protection Act 1998'. [Online]. Available: http://www.legislation.gov.uk/ukpga/1998/29/contents. [Accessed: 28-Jun-2016].
[132] B. S. S. Onggo, 'BPMN pattern for agent-based simulation model representation', in *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 2012, pp. 1–10.
[133] L. F. Degner, J. A. Sloan, and P. Venkatesh, 'The Control Preferences Scale', *Can. J. Nurs. Res. Rev. Can. Rech. En Sci. Infirm.*, vol. 29, no. 3, pp. 21–43, 1997.
[134] C.-H. Yeh and Y.-H. Chang, 'Modeling subjective evaluation for fuzzy group multicriteria decision making', *Eur. J. Oper. Res.*, vol. 194, no. 2, pp. 464–473, Apr. 2009.
[135] S.-J. Chen and S.-M. Chen, 'Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers', *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 1, pp. 45–56, Feb. 2003.
[136] C.-T. Chen, 'A fuzzy approach to select the location of the distribution center', *Fuzzy Sets Syst.*, vol. 118, no. 1, pp. 65–73, Feb. 2001.

[137] C.-H. Cheng, 'Evaluating weapon systems using ranking fuzzy numbers', *Fuzzy Sets Syst.*, vol. 107, no. 1, pp. 25–35, Oct. 1999.
[138] 'Research'. [Online]. Available: http://www.ncl.ac.uk/ihs/research/project/3842. [Accessed: 31-Mar-2016].
[139] J. Aczél and T. L. Saaty, 'Procedures for synthesizing ratio judgements', *J. Math. Psychol.*, vol. 27, no. 1, pp. 93–102, Mar. 1983.
[140] J. Barzilai, W. D. Cook, and B. Golany, 'Consistent weights for judgements matrices of the relative importance of alternatives', *Oper. Res. Lett.*, vol. 6, no. 3, pp. 131–134, Jul. 1987.
[141] P.-T. Chang and K.-C. Hung, 'Applying the fuzzy-weighted-average approach to evaluate network security systems', *Comput. Math. Appl.*, vol. 49, no. 11, pp. 1797–1814, Jun. 2005.
[142] D. Dubois, L. Foulloy, G. Mauris, and H. Prade, 'Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities', *Reliab. Comput.*, vol. 10, no. 4, pp. 273–297, 2004.
[143] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic*, vol. 4. Prentice Hall New Jersey, 1995.
[144] M. Pal and others, 'Triangular fuzzy matrices', *Iran. J. Fuzzy Syst.*, vol. 4, no. 1, pp. 75–87, 2007.
[145] M. G. Voskoglou, 'Use of the Triangular Fuzzy Numbers for Student Assessment', *Am. J. Appl. Math. Stat. Am. J. Appl. Math. Stat.*, vol. 3, no. 4, pp. 146–150, Jul. 2015.
[146] K. Delaere *et al.*, 'Fuzzy Webtools Website User Manual', 2010.
[147] A. Eissa, I. Krass, and B. V. Bajorek, 'Barriers to the utilization of thrombolysis for acute ischaemic stroke', *J. Clin. Pharm. Ther.*, vol. 37, no. 4, pp. 399–409, Aug. 2012.
[148] C. Charles, A. Gafni, and T. Whelan, 'Shared decision-making in the medical encounter: what does it mean?(or it takes at least two to tango)', *Soc. Sci. Med.*, vol. 44, no. 5, pp. 681–692, 1997.
[149] L. Sandman and C. Munthe, 'Shared Decision Making, Paternalism and Patient Choice', *Health Care Anal.*, vol. 18, no. 1, pp. 60–84, Mar. 2010.
[150] D. D. Flynn *et al.*, 'A Computerised Decision Aid for Treatment of Acute Stroke with Thrombolysis (COMPASS)', *http://eprints.ncl.ac.uk*, 2015.
[151] C.-L. Hwang, Y.-J. Lai, and T.-Y. Liu, 'A new approach for multiple objective decision making', *Comput. Oper. Res.*, vol. 20, no. 8, pp. 889–899, Oct. 1993.
[152] H.-S. Shih, H.-J. Shyur, and E. S. Lee, 'An extension of TOPSIS for group decision making', *Math. Comput. Model.*, vol. 45, pp. 801–813, 2007.
[153] J. W. Gerdes and E. Spero, 'A Compact Review of Multi-criteria Decision Analysis Uncertainty Techniques', DTIC Document, 2013.
[154] O. S. Vaidya and S. Kumar, 'Analytic hierarchy process: An overview of applications', *Eur. J. Oper. Res.*, vol. 169, no. 1, pp. 1–29, Feb. 2006.
[155] Thomas L. Saaty, 'Models, methods, concepts & applications of the analytic hierarchy process'. Springer, 2012.
[156] Z. Zhang, X. Liu, and S. Yang, 'A Note on the 1-9 Scale and Index Scale In AHP', in *Cutting-Edge Research Topics on Multiple Criteria Decision Making*, Y. Shi, S. Wang, Y. Peng, J. Li, and Y. Zeng, Eds. Springer Berlin Heidelberg, 2009, pp. 630–634.
[157] R. F. Dyer and E. H. Forman, 'Group decision support with the Analytic Hierarchy Process', *Decis. Support Syst.*, vol. 8, no. 2, pp. 99–124, Apr. 1992.
[158] Salih Duffuaa, 'TOPSIS presentation'. Systems Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM).
[159] M. Salatino, *JBPM 5 Developer Guide*. Packt Publishing Ltd, 2012.

[160] ‘Inteligencia Artificial: JBoss JBPM - Generating an image at runtime (JPDL -> PNG)’. .
[161] ‘ProcessEventListener (Knowledge API 5.2.0-SNAPSHOT API)’. [Online]. Available: https://docs.jboss.org/jbpm/v5.1/javadocs/org/drools/event/process/ProcessEventListener.html. [Accessed: 06-Jul-2016].
[162] ‘org.w3c.dom (Java Platform SE 7)’. [Online]. Available: https://docs.oracle.com/javase/7/docs/api/org/w3c/dom/package-summary.html. [Accessed: 21-Oct-2015].
[163] T. Wahl and G. Sindre, ‘An analytical evaluation of BPMN using a semiotic quality framework’, *Adv. Top. Database Res.*, vol. 5, pp. 94–105, 2006.
[164] S. Opricovic and G.-H. Tzeng, ‘Extended VIKOR method in comparison with outranking methods’, *Eur. J. Oper. Res.*, vol. 178, no. 2, pp. 514–529, Apr. 2007.