

Requirements Engineering for Electronic Healthcare Records

Alexandru Matei

Supervisor: Dr Emmanuel Letier

Supervisor: Alan Payne

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of the
University College London.

Department of Computer Science
University College London

October 15, 2018

Declaration

I, Alexandru Matei, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This thesis investigates requirements engineering methods based on process modelling for Electronic Healthcare Record (EHR) systems.

The relation between software requirements and user workflows is essential in healthcare settings: EHRs are expected to improve clinical and administrative workflows. In turn, the new workflows are expected to satisfy a number of business goals. If a new software system does not support the desired clinical workflows or patient journeys, then its value and benefits are often disputed by stakeholders. Our hypothesis is that requirements engineering methods based on process models will contribute to the overall success of EHR projects in the industry. By success, we mean software systems that are in use and meet the business benefits expected of them.

The experiments presented in this thesis are aimed to develop and evaluate a method that allows business analysts to make use of process models during requirements engineering for EHRs. The goal of the method is to ensure the software specification is aligned to and supports the user workflows. Each of the four experiments addresses a specific research objective, and thus the findings from each experiment constitute the basis for one of our four contributions to science.

Experiment 1: Relating Goal Oriented Requirements Engineering and Process Modelling

This experiment investigates the design of a common framework for describing process models and software requirements. It relates the KAOS framework for goal oriented requirements engineering and the Business Process Modelling Notation (BPMN). Our goal is to facilitate requirements elicitation. Specifically, business analysts using our framework should be able to reason about the alignment of the software specification to the business processes, and identify specific changes that improve this alignment (either changes in the design of the system, or changes in the business processes).

This first experiment was conducted as part of the WellbeingUCL project, supported by Boots.

Experiment 2: Inferring Goal Models from Process Models

The second experiment investigates a method for business analysts to derive software requirements from process models. The purpose for defining such a method is to provide sufficient guidance to business analysts, during requirements elicitation. Our aim is to help business analysts elicit meaningful goal models and shape the design of the system-to-be, in light of these goals. A number of heuristics to facilitate requirements elicitation are proposed and evaluated, considering the trade-offs between a fully automated and a human driven process.

Experiment 3: Electronic Healthcare Record for Bupa

The third experiment evaluates the requirements engineering method during an EHR implementation for a chronic condition management service delivered by Bupa nurses in South West England. Action research is used to assess the impact and fit of the requirements elicitation process, in relation to the current work practices of business analysts in the industry. The extended KAOS framework and goal inference heuristics have been used to inform the final software specification, guide the workflow redesign and clarify the business benefits. From a project management perspective, this experiment evaluates how the KAOS method aligns with the Agile and Lean methodologies used in Bupa. The project has delivered an EHR system actively used to support the care of 2,600 patients.

Experiment 4: Personal Health Record for Nuffield Health

The fourth experiment evaluates the extended KAOS framework when developing a new digital customer proposition with an underlying EHR system.

It investigates how consumer journeys can be modelled as KAOS process models. Of specific interest is the ability of the framework to clarify the responsibility assignments among the different agents (i.e. system components) that need to collaborate to deliver the end to end customer journey.

The experiment was run as an action research project, in partnership with Nuffield Health. The results have informed the architecture of an open source personal health record for lifestyle data.

Contributions to science

This thesis advances the field of requirements engineering by introducing and evaluating a requirements elicitation method based on business process models. It also presents new evidence into the use of goal oriented requirements engineering for the design and implementation of EHR systems in the industry.

Our four contributions to science directly follow from the results of the four experiments conducted as part of this research. Our first two contributions cover the conceptual framework and our proposed method for requirement elicitation based on process models. Our last two contributions present evidence for the practical use and benefits of our goal oriented requirements engineering method in industry based projects.

First, we present an extension of the KAOS requirements engineering framework which includes a business process view with clearly defined syntax and execution semantic. This approach ensures process models and goal models have a shared semantic. A new concept, that of Intentional Fragment, captures the explicit relation between fragments of a process model and a specific goal. We also define additional consistency rules, to clarify how the process view relates with other KAOS models: object, agent and operation model.

Secondly, we present a set of goal inference techniques to help analysts build goal models starting from process models. In effect, analysts can start from the artefacts that are most familiar to them (i.e. the workflow models) and gradually derive a goal model for the system-to-be. A set of 12 heuristics have been fully defined and integrated into a semi-structured method for goal elicitation.

Our third contribution is an evaluation of how the goal oriented requirements engineering method (incorporating workflow analysis) supports the design and deployment of a EHR system in a clinical setting. The project was representative for the challenges faced by healthcare organisations wishing to deploy EHRs: quality of care standards that impose constraints on process redesign; legacy systems that have shaped the workflow; organisational complexity and competing stakeholder interests. We show that by methodically applying our goal inference techniques we were able to produce a valid goal model starting from models of the nurses workflows. The resulting goal model was used to reason about alternative design options in the system-to-be, and to clarify the benefit case in deploying the EHR system.

Fourth, we examine the requirements engineering process for an EHR system meant to support a new customer proposition. This project was representative for the challenges faced in the digital health industry: a target consumer journey driven by user experience research; many different systems required to collaborate; focus on the architectural design of the system. We show that we can apply our goal inference techniques to customer journey maps and produce a meaningful goal model. This has been used to shape the architecture of the EHR system and reason about integration requirements. We also argue that our goal inference techniques complement agile development practices used within the organisation.

Impact Statement

The four experiments we conducted and the results presented in this thesis impact the work of academics, practitioners and healthcare organisations.

First, this thesis supports future scholarship in the field of goal oriented requirements engineering (GORE). KAOS, one of the main GORE frameworks, is not widely used in the industry because of its steep learning curve and perceived lack of guidelines [FGZ15]. This thesis presents a set of heuristics that guide practitioners during requirements elicitation, using process models as a starting point. The expected impact is an increased adoption of GORE frameworks in general and KAOS in particular in the industry. While additional research is required to quantify this impact, the process-based heuristics for goal elicitation do provide one way to address a documented problem - the perceived lack of methodological guidelines for novice GORE practitioners. From a methodological perspective, we have used action research to investigate requirements engineering methods for real industry projects. Action research has been only sporadically used in software engineering research. This thesis will further acceptance of action research as a suitable research method in this field.

Second, our contributions have an impact on the day to day practice of business analysts (BAs). In our third experiment, we used KAOS techniques to inform the system design and the benefit case for an industry based project - these are common BA responsibilities. To facilitate adoption of the goal inference heuristics in the wider BA community, we have also engaged Signavio, a vendor of business process management software, to present and discuss the findings from our experiments. Our extension to the KAOS framework is also relevant for software development teams following behaviour driven development (BDD). In our fourth experiment, we show how goal modelling can be integrated in the BDD methodology used in Nuffield Health. Software delivery teams familiar with BDD could adopt KAOS as an additional method in their tool set.

Third, our work is relevant for healthcare companies looking to develop or configure Electronic Healthcare Records (EHR) systems. EHRs are notoriously difficult to implement successfully. Often cited problems, such as undocumented workarounds or poor user adoption, can be traced back to misalignment between IT functionalities and user workflows. Combining goal and process modelling into a unified framework has the potential to mitigate these problems. Using the extended KAOS framework to design an EHR system and associated workflows can improve clinical practice, as the systems will be more robust and fit for purpose. To conduct our third experiment, we worked with Bupa to deliver an EHR system for 2,600 patients living in the Somerset area in the UK, diagnosed with chronic obstructive pulmonary disease (COPD). We have demonstrated improved user adoption and no emergent workarounds in the first 6 months of use. Our method has a direct impact on commercial activity as well, by helping healthcare providers demonstrate efficiencies gained from deploying EHR systems - relating investment in IT to improved operational performance. For our fourth experiment we worked with Nuffield Health to develop a reference architecture for a personal health and lifestyle record. The platform has been open sourced and has been made available to the National Health Service (NHS) in the UK. Thus, it can be adopted and deployed by both private healthcare organisations and NHS trusts. Two additional organisations have shown early interest in using the platform and supporting its development: UK Active and Aetna International. Aetna will start contributing to the open source project from October 2018. The initial area of interest is adapting the platform for use in the medical insurance domain.

Acknowledgements

First, I would like to thank my PhD supervisors, Dr. Emmanuel Letier and Alan Payne, for their continued guidance throughout my PhD. Their insights and expertise were invaluable in shaping this thesis, as I attempted to embed the rigour of academic research in software engineering into the reality of the healthcare industry.

I would also like to thank Prof. Philip Treleaven for his steadfast support and encouragement - his advice has been instrumental in completing this work, and in transforming my early research interest into what has become my career.

Furthermore, I would like to acknowledge the support of the healthcare organisations involved in this research: Boots UK, Bupa and Nuffield Health have offered me the access and flexibility required to pursue my academic work, and the opportunity to be part of some truly inspiring projects.

Of course, none of these would have been possible without the encouragement and patience of my friends and family. They have been a constant source of energy and inspiration throughout my journey. And last, but not least, to my mother.

Contents

1	Introduction	12
1.1	Research Motivation	12
1.2	Common Challenges of EHR Deployments	13
1.3	Research Objectives and Contributions	14
1.4	Structure of Thesis	15
2	Background	17
2.1	Process Modelling with BPMN	17
2.1.1	Business Process Modelling Notation	17
2.1.2	BPMN Semantic	18
2.1.3	Other Process Modelling Languages	19
2.1.4	OpenEHR - Domain Specific Languages for Modelling Care Pathways	19
2.2	Requirements Engineering with KAOS	20
2.2.1	The Goal Model	22
2.2.2	The Object Model	24
2.2.3	The Agent Model	25
2.2.4	The Operation Model	26
2.2.5	Goal Operationalisation	26
2.2.6	Consistency Rules	27
2.3	Meta-Model Comparison: BPMN and KAOS	28
3	Relating Goal Oriented Requirements Engineering and Process Modelling	29
3.1	Motivation	29
3.2	Related Work	30
3.3	Approach	32
3.4	WellbeingUCL Project	32
3.5	A Process Model View for KAOS	33
3.5.1	Meta-Model	33
3.5.2	Example Process Model: WellbeingUCL	35
3.5.3	Structural Constraints	36
3.5.4	Naming Conventions	37
3.5.5	BPMN Messages as KAOS Events	38
3.6	Process Annotations	38
3.6.1	Activities Specification	39
3.6.2	Gateways Specification	42
3.6.3	End Events Specification	43
3.6.4	Consistency Rules between Graphical and Textual Notation	44
3.7	Object Model and Process Model: Activities as KAOS Objects	47
3.8	Operational Model and Process Model: Operational Semantics of a Process Model	48
3.9	Goal Model and Process Model: Goal Satisfaction Arguments	50
3.9.1	Process Level Goal Satisfaction Arguments	51
3.9.2	Partial Goal Satisfaction Arguments	53
3.9.3	System Requirements and Domain Expectations	54
3.10	Intentional Fragments	55
3.10.1	Intuitive Definition	55
3.10.2	Motivation	56

3.10.3	Intentional Fragments in the WellbeingUCL Process	56
3.10.4	Applications of Intentional Fragments	57
3.10.5	Formal Definition	58
3.11	Alignment between Process and Goal Models	58
3.12	Summary	59
4	Inferring Goal Models from Process Models	61
4.1	Motivation	61
4.2	Related Work	62
4.3	Approach	63
4.3.1	Infer Process Global Goals	64
4.3.2	Discover Intentional Fragments	64
4.3.3	Build a Goal Model	66
4.3.4	Refine the Goal Model	66
4.3.5	Tool Support	67
4.4	Conventions	69
4.5	Heuristics to Infer the Process Global Goals	69
4.5.1	H1: Global Goal From Start and End Events	69
4.5.2	H2: Avoid Goals From Exceptional End Events	71
4.5.3	H3: Avoid Goals From Exclusive Gateway	73
4.5.4	H4: Global Goal From the Process Name	74
4.5.5	H5: Global Goals From the First and Last Activities	74
4.6	Heuristics to Discover Intentional Fragments	74
4.6.1	H6: Intentional Fragments From Process Structure	75
4.6.2	H7: Intentional Fragments From Activities Annotations	77
4.6.3	H8: Intentional Fragments From Non-Functional Goal Categories	80
4.7	Heuristics to Build a Goal Model	80
4.7.1	H9: Refining the Global Goals	80
4.7.2	H10: Relation Between Intentional Fragments	82
4.8	Heuristics to Refine the Goal Model	82
4.8.1	H11: Case Refinement of an Intentional Fragment	83
4.8.2	H12: Milestone Refinement of an Intentional Fragment	83
4.9	Final Goal Model for the WellbeingUCL process	83
4.10	Summary	85
5	Evaluation: Electronic Healthcare Record for Bupa	87
5.1	Research Questions	87
5.2	Research Method: Action Research	88
5.2.1	What Is Action Research	88
5.2.2	Why Action Research	89
5.2.3	Action Research in Healthcare	90
5.2.4	Action Research in Software Engineering	90
5.3	Context and Purpose for Bupa COPD System	91
5.4	Diagnosis	92
5.4.1	Business and Clinical Objectives	92
5.4.2	Tolven Overview - Open Source EHR	94
5.4.3	Process Modelling In Bupa	95
5.4.4	Conclusion	97
5.5	Planning Action: Requirements Gathering for the System-to-Be	98
5.5.1	Goal Model for the As-Is System	99
5.5.2	Goal Model for the System-to-Be	110
5.5.3	Process Model for the System-to-Be	113
5.6	Taking Action: Extending Tolven for Bupa COPD Service	115
5.7	Evaluating Action: Impact of the new COPD System	116
5.7.1	RQ1: How Successful was the EHR Deployment	116
5.7.2	RQ2: Contributions of Our Techniques	117
5.7.3	RQ3: Project Impact in the BA community	118
5.8	Lessons Learned	120

6	Evaluation: Personal Health Record for Nuffield Health	122
6.1	Rationale	122
6.2	Research Questions	124
6.3	Context and Purpose for a Personal Health Record	124
6.4	Diagnosis	126
6.4.1	The Target User Experience	126
6.4.2	Architecture Overview - FHIR for Fitness and Lifestyle Data	127
6.4.3	Software Development Processes	129
6.5	Planning Action: Requirements Gathering	130
6.5.1	Customer Journey as a Process Model for the System-to-Be	130
6.5.2	Goal Model for the System-to-Be	132
6.5.3	Requirements for the Nuffield PHR Platform	139
6.6	Taking Action: FHIR FLI - a PHR System for Nuffield Health	141
6.7	Evaluating Action	143
6.7.1	RQ1: How Successful Was the FHIR FLI Platform	143
6.7.2	RQ2: Contributions of Our Techniques	144
6.7.3	RQ3: Fit with the Software Delivery Process	145
6.8	Lessons Learned	146
7	Conclusion	147
7.1	Summary of Contributions	147
7.1.1	Relating Goal Oriented Requirements Engineering and Process Modelling	147
7.1.2	Inferring Goal Models from Process Models	148
7.1.3	Electronic Healthcare Record for Bupa	149
7.1.4	Personal Health Record for Nuffield Health	149
7.2	Future Work	150
	Appendices	151
	Bibliography	151

List of Figures

2.1	BPMN Process Model	18
2.2	BPMN Meta-Model	19
2.3	Overview of KAOS models	21
2.4	KAOS Goal Model	23
2.5	ServeRX - KAOS Object Model	25
3.1	WellbeingUCL Mobile Unit and Data Collection Devices	33
3.2	KAOS Process View	34
3.3	WellbeingUCL - Participant Registration Process	35
3.4	Structural Constraints in KAOS	36
3.5	Representing Messages in KAOS	38
3.6	WellbeingUCL - Consent Declaration	39
3.7	WellbeingUCL - Process End Events	44
3.8	Consistency Check: Sequence of Activities	44
3.9	Consistency Check: Parallel Gateway	45
3.10	Consistency Check: Exclusive Gateway	45
3.11	Consistency Check: Exclusive Convergent Gateway	46
3.12	Consistency Check: Parallel Convergent Gateway	46
3.13	Extended KAOS Object Model	47
3.14	Process and Goal Models	51
3.15	WellbeingUCL - Collect Feedback	51
3.16	Goal Satisfaction Arguments	52
3.17	WellbeingUCL - Intentional Fragments	57
4.1	Problematic Goal Refinement from Process	62
4.2	Improved Goal Refinement from Process	63
4.3	Goal Inference Heuristics: Global Goals	64
4.4	Goal Inference Heuristics: Intentional Fragments	65
4.5	Goal Inference Heuristics: Refinement Links	66
4.6	Tool Support for Goal Inference Heuristics	68
4.7	H1: WellbeingUCL End Events	70
4.8	H2: WellbeingUCL Exceptional End Events	72
4.9	H6: Wellbeing UCL Structural Patterns	77
4.10	H7: WellbeingUCL Consent Declaration	78
4.11	H7: WellbeingUCL Consent Declaration (Complete)	79
4.12	H10: WellbeingUCL Goal Refinement	82
4.13	WellbeingUCL Goal Model	84
4.14	WellbeingUCL Additional Goals	84
5.1	Action Research Cycle	89
5.2	Somerset COPD Service - Care Pathway	92
5.3	Tolven Overview	94
5.4	Tolven Patient Record	95
5.5	COPD Service: Proposed Workflow	96
5.6	COPD Service: Assessment Process	96
5.7	Model Development	98
5.8	COPD Service Goal Model	99

5.9	COPD Enrolment Process Model - 2013	100
5.10	Enrolment Process Model - Heuristic H1	101
5.11	COPD Enrolment Process Model - Heuristic H9	101
5.12	Patient Enrolment - First Goal Model Refinement	102
5.13	COPD Enrolment Process Model - Heuristic H6	105
5.14	COPD Enrolment Process Model - Heuristic H10	107
5.15	Patient Enrolment - Second Goal Model Refinement	107
5.16	COPD Enrolment Process - Heuristic H7	108
5.17	Patient Enrolment - Third Goal Model Refinement	109
5.18	Patient Enrolment - Fourth Goal Model Refinement	110
5.19	Referral Process Model	110
5.20	Tolven Functionality	111
5.21	Consent in the System-To-Be	112
5.22	Eligibility Verification in the System-To-Be	112
5.23	Performance Reporting in the System-to-Be	113
5.24	New COPD Enrolment Process	114
5.25	Benefit Case for the Tolven Based EHR	115
5.26	Process Modelling Examples for BAs	119
5.27	Process Modelling Training Materials	119
5.28	Bupa Lean Approach	120
6.1	Post Physiotherapy Journey	125
6.2	Post Physiotherapy Journey (Detailed)	127
6.3	Information Architecture	128
6.4	Nuffield Health PHR - Example Process Model	131
6.5	Post Physiotherapy Goals	133
6.6	Post Physiotherapy - First Goal Model Refinement	134
6.7	Post Physiotherapy - Heuristic H6	135
6.8	Post Physiotherapy - Second Goal Model Refinement	138
6.9	Fitness Goals Setting in the System-to-Be	139
6.10	Fitness Goals Tracking in the System-to-Be	140
6.11	Prevent Customer Churn in the System-to-Be	141
6.12	FHIR FLI Architecture	142
6.13	Nuffield PHR - Responsibility Assignments	144

Chapter 1

Introduction

This chapter presents an overview of the thesis. We discuss the motivation for this research, with reference to the challenges faced by organisations wishing to deploy Electronic Healthcare Records. We then present the research objectives and our scientific contributions. We conclude by describing the structure of the whole thesis.

1.1 Research Motivation

This thesis investigates requirements engineering methods based on process modelling that business analysts can use for Electronic Healthcare Record (EHR) systems.

First, we focus on EHRs as these systems are increasingly being implemented in the healthcare industry, while still facing challenges related to user adoption rates and ability to demonstrate a positive impact on the quality of care [LD09].

Electronic Healthcare Records are defined by the International Organisation for Standardisation (ISO) as a repository of patient data in digital form, stored and exchanged securely, and accessible by multiple authorised users [ANS05]. EHRs are used across the whole spectrum of care, including GP or dental practices, community care services, care homes, specialised clinics or acute care hospitals. As such, they are used by health care professionals, including physicians, nurses, pharmacists, as well as by administrative staff and even patients. Depending on the clinical setting where an EHR is deployed, its functionality may include support for patient referral, medical history, diagnoses, tests, procedures, treatment, medication and discharge. To support day to day care, an EHR may also include support for medication administration, physical and mental health assessments, capturing vital signs and designing care plans [HSN08a]. We include in the scope of our research both Electronic Medical Records (records that care providers maintain about their patients) and Personal Health Records (records that consumers maintain about themselves). EHRs will have a number of functionalities that support "efficient and quality healthcare". From a software engineering perspective, we are interested in the requirements analysis and design of these functionalities.

Second, we focus on the role of business process models to inform the design and implementation of EHR systems. EHRs are meant to support coordination between the various actors involved with care delivery (software, hardware and humans). At the same time, EHRs have an essential role in supporting

and improving care pathways. The system functionalities directly influence how work tasks are carried out and may contribute to new levels of quality, efficiency, and work satisfaction [Ber01]. Alignment between IT and the clinical work processes will influence whether end users accept or reject the IT system, whether they introduce it into their day to day practice or work around it [LR05], [LEF01]. The difficulty of getting this alignment right is demonstrated by the many instances of underuse, resistance, work-arounds and even abandonment of EHR systems [HK10]. Ultimately, how well aligned an EHR system is to the organisational processes directly impacts the success of a project [HK10].

Third, we focus on the practice of Business Analysts (BAs) as they have a central role in gathering, formalising and communicating requirements to the technical teams. A core part of BAs toolset revolves around analysing business processes. In a healthcare context, this makes them well positioned to ensure that the EHR system is aligned to the organisational structure and user workflows [LD09]. However, they currently lack the tools to establish traceability between software requirements and user workflows.

Fourth, we are interested in how requirements engineering methods can incorporate process models and how we can increase their adoption in the BA community. BAs have difficulties adopting in their day to day practice requirements engineering methods developed in academia [FGZ15]. However, goal oriented requirements engineering provides a range of techniques useful in the health domain, such as conflict and obstacle analysis.

Within this context, our thesis specifically aims to help business analysts write better requirements specification for EHR systems. The quality of the requirements engineering concerns two aspects:

- alignment of the software product with business objectives, stakeholder needs and expectations [EEM95];
- measurement of impact traced back to supported processes and system goals[EEM95].

1.2 Common Challenges of EHR Deployments

From the literature describing EHR implementations we identify three problems that originate from the poor quality of initial requirements.

First, new workflow designs could impose responsibilities on agents (humans or software) that are proven unrealistic. If this is uncovered too late, there is an increased risk of undesired deviations from the prescribed workflow, once the system is in use [KWTK08]. To illustrate this case, we refer the reader to an evaluation report of ServeRX, an electronic dispensing system deployed in a London hospital [CAA⁺11]. The system required all the patients to wear a bracelet ID for identification purposes. Nurses had to scan this bracelet before dispensing any medicine in order to reduce the risk of administering the medication to the wrong patient. After the new system was deployed, nurses discovered that many patients were not wearing their bracelets. As a result, the nurses were not able to scan their IDs during drug rounds, and could not dispense the required medication. Workarounds soon emerged: for example, the nurses started keeping themselves the ID bracelets of the patients and scanning them when needed. This emerging workflow, different than the one envisioned when the system was deployed, increased the risk of incorrect patient identification since the nurses could accidentally scan the wrong bracelet. The

challenge for business analysts is identifying these unrealistic expectations early on, in order to prevent undesired workarounds.

Second, when a new software system alters the established workflow and the rationale for the changes is not clear, this will result in poor user adoption rates of the software [Hee06], [AGH⁺03],[LR07], [HK10],[LR06]. For example, an EHR implementation in an acute care hospital in North America required physicians to spend an estimated two hours of additional work every day to manage the patient records [LR05]. As a result, the system was never adopted on a large scale within the hospital, operating at only 25% of capacity. The challenge for business analysts is ensuring early on in the requirements elicitation phase that they capture the users' needs and preferences for the new system and workflow.

Third, for many Health IT projects, success is partly defined by whether the new system improves existing workflows. Without explicit links between software requirements and business processes, it is difficult to establish if processes have improved as a result of new software functionalities. Furthermore, process improvements have to be related to business goals. The challenge for business analysts is to establish traceability links between software functionalities, workflow fragments and business goals, to create a clear benefit case for stakeholders.

1.3 Research Objectives and Contributions

Our aim is to develop requirements engineering methods based on process modelling that business analysts can use to specify functionalities for EHR systems.

We hypothesise that process modelling, used in conjunction with goal oriented requirements engineering, will improve the quality of requirements specification and help address the challenges discussed in Section 1.2.

To test this hypothesis, this research has four main objectives. To address each objective, we present in this thesis four corresponding experiments. The findings from each experiment constitute the basis for our contributions to science.

1: Relating Goal Oriented Requirements Engineering and Process Modelling

Our first objective is to allow business analysts to evaluate and improve the alignment between software requirements, user workflows and business goals.

Our contribution is defining a novel process model view in the KAOS multi-paradigm requirements modelling language. The execution semantic of each process activity is defined by its pre and post-conditions, expressed over domain attributes. This approach has been developed after evaluating a number of alternative proposals.

We introduce and formally define the concept of Intentional Fragment that allows business analysts to describe and analyse fragments of a process related to a specific goal. The relation between process models, business goals and software requirements thus becomes explicit.

2: Inferring Goal Models from Process Models

Our second objective is to facilitate the adoption of goal oriented requirements engineering methods by the business analysts community.

Our contribution is to develop goal inference techniques that use process models as a starting point to help business analysts build correct goal models. We focus on guiding BAs towards eliciting correct goal refinements, as this aspect is particularly challenging for novice requirements engineers.

3: Electronic Healthcare Record for Bupa

Third, we evaluate the extended KAOS framework in a real industry setting, that is representative for clinical environments where EHRs are usually deployed.

We conducted the evaluation as an action research project in partnership with Bupa. We have been involved with the development and evaluation of the EHR system meant to support care delivery to more than 2,600 out-patients with a chronic condition.

We investigate how business analysts use the extended KAOS framework and whether it contributed to the success of the project. The extended KAOS framework and goal inference tactics have been used to inform the final software specification, guide the pathway redesign and clarify the business benefits. Working alongside business analysts, we reflect on how process modelling within a requirements engineering framework helps reduce the number of workarounds, improves the adoption rate and helps prove the benefit case for the project. All these are challenges we have previously identified in electronic healthcare projects.

Furthermore, we assess the impact and fit of the requirements elicitation process, in relation to the current work practices and the organisational structure in Bupa.

4: Personal Health Record for Nuffield Health

Fourth, we evaluate the extended KAOS framework in an industry setting representative of digital health initiatives - developing a new digital customer proposition based on an underlying EHR system.

We conducted the evaluation as an action research project in partnership with Nuffield Health. We have been involved in the delivery of new customer journeys for Nuffield Health customers, aiming to help people maintain or improve their health and wellbeing.

Our specific example is an exploratory project to support the recovery of physiotherapy customers, through ongoing gym sessions, support and advice. To deliver the prescribed customer journey, a number of different systems had to communicate and collaborate. We show how requirements engineering with process modelling was used to define integration requirements and responsibility assignments among the different systems, to support the target customer journey.

1.4 Structure of Thesis

The remaining of this thesis is structured as follows.

In **Chapter 2** we give an overview of existing goal and process modelling frameworks, with a focus on KAOS for goal modelling and BPMN for process modelling.

In **Chapter 3** we present our first experiment - the integration of goal and process modelling. This experiment directly addresses our first research objective. The chapter details our first contribution to science - the proposed extension to the KAOS framework. It covers the syntax and semantics of the new KAOS process modelling view and introduces the concept of Intentional Fragment.

In **Chapter 4** we present our second experiment - aimed at identifying goal inference heuristics

which business analyst may apply to a process model in order to identify goals and requirements. Our second contribution to science is a semi-structured method incorporating 12 such heuristics, which provides guidance to business analysts during goal oriented requirements elicitation.

In **Chapter 5** we present our third experiment, an action research project aimed at evaluating the process based goal elicitation method in an industry setting representative for the clinical domain. The KAOS method extended with process modelling has been used to design an EHR system for a clinical service offered by Bupa. The findings related to the practical use of a process based goal oriented requirements engineering method in the delivery of a clinical EHR system represent our third contribution to science.

In **Chapter 6** we present our fourth experiment, aimed at evaluating our techniques in an industry setting representative for the digital health domain. This chapter presents our findings related to the use of a process based goal oriented requirements engineering method in the delivery of a consumer EHR system for Nuffield Health- this constitutes our fourth contribution to science.

In **Chapter 7** we present a summary of the research and discuss potential directions for future work.

Chapter 2

Background

In this chapter we review two methodologies that have been used to support the design and development of electronic healthcare records. Section 2.1 examines frameworks for process modelling, with a focus on the Business Process Modelling Notation, a de-facto standard used in the industry. Section 2.2 presents KAOS, a goal oriented requirements engineering framework developed in academia. Finally, Section 2.3 compares the meta-models of the two frameworks, in order to inform our integration proposal. Establishing a framework that integrates both goal modelling and process modelling is our first objective - we present our proposal in the next chapter.

2.1 Process Modelling with BPMN

Business process models provide a natural way to describe real-world processes to be supported by software-intensive systems. Process modelling is also used in the context of performance measurement, workflow improvement projects, change management and software requirements specification [IGRR09].

A business process is a collection of interrelated activities, initiated in response to a triggering event, which achieves a specific, discrete result for the customer and other stakeholders of the process [SM01].

2.1.1 Business Process Modelling Notation

The ISO standard for process modelling is the Business Process Modelling Notation (BPMN), currently at version 2.0 [OMG11], [ISO13]. BPMN takes inspiration from traditional swim-lane charts. It contains graphical elements that can be used to describe the steps in a business process from start to end (activities, decision gateways, start and end events etc). These graphical elements, collectively known as flow elements, are connected by sequence flows, following a set of structural constraints. The execution semantic of BPMN is described informally using the concept of a token that passes through the elements of the process and along the flows connecting them. The behaviour of the flow elements is defined by describing how they interact with a token as it reaches them.

A small example of a BPMN process model for a drug dispensing system is given in Figure 2.1, based on information about a ServerRX deployment in a UK hospital [CAA⁺11]. Once all medication has been prepared for a given drug round, the nurse takes the medication trolley and visits each patient in the ward. The nurse scans the barcode on each patient's wristband and this triggers the corresponding

patient drawer in the trolley. Once the drawer is opened, the nurse can administer the medication. The nurse confirms that each patient has taken his medicine by using a touch-sensitive screen on the trolley.

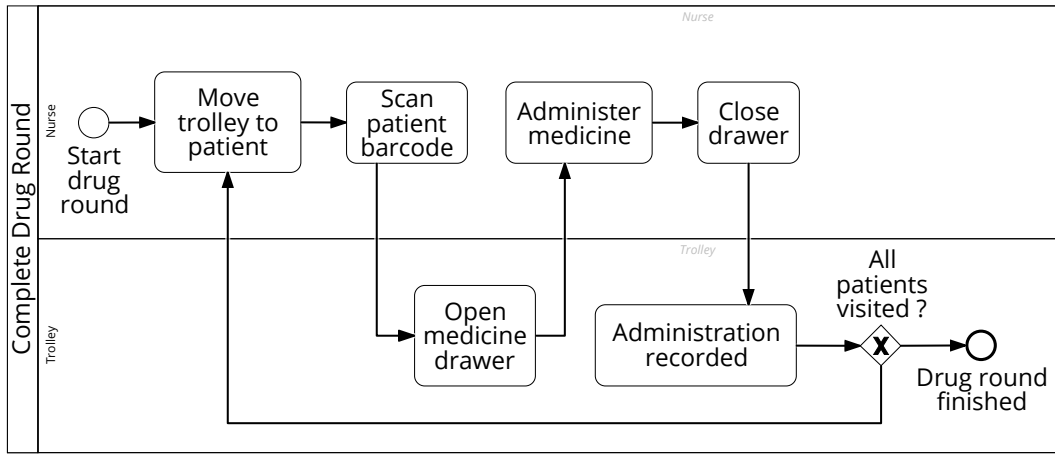


Figure 2.1: BPMN Process Model for Drug Administration

Figure 2.2 shows the meta-model for the core BPMN elements. The **FlowElement** is the abstract super class for all elements that can appear in a **Process** model. The **ActionNode** is a specialisation of the Flow Element used to provide a single element as the source and target for **Sequence Flow** associations. Only the **Gateway**, **Activity** and **Event** elements can connect to sequence flows and thus, these elements are the only ones that are sub-classes of the ActionNode. Similarly, the **InteractionNode** is used to provide a single element as the source and target for **Message Flow** associations. Only the **Pool**, **Participant**, **Activity**, and **Event** elements can connect to message flows. An **Activity** is an atomic process step executed by either a system (automated) or humans (manual). An **Event** represents a specific occurrence during the course of a process. BPMN has restricted the use of events to include only those types of events that will affect the sequence or timing of activities of a process. **Gateways** are used to control how the process executes along different sequence flows as they converge and diverge within a process.

2.1.2 BPMN Semantic

The BPMN standard only describes a token based semantic - tokens are created by start events when an instance of the process starts execution, they travel through sequence flows and are eventually consumed by process end events. The flow of tokens is controlled by gateways, and when a token reaches an activity, the activity is executed. This type of semantic does not describe the effects a process execution has in the environment.

To add rigorous execution semantics to BPMN models, BPMN has been mapped to Business Process Execution Language for Web Services (WS-BPEL) [Whi04] or YAWL [DDDGB08]. BPMN semantics have also been expressed using Petri nets [DDO08], Calculus of Orchestration of Web Services (COWS) [PQZ08] and Communicating Sequential Processes [WG08].

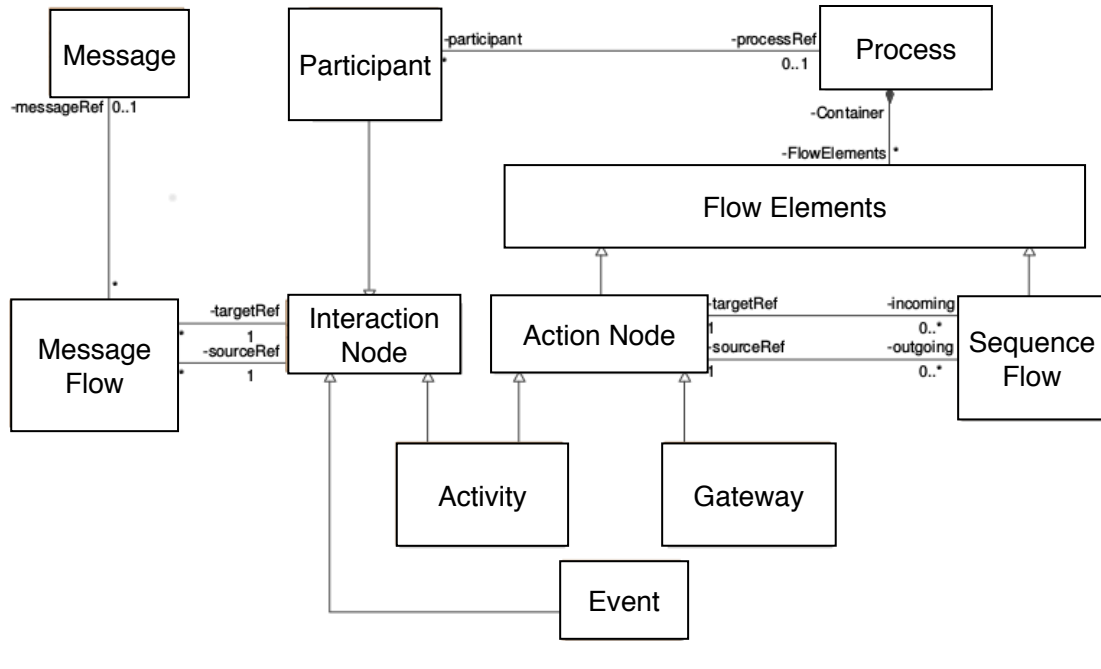


Figure 2.2: BPMN Meta-Model

2.1.3 Other Process Modelling Languages

BPMN is not the only process modelling language used in software engineering. Other process modelling languages include Little-JIL [CLM⁺00], Guarded High Level Message Charts [DLRVL09] and Event Driven Process Chains [VdA99].

Little-JIL is an executable, high-level language with a formal syntax. Using a graphical tool, processes can be expressed in terms of their steps (units of work) and responsibility assignments of agents to steps. The model can then be translated to a state transition system, which allows for model checking. This technique was successfully used to identify potentially dangerous situations in existing medical processes or to highlight processes that had an incomplete specification [CAH⁺08].

Guarded high-level Message Sequence Charts have also been used to model medical processes. A Message Sequence Chart is a commonly used way to capture scenarios in a system, with the benefit that stakeholders find it easier to follow and contribute to the modelling process. To make more complex scenarios easier to comprehend, several MSCs are composed into high-level Message Sequence Charts. With additional user input, Labelled Transition System models can be generated from the MSCs [UKM03]. Interaction scenarios produced from the LTS models can be used to mine for additional goals, or to validate safety goals [DLVL06].

2.1.4 OpenEHR - Domain Specific Languages for Modelling Care Pathways

OpenEHR [Bea02] is an open platform for electronic healthcare systems which also has incipient support for modelling sequences of activities (as part of a care plan) and corresponding clinical goals [BB⁺05].

We review the OpenEHR approach towards modelling care pathways in order to identify any design choices specific to the healthcare domain, which would inform our proposal for process modelling.

An extension to the OpenEHR Information Model has been in development since 2017 [Tas], to enhance support for representing lists of tasks, and building on that, modelling individual care plans and care pathways. In the OpenEHR proposed model, a care plan for a patient is modelled as a list of planned tasks, and each task specifies an action to be performed by human or software actors involved in the care provision.

The proposed model defines clinical workflows as a progression through three phases: Order, Planned Task, and (performed) Task. **Orders** are prescriptive - they capture expectations that certain actions will be performed in certain conditions. For example, a patient should take a medicine every morning for 7 days. A **Planned Task** is a small unit of planned work that typically corresponds to the finest level of clinical responsibility or a single item in a care protocol, to be performed by an agent at a specific time in the future [Tas]. The concept of **Performed Tasks** is represented in OpenEHR as **Action** or **Observation**. They represent events that have taken place in the environment, like administering a medicine or measuring the blood pressure of a patient. It is then possible to check conformance between planned tasks (i.e. actions to be performed) and performed tasks (i.e. actions that have been performed). Each planned task may also reference one order. A **Task List** is a logical list of planned tasks whose execution is intended to achieve completion of a goal, for example deliver a chemotherapy treatment over the course of several sessions. Task Lists can also consist of planned tasks that define administrative work (specialising the OpenEHR **Admin Entry** concept [Tas]).

In OpenEHR, the nature of workflows is prescriptive. There is a strong focus on enabling practitioners to demonstrate that the care protocol for a given patient has been followed. We consider the OpenEHR process modelling support to be closer to model checking techniques, aimed at demonstrating compliance. Specifically, within the OpenEHR framework, one can reason whether the process execution matches the process definition. However, our interest is in the early analysis phase of projects, where neither the process, nor the software functionalities are fixed or fully specified. We agree with the OpenEHR direction of identifying a prescriptive layer (to capture activities that *have* to take place), but we also acknowledge that many projects have a process re-engineering component, and this means the process models should be more expressive. For example, the models should allow one to reason if an activity can be removed or replaced while keeping the business goals satisfied.

2.2 Requirements Engineering with KAOS

Goal-oriented requirements engineering (GORE) frameworks are used for eliciting, elaborating, structuring, specifying, analysing, negotiating, documenting, and modifying requirements [VL01], [VL09].

KAOS is one of the most important GORE frameworks. It is aimed at supporting the whole process of requirements elaboration - from the high-level goals to be achieved to the requirements, objects and operations to be assigned to the various agents in the system. A KAOS goal model captures the **goals**, **requirements** and **goal refinement links** identified for the system; **objects**, **agents** and **operations** are described in separate models. KAOS also defines inter-model consistency rules, in order to avoid ambiguities or contradictions between the different views over the system.

Each KAOS view has a two layer structure: a graphical layer for modelling concepts (such as goals,

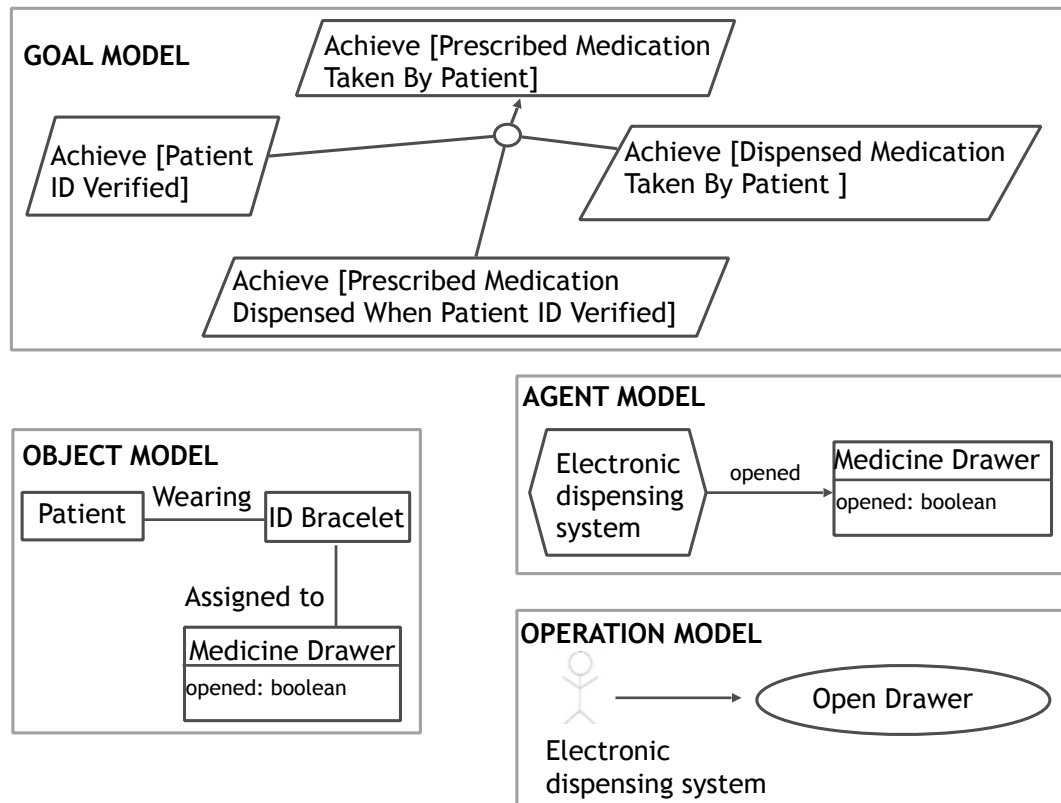


Figure 2.3: Overview of KAOS models

objects or agents as shown in Figure 2.3) and an optional assertion layer for specifying these concepts formally. Objects in the object model have attributes and relationships which are referenced in the assertion layer.

KAOS uses a state based semantic, where models are interpreted over sequences of system states observed at a fixed time rate.

Def State variable: State variables correspond to object attributes and relationships in the application domain object model. Ex: registrationFormPrinted, consentTaken;

Def System state: A system state is a mapping that assigns a value to each state variable. Ex: registrationFormPrinted = true; consentTaken = false;

In KAOS, consecutive states in a trace are always separated by a single time unit. The time unit corresponds to some arbitrarily chosen smallest possible time unit for the application domain. Zero, one or more state variables may change their values between two consecutive states.

Def RT-LTL Assertions: KAOS real-time linear temporal logic assertions are formed from state variables and temporal operators:

- | | |
|----------------------------|--------------------------|
| ○(in the next state) | ●(in the previous state) |
| ◇(some time in the future) | ◆(some time in the past) |
| □(always in the future) | ■(always in the past) |

Bounded versions of the above temporal operators are also used, such as:

$\diamond_{\leq d}$ (some time in the future within deadline d)

$\square_{\leq d}$ (always in the future up to deadline d)

Finally, we will use the following standard logical connectives: \wedge (and), \vee (or), \neg (not), \rightarrow (implies), \leftrightarrow (equivalent).

2.2.1 The Goal Model

A goal is an objective the system should meet; it captures a set of desired behaviours of the system. Goals are classified according to the category of requirements they capture. Functional goals result in functional requirements. For example, **Satisfaction Goals** are functional goals concerned with satisfying agent requests; **Information Goals** are goals concerned with keeping agents informed about object states. Likewise, non-functional goals result in non-functional requirements. For example, **Accuracy Goals** are non-functional goals concerned with maintaining the consistency between the state of objects in the environment and the state of their representation in the software; other sub-categories include safety, security and performance goals.

Every goal has a name and a definition, and may optionally have a formal specification. For example, the main functional goal for the electronic dispensing system is ensuring each patient takes her prescribed medication:

Goal	<i>Achieve</i> [Prescribed Medication Taken By Patient]
Def	Given a patient has been prescribed a medication to take during a drug round, When the drug round occurs, Then the patient takes the prescribed medication during that round.
FormalDef	$\forall p : Patient, m : Medication, r : DrugRound$ $Prescribed(p, m, r) \wedge r.occurs \Rightarrow \diamond HasTaken(p, m, r)$

The declaration part of this specification introduces a goal named `PrescribedMedicationTaken-ByPatient`, stating a target property that should eventually hold, referring to objects such as `Patient` or `Medication`. The optional assertion in the specification formally defines the goal using state variables (e.g. the *occurs* attribute of the `DrugRound` object) and linear temporal logic operators.

Goals in KAOS are classified according to the pattern of temporal behaviour they capture.

Achieve goals: given a condition C that holds in the current system state, eventually in the future the target condition T will hold: $C \Rightarrow \diamond T$

Cease goals: given a condition C that holds in the current system state, eventually in the future that target condition T will not hold: $C \Rightarrow \diamond \neg T$

Maintain goals: given a condition C that holds in the current system state, target condition T will hold always in the future: $C \Rightarrow \square T$

Avoid goals: given a condition C that holds in the current systems state, target condition T will never hold in the future: $C \Rightarrow \square \neg T$

An example of goal of type Avoid in the electronic dispensing system is the safety goal requiring

that patients shall only get the medication prescribed to them.

Goal	<i>Avoid</i> [Wrong Medication Taken By Patient]
Def	Given a patient has taken some medication during a drug round Then the patient has been prescribed that medication
FormalDef	$\forall p : Patient, m : Medication, r : DrugRound$ $HasTaken(p, m, r) \Rightarrow Prescribed(p, m, r)$

Figure 2.4 shows the goal model for the hospital ward in which the ServeRx dispensing system was deployed. Although limited in scope (it does not show the goals related to electronic prescribing of medicines for example), it is a good illustration of the core elements of KAOS goal models.

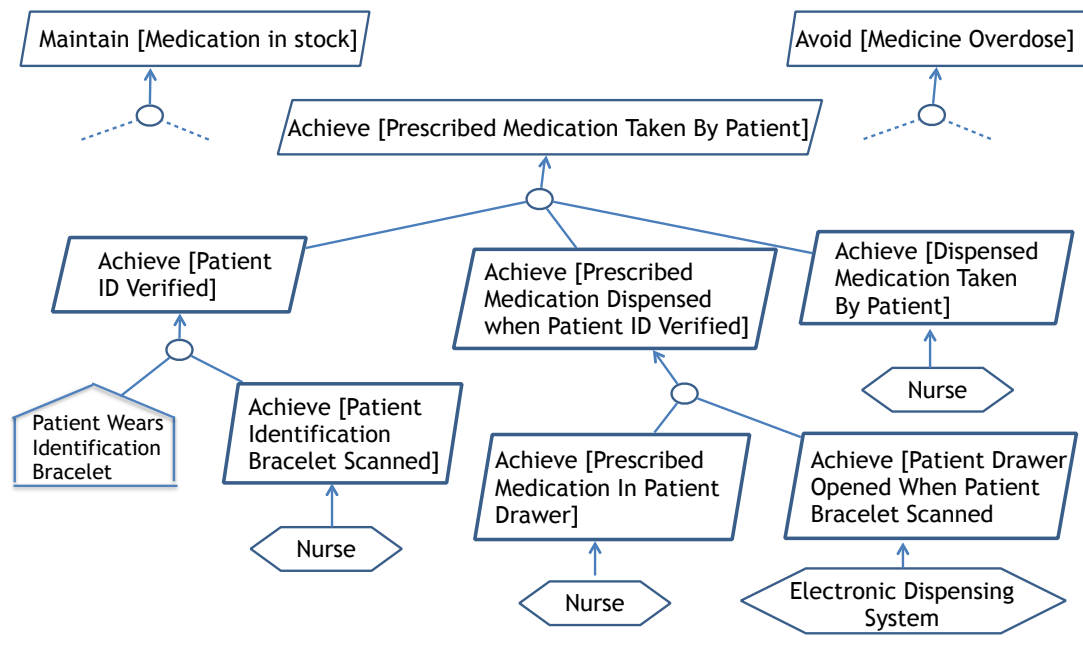


Figure 2.4: KAOS Goal Model for a Drug Administration System

In a goal model, **AND-refinement links** relate a goal to a set of subgoals - satisfying all subgoals in the refinement is a sufficient condition to ensure the goal itself is satisfied. **OR-refinement links** relate a goal to an alternative set of refinements - satisfying one of the refinements is a sufficient condition to satisfy the goal. Goals are refined until they are assignable to individual agents. A **Requirement** is a goal assigned to an agent in the software to be. In the electronic dispensing system, opening each medication drawer when the corresponding bracelet is scanned is an example of a requirement.

Goal	<i>Achieve</i> [PatientDrawerOpenedWhenPatientBraceletScanned]
Def	Given the patient is wearing an identification bracelet and the patient has an assigned drawer in the cabinet, When the nurse scans the patient identification bracelet, Then the dispensing system opens the drawer associated with the patient ID.
FormalDef	$\forall p : Patient, b : IDBracelet, d : Drawer$ $Wearing(p, b) \wedge AssignedTo(b, d) \wedge Scanned(b) \Rightarrow \circ Opened(d)$
Resp	Electronic Dispensing System

An **Expectation** is a goal assigned to an agent in the environment. Unlike requirements, expectations can not be enforced in general.

A **Domain Property** is a descriptive assertion about objects in the environment which holds independently of the software to-be. It may be a domain invariant or a hypothesis. A **Domain Invariant** is a property known to hold in every state of some domain object, e.g. physical laws, regulations etc. A **Hypothesis** is a property about some domain object supposed to hold and be used when arguing about the completeness of a goal refinement. Our goal model also contains one domain property, the fact that patients wear identification bracelets which are used to verify their identity before dispensing any medication to them.

Starting from this initial goal model, KAOS offers systematic techniques for model elaboration using goal based refinement patterns [DVL96], obstacle analysis [VLL98] or reasoning about partial goal satisfaction [LVL04].

Obstacle analysis is used in KAOS to try and identify assumptions that are likely to be violated in real life. Once conditions that could prevent the satisfaction of a goal are identified, requirements engineers will refine the goal model to add mitigation strategies. Overall, methodical obstacle analysis will lead to more reliable software. Reviewing the goal definition for opening the medication drawer, one could ask what happens if the bracelet is scanned but the command to open the drawer is not sent. In other words, a situation has been identified where a failure of the scanner would be an obstacle towards goal satisfaction. Requirements engineers could introduce as a mitigation strategy an alternative way of opening the drawer, which does not rely on the scanner.

To reason about partial goal satisfaction in KAOS, a goal should be annotated with domain specific attributes. *Quality variables* define a quantity that can be measured in the domain, over a set of instances (a sample space). *Objective functions* declare whether the quality variables should be minimised or maximised. The goal is satisfied to the degree in which the quality variables meet the target values. Using our drug round example, the goal could be annotated as follows:

Goal	<i>Achieve</i> [PatientDrawerOpenedWhenPatientBraceletScanned]
FormalDef	$\forall p : Patient, b : IDBracelet, d : Drawer$ $Wearing(p, b) \wedge AssignedTo(b, d) \wedge Scanned(b) \Rightarrow \circ Opened(d)$
Objective function	Maximise openRate
target	98%
Quality variable	openRate
Sample space	Completed bracelet scans
Definition	The rate between successful drawer openings and total bracelet scans

2.2.2 The Object Model

The object model defines the domain entities, relationships and attributes that are relevant to goal formulations. Entities are autonomous and passive objects of the system.

Figure 2.5 shows the corresponding object model for the ServeRX system, consistent with the goal definitions we have given before. For example, the Medication type referenced in the formal definition

of the goal Achieve [Prescribed Medication Taken By Patient] is an example of a KAOS entity. The predicate wearing(patient,bracelet) captures a relationship between Patient and Bracelet objects.

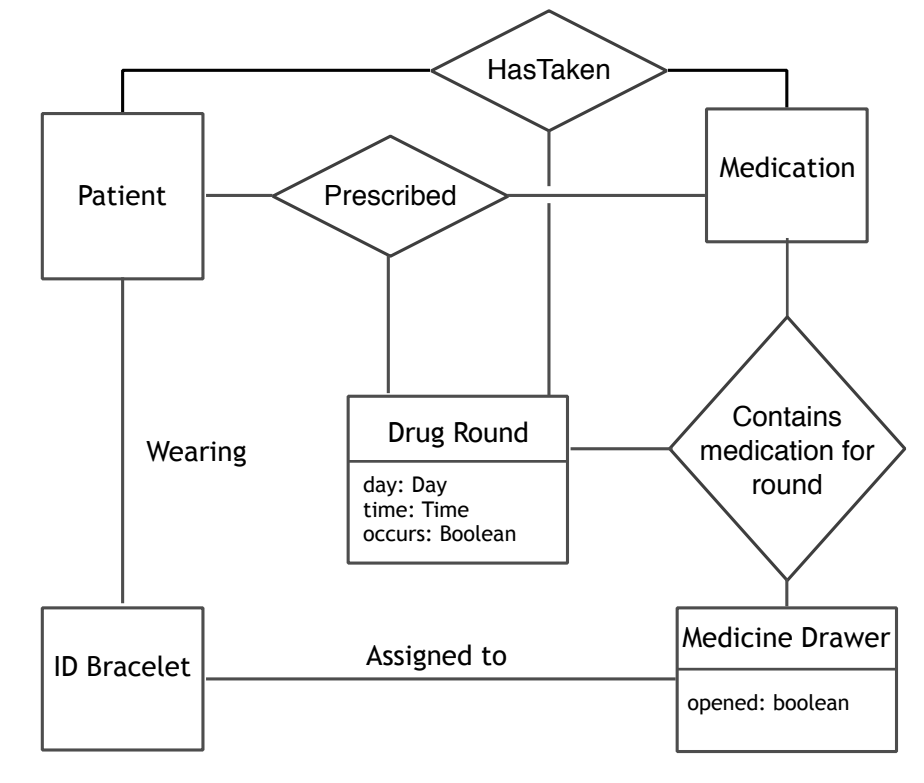


Figure 2.5: ServeRX - KAOS Object Model

2.2.3 The Agent Model

An **Agent** is an active object which plays a role towards achieving a goal by controlling specific object behaviours. It may be a software agent in the system-to-be or an environment agent (sensors, humans, organizational units etc.).

The agent model defines the responsibilities and interfaces of the various agents forming the composite system (humans, devices or software). The model allows requirements engineers to capture the distribution of responsibilities among the active elements of the system under consideration. Agents may collaborate to achieve high level goals, but they are individually responsible for leaf goals (requirements and assumptions).

If an agent is responsible for a goal, it should have sufficient monitoring and control capabilities over the objects of the system to realise the goal. An agent monitors or controls an object if the states of the object are directly observable or controllable by it.

Consider for example the goal Achieve [Patient Drawer Opened when Patient Bracelet Scanned] defined previously. There is one drawer for each patient and whenever a patient bracelet is scanned, the corresponding drawer should open. As the Electronic Dispensing System is responsible for the goal, it has to control the opened attribute of the drawer.

2.2.4 The Operation Model

An operation is an input-output relation over components of the object model. Operations are declared by signatures over objects and specified by pre-, post- and trigger conditions. These conditions are defined as follows [LVL04]:

- A domain pre-condition characterises the states before any application of the operation;
- A domain post-condition defines a relation between states before and after applications of the operation;
- Required pre-conditions define those states in which the operation is allowed to be applied;
- Required trigger conditions define those states in which the operation is obliged to be immediately applied provided the domain pre-condition is true;
- Required post-conditions define conditions that applications of the operation must satisfy.

The distinction between domain pre/post-conditions and required pre/trigger/post-conditions is that the former capture elementary state transitions defined by operation applications in the domain, while the latter capture additional strengthenings to ensure that the goals are met [LVL04].

As an example, we give below the complete definition for the operation *OpenDrawer*. We note that when it comes to building goal satisfaction arguments, which we discuss in the next section, the signature of each operation is interpreted taking into account additional domain assumptions. To ensure the medicine is administered, there is an expectation for a human agent to fill the drawer with the correct medicine, so that once the drawer opens, the medicine may be dispensed.

Operation	OpenDrawer
Input	d: <i>Medicine Drawer</i> b: <i>ID Bracelet</i>
Output	d: <i>Medicine Drawer</i>
DomPre	\neg Opened(d)
DomPost	Opened(d)
ReqTrig for Achieve [Patient Drawer Opened When Patient Bracelet Scanned]	Scanned (b)

2.2.5 Goal Operationalisation

The formal semantics of the operation model is defined by mapping every construct of the operational language into temporal logic assertions [LVL02]. For every operation *op* in the operation model with logical variables arg_1, \dots, arg_n as arguments and res_1, \dots, res_n as results, there is a temporal logic predicate denoted by $[| op |](arg_1, \dots, arg_n, res_1, \dots, res_n)$. This predicate expresses that the operation is currently being applied on the given arguments and results. Similarly, the pre and post-conditions of an operation are formalised as temporal logic state predicates.

For every operation *op* in the operation model, the predicate $[| op |]$ is defined as follows:

$$[| op |](arg_1, \dots, arg_n, res_1, \dots, res_n) \Leftrightarrow \text{DomPre}(op) \wedge \circ \text{DomPost}(op),$$

where $\text{DomPre}(\text{op})$ represents the domain pre-conditions of the operation and $\text{DomPost}(\text{op})$ the domain post-conditions.

This definition states that every application of an operation implies that the operation's domain pre-condition is satisfied in the state before the application and the domain post-condition is satisfied in the state after the application. In other words, an operation defines a relation over two consecutive system states whose distance is one time unit. Conversely, every state transition that satisfies the domain pre and post-conditions of an operation corresponds to an application of that operation.

The formal semantics of the required pre, trigger and post-conditions of an operation op are defined in relation to a temporal logic predicate for each condition R , denoted $\llbracket R \rrbracket$. The predicate is defined as:

if $R \in \text{ReqPre}(\text{op})$ **then** $\llbracket R \rrbracket =_{def} (\forall^*) \llbracket \text{op} \rrbracket \Rightarrow R$

if $R \in \text{Trig}(\text{op})$ **then** $\llbracket R \rrbracket =_{def} (\forall^*) R \wedge \text{DomPre}(\text{op}) \Rightarrow \llbracket \text{op} \rrbracket$

if $R \in \text{ReqPost}(\text{op})$ **then** $\llbracket R \rrbracket =_{def} (\forall^*) \llbracket \text{op} \rrbracket \Rightarrow \circ R$

In the above definition we use the standard notation $(\forall^*)P$ for the universal closure of P .

Def Goal Operationalisation

A set R_1, \dots, R_n of required conditions on operations in the operational model correctly operationalise a goal G in the goal model iff the following conditions hold:

$\llbracket R_1 \rrbracket, \dots, \llbracket R_n \rrbracket \models G$ (completeness)

$\llbracket R_1 \rrbracket, \dots, \llbracket R_n \rrbracket \not\models \text{false}$ (consistency)

$G \models \llbracket R_1 \rrbracket, \dots, \llbracket R_n \rrbracket$ (minimality)

Def Operationalisation Pattern

An operationalisation pattern is an abstract AND-operationalisation link between a goal specification pattern in RL-LTL and a set of required pre-, trigger and post-condition specification patterns. The pattern is formally proven correct with respect to the completeness, consistency and minimality criteria once and for all. Goal operationalisations that follow one of these patterns are already proven to be correct. For example, a bounded Achieve goal of the form $C \Rightarrow \diamond_{\leq d} T$, where T represents a target state and C the current state, is operationalised by the following operation:

Operation	Op
DomPre	$\neg T$
DomPost	T
ReqTrig	$\neg T S_{=d-1}(C \wedge \neg T)$

The trigger for the operation states that the operation *must* be applied when T has remained false since C was true $d-1$ time units ago without T being true.

2.2.6 Consistency Rules

The various KAOS models must satisfy a set of inter-model consistency rules. For example, one consistency rule that connects the agent, the operation and the goal model states that if an agent is responsible for a goal, that agent should perform all the operations that operationalise that goal. Similarly, if an object is referenced in a goal under the responsibility of an agent, one or more attributes of the object

must be monitored or controlled by this agent. In Chapter 3 we will define new inter-model consistency rules relating the new process modelling view to the other views in KAOS.

2.3 Meta-Model Comparison: BPMN and KAOS

OpenEHR includes incipient support for modelling workflows. This is driven by the clinical requirements of being able to observe how a specific case is being managed (i.e. are the prescribed actions for a patient actually executed). The set of capabilities are under active development, but they are not mature enough to be used as a fully fledged process modelling framework. As such, we propose to focus on KAOS and BPMN as the more established frameworks for requirements analysis and designing a new software system.

After reviewing the KAOS meta-model and the BPMN standard meta-model, we present a partial mapping of their core elements. This mapping, given below, will constitute the basis for our approach towards integrating the two models into a single framework.

BPMN Concept	KAOS Concept	Definition
Participant	Agent	An active entity capable of performing actions
Activity	-	Work carried out by a participant during the process
-	Operation	Input - output relation over objects in the system
Event	Event	An instantaneous change in the system state
-	Goal	A prescriptive assertion capturing some objective

Some of the concepts used in both models can be considered synonymous. For example, the KAOS concept of **agent** and the BPMN concept of **participant** both denote an active entity, which has responsibilities and acts in order to meet them. In BPMN, **activities** express work that needs to be carried out. They have duration and can be decomposed into finer grained activities. There is no direct equivalent in the standard KAOS meta-model. Finally, BPMN has no representation for KAOS **goals**. Business analysts are expected to express business objectives and rules outside BPMN.

Chapter 3

Relating Goal Oriented Requirements Engineering and Process Modelling

In this chapter we extend the KAOS framework for requirements engineering to include process modelling. Section 3.1 presents the motivation for relating process and goal modelling. Section 3.2 offers an overview of related work, while Section 3.3 specifically discusses the options we considered for relating KAOS and BPMN. Section 3.4 introduces our first industry project - a data collection platform for lifestyle data. Section 3.5 presents the meta-model and syntax of the KAOS process view. Section 3.6 introduces the most important departure from standard BPMN - annotations for activities, gateways and events. Section 3.7, Section 3.8 and Section 3.9 discuss the relation between the new process view and the KAOS object, operational and goal model respectively. Section 3.10 introduces a new concept - Intentional Fragments - to help business analysts relate fragments of a process to individual goals. Section 3.11 discusses how intentional fragments can be used to align the process and the goal model, even if the models are not fully specified.

3.1 Motivation

Our overarching objective is to allow business analysts to evaluate and improve the alignment between the software specification and the prescribed workflows, in the context of EHR systems. In Chapter 1 we have discussed why this is an important challenge for organisations deploying EHR systems and Chapter 2 reviews two frameworks that could help address this challenge. We have shown that BPMN and KAOS operate with complementary concepts and together the two frameworks will describe a system in sufficient detail for our objective.

As such, we propose that relating an established goal oriented requirements engineering framework such as KAOS with a widely used process modelling framework like BPMN will allow business analysts to reason about and improve the alignment between software systems, user workflows and business goals.

In relating the two frameworks, we will also address four specific modelling concerns.

First, BPMN does not distinguish between prescriptive and descriptive models. This lack of expressive power means a business analyst can not use a process model alone to derive software requirements.

For example, if in a process model an activity is followed by another activity, the model itself does not clarify whether this sequence is a requirement or simply a description of the current state of affairs.

Second, business analysts can not rigorously analyse if process models are aligned to known business goals, since BPMN does not provide support to model business goals. The rationale of the process activities in a BPMN model is not explicit [IRRG09], [dIVSP08].

Third, BPMN users lack adequate means for depicting business rules or organisational policies pertaining to the execution of the business process [RIRG10]. Using BPMN models alone, business analysts can not capture whether a new or redesigned workflow, observed over a number of instances, satisfies the organisational goals to a greater degree than the original workflow.

Fourth, KAOS has so far lacked a way to incorporate business processes in the requirements elicitation phase. In the requirements engineering community, scenarios (sequences of interaction steps between the intended software and other agents in the system) have been used to inform or explain features or functionalities during requirements elicitation [SW11], [VLW98]. Compared to scenarios, process models offer a more comprehensive description of organisational workflows. They describe how agents carry out work to deliver on many distinct, potentially conflicting, business outcomes. Process models are well suited to support goal refinement, and they represent a relevant knowledge source to inform software requirements.

3.2 Related Work

The need to relate the software specification to the business process models has been discussed both in the business process management community and in the software engineering community [PDRS13], [LR12], [Car13],[BAM10],[MK08].

First, early work has established the core relationship between process and goal models: business processes constitute the way to satisfy strategic business goals. Goals also shape business processes, by informing business improvement initiatives which often result in changes to business processes [KL99]. Process models will also contribute to non-functional (“non value-added”) goals [KK97]. The focus here is on the equivalence between leaf goals and process activities. However, this equivalence is not backed by a shared semantic. These early works do not detail how the execution semantic of process models may be related to the formalism used to specify business goals.

Second, a large body of work investigates how BPMN process models can be converted into executable processes, which can be fed into a workflow engine and automated using a composition of web-services. For the purpose of specifying the executable process, the Execution Language for Web Services (BPEL) [ACD⁺03] is commonly used [ZM05], [RM06]. This transformation can be direct from BPMN to BPEL [ODTHVdA06], [Whi05]: structural patterns (sequence, flow, loop) in the BPMN models are mapped onto BPEL constructs. The transformation could also involve an intermediate model that keeps track of the transformations applied on the business process in order to generate the web-service composition schema (equivalent to an executable process defined in BPEL) [BBR11]. Business analysts are responsible for selecting and applying the transformations, for the specific purpose of obtaining a specification that can be entered into a workflow management system. This area of research is focused

on automating a given process model. The focus here is on the equivalence between process activities and web-services as discrete units of software functionality. The existing work does not consider higher level goals, alternative goal refinements, or how the process model itself may be changed or improved. As we have argued before, in complex socio-technical systems the BPMN process models are often incomplete, ambiguous or not representative of the true business needs. They can not be fully automated, as the human agents play a central role in delivering business value. Even more, the business process itself may not be sufficient to satisfy the business goals, and automating it will not deliver the value expected by stakeholders. As such, our own contributions focus on eliciting the information embedded into the process models to shape the design and functionality of the software system, in combination with other requirements elicitation techniques. Our aim is to ensure the final software supports the human agents towards achieving the high level business goals.

Third, there exist approaches to derive process models from a fully specified goal model. For example, [DP09] proposes a way to derive a Business Process Modelling Ontology (BPMO) diagram from a goal model defined in Formal Tropos [FLM⁺04], a temporal specification language inspired by KAOS. [LYM07] also proposes a method to derive an executable process model from a given goal model. These works are focused on deriving a final executable process specification, assuming a correct and complete goal model already exists. Our approach, on the other hand, allows for both the goal and process models to be iteratively refined until alignment is reached. Ultimately, we aim to support the early requirements elicitation process when design decisions are made through alternative goal refinements. As such, our focus is on how a correct goal model can be derived in the first place, using information implicit in the process models.

Fourth, there are a number of proposals that support formal model checking, to verify if the execution semantic of a process satisfies the formal specification of a goal or a business constraint. For example, GOALBPM [KG06] assigns each process activity an effect annotation. The annotation describes the system state after an activity completes, in the same formalism as the KAOS goals. Given an execution path, the effects of each activity on the path accumulate and allow a business analyst to reason whether the path satisfies the goal or not. BPMN-Q [ADW08] defines a query language that allows business analysts to verify if certain business rules hold in a process, specifically around the ordering of activities. Specifically, a temporal logic formula (derived from the query) is checked against a finite state machine (derived from the part of process model relevant to answer the query). The focus of these works is on formal verification arguments, while our aim is to integrate process models into the broader requirements elicitation process - we are interested in how process models can assist business analyst in uncovering new goals and requirements, and how the goal model can in turn inform changes to the process.

3.3 Approach

To relate process models and goal models, the first option considered was extending BPMN with a prescriptive layer, where business goals could be formalised and attached to process activities. However, this was discarded because our aim is to support the complete requirements elicitation process. KAOS already offers a wide range of techniques, as it is a well established goal oriented requirements engineering framework. It is thus preferable we introduce process modelling into the KAOS framework.

The second option considered was to represent each process activity from a given BPMN model as an object in the KAOS object model for the system under consideration. Agents could then be assigned goals related to the process execution. For example, agents would be responsible to start and complete activities, and would also have to ensure that process activities follow the prescribed execution order. This approach was discarded because it simply assumes the process model is always prescriptive, correct and complete. Further more, this type of goal definition would provide little insight into the rationale for executing an activity, in contradiction with our overarching goal oriented approach.

Our chosen approach, considering our goal of relating goal and process models, and the specific challenges with each modelling framework, is to extend the KAOS framework by adding a KAOS process view. On one hand, this is a more complex undertaking, since we are making changes to the KAOS meta-model and we face the risk of introducing inconsistencies between the new process view and existing KAOS views and techniques. On the other hand, establishing process models as a new type of KAOS model offers business analysts a complete and consistent view over a system of interest, maximising the information available to them while eliciting requirements for a system-to-be.

3.4 WellbeingUCL Project

To illustrate our proposal for KAOS based process modelling, we refer to requirements analysis, design and software development work we have carried out for the WellbeingUCL project. WellbeingUCL was designed as a proof of concept for a new way of conducting wellbeing studies that favours a holistic approach towards data collection. It was conducted from August to October 2013, in the campus of University College London in collaboration with Boots. The project team used a purpose-built Mobile Unit that can be driven to different locations. The unit contains a 3D body scanner, height gauge, a body composition monitor and various Internet-enabled healthcare devices, as shown in Figure 3.1. The platform was able to collect over 100 anthropometric measures, 70 body composition measures as well as weight, height, heart and lung function. The goal of the project was to accelerate the collection of data as much as possible, and also to drastically reduce the cost of gathering the data points mentioned above for a large population.

An essential aspect of the project was to establish the workflow that should be followed by each participant. Over 28 volunteers were trained and helped guide the participants through the data collection process. Over 650 UCL students and staff registered for participation, by completing an initial online questionnaire. Out of the 650 registrations, 175 individuals visited the Mobile Unit to complete the data collection process with in-person measurements. The data collected from the 175 individuals who visited

the mobile unit was analysed by UCL students and staff, while the data of the 475 individuals who did not visit the mobile unit despite registering online has been deleted. The analysis of the resulting data set and associated findings fall outside the scope of this thesis - our focus is on the requirements elicitation process and the design of the software system. The author of this thesis was responsible for designing the overall infrastructure supporting the survey and implementing the data collection system.



Figure 3.1: WellbeingUCL Mobile Unit and Data Collection Devices

3.5 A Process Model View for KAOS

BPMN is the ISO standard for process modelling[ISO13]. We have presented examples of BPMN process models, its underlying meta-model and definitions of the core BPMN modelling elements (e.g. activities, events, participants) in Chapter 2. Despite its wide adoption[Har16], empirical research has identified several recurring problems when BPMN is used during actual projects in the industry. Process models often have ambiguous activity labels, contain structural errors and larger models often become difficult to comprehend [IRRG09], [MRR10], [MRvdA10], [MNVDA07].

To ease adoption among business analysts already accustomed with BPMN as the process modelling language of choice, KAOS process models use a subset of the same graphical elements.

However, to address the shortcomings of BPMN, we introduce some additional constraints on the KAOS process models compared to the BPMN standard. The purpose of these changes is to make the models easier to understand and reduce the risk of modelling errors. Process models that have a clear, unambiguous meaning will be more valuable as a source of software requirements.

3.5.1 Meta-Model

The KAOS meta-model for describing business processes, shown in Figure 3.2, retains only a subset of the BPMN graphical elements. This selection is similar to what practitioners call the Level 1 palette [Sil09]: those BPMN elements carried over from traditional flow charts.

At the highest level, a **Business Process** is composed of **Flow Nodes** connected by **Sequence Flows**. **Activities** and **Gateways** are two types of **Flow Nodes** - in other words, they are specialisations of the abstract Flow Node element. In turn, **Events** are a specialisation of activities. An event is a special type of activity that occurs instantaneously, whereas activities have a duration. Each sequence flow connects precisely two flow nodes (designated as source and target). The source node of a sequence flow is executed before the target node.

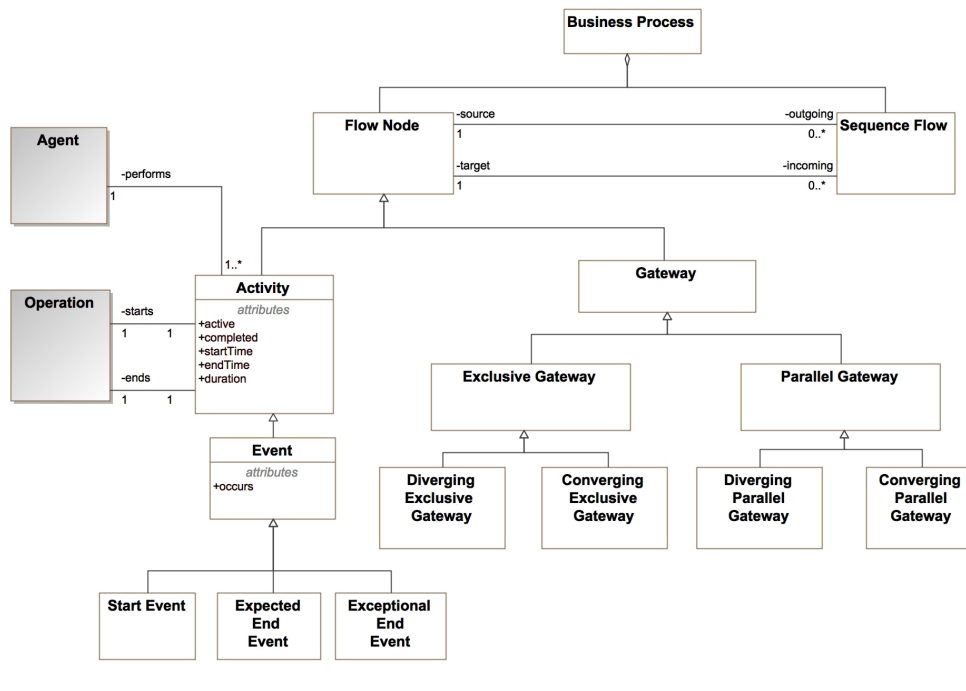


Figure 3.2: KAOS Process View

The **Gateway** element of the meta-model has two specialisations: **Exclusive Gateway** and **Parallel Gateway**. In the KAOS process models, exclusive gateways have guard conditions that reference objects and attributes from the system object model. A **Diverging Exclusive Gateway** has one incoming sequence flow and two outgoing sequence flows. In each process execution, precisely one activity will be executed after an exclusive gateway - which one depends on the system state when the guard condition is evaluated. A **Diverging Parallel Gateway** has one incoming sequence flow and at least two outgoing sequence flows. A parallel gateway signifies that at least two activities following the gateway should start execution, independently of each other. Converging gateways join the paths emerging from the diverging gateways.

Compared to the BPMN meta-model, one change we introduce is in the way we present **Events**. We consider them a specialisation of **Activities** because treating activities and events uniformly makes it easier to express the execution semantic of the business process. Both events and activities denote a change in the system - a transition from an initial state into a new state. We retain and make explicit the BPMN distinction between exceptional end events and expected end events. This specialisation of end events can be used to capture business rules - by clarifying which end events deliver value to business stakeholders and which end events do not.

Another departure from the BPMN meta-model is our reference to KAOS **Agents**, as the performers of activities. Agents replace a number of overlapping concepts in BPMN - **Lanes**, **Pools** and **Performers**.

Finally, we relate the process meta-model with the KAOS operation model by imposing that each process activity should have a start and an end operation.

3.5.2 Example Process Model: WellbeingUCL

Figure 3.3 shows an example process model, using for illustration the participant registration process for the WellbeingUCL project. Before any wellbeing data can be collected, prospective participants have to follow a standard procedure, to ensure they are eligible for participation and the study complies with the research ethics guidelines. Due to space constraints, we have split the process, following a graphical convention used in BPMN.

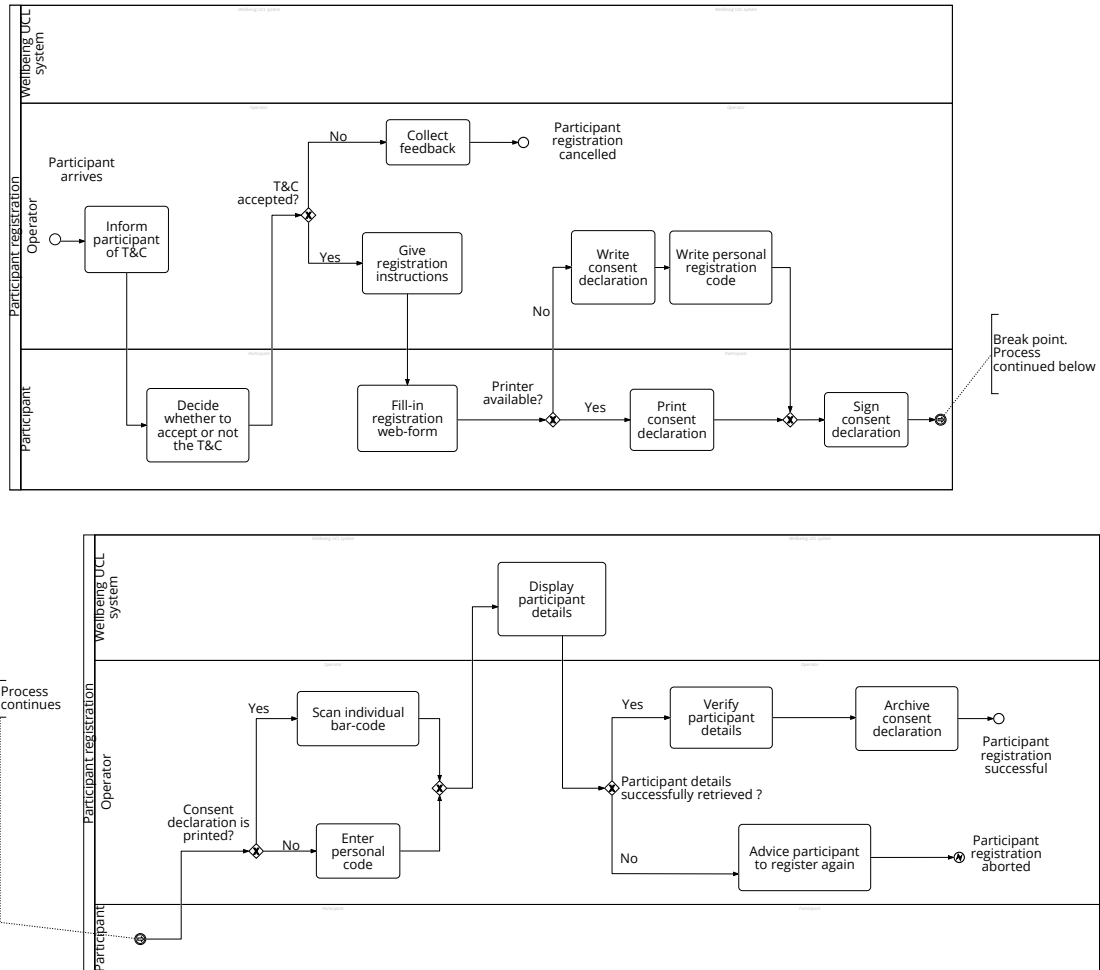


Figure 3.3: WellbeingUCL - Participant Registration Process

In the KAOS process view, agents responsible for performing activities are represented as horizontal lanes. Participants are the UCL staff and students willing to take part in the survey. The Operator is a volunteer that handles the data collection and offers information to the participants. Finally, the WellbeingUCL System maintains a database of all participants, their details and the unique IDs assigned to each.

Process activities are represented graphically as labelled rectangles, connected by sequence flows. Each activity represents a discrete step in the process.

The registration workflow starts with the operator giving a brief explanation about the WellbeingUCL survey to the interested participant. If she decides not to proceed with the registration, she will

answer a quick questionnaire. The feedback collected will be used to improve the design of future studies. If she decides to proceed with the registration, the operator will direct her to a website, where she can fill in the registration form. This form contains a questionnaire, meant to capture a subjective, self-reported measure of wellbeing. Once the participant fills in all the required information, she has the option to print a consent form. The printed consent form will contain a standard text that gives UCL researchers the right to use the collected information for research purposes. The form also contains a barcode representing a unique participant ID. If the participant can not print the consent form, she will have to write the consent declaration herself, adding her participant ID as well. In either case, the consent declaration has to be signed by the participant and then given to the operator. The operator will then have to enter the participant ID into the system and confirm that a record has been created for the respective participant. If the system does not display an entry for a given participant ID, the operator will assume the participant has not registered successfully and she will have to start the process again. If the WellbeingUCL system finds a record corresponding to the participant ID, it will display the data collected so far, through the web questionnaire. The operator will validate the record created for the participant is correct, by asking her to confirm some of the details. Finally, he will place the consent declaration for the new participant in a specifically assigned folder.

3.5.3 Structural Constraints

To reduce ambiguity in the process models, we introduce a number of constraints that govern how the process elements relate to each other. The role of these constraints is to make modelling assumptions explicit, ensuring the workflow is described and understood in a consistent manner, by all stakeholders.

For example, analysts would often link one activity with two or more consequent activities, as shown in Figure 3.4. It is not clear whether the activities B and C should start in parallel, or if only one activity may start during a process execution. In these cases, using either an exclusive or a parallel gateway removes ambiguity. In our example, the KAOS process model on the left uses an exclusive gateway to clarify that either Activity B or Activity C should be executed (but not both). The KAOS process model on the right uses a parallel gateway to clarify that both Activity B and Activity C should be executed.

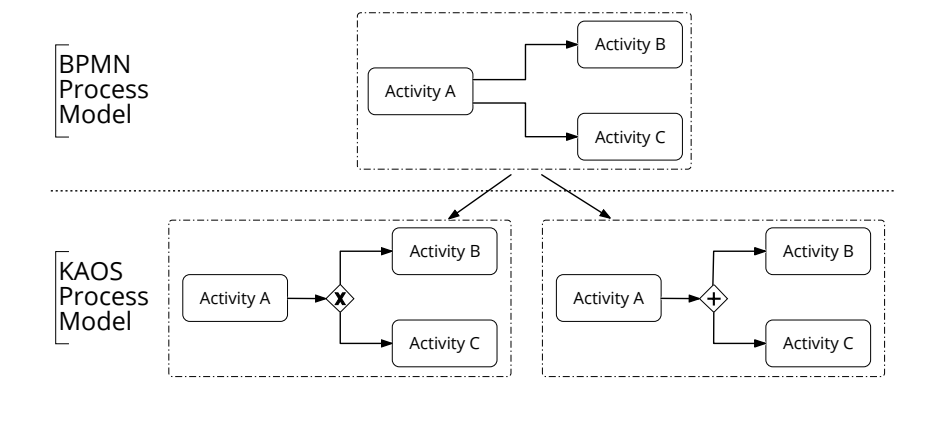


Figure 3.4: Structural Constraints in KAOS

The following rules should hold in any KAOS process model:

- Activities should have only one incoming and one outgoing sequence flow
- End events should have only one incoming sequence flow and no outgoing flows
- Start events should have only one outgoing sequence flow and no incoming flows
- Gateways should have only one incoming flow and several outgoing flows; or several incoming flows and only one outgoing flow.

Considering these constraints, we define a KAOS Process Model $\mathbf{P} = (N, Start, End, \delta)$ as:

- A set N of flow nodes partitioned into *Activities*, *Events* and *Gateways*
- A set of start nodes $Start \subset Events$
- A set of end nodes $End \subset Events$
- A sequence relation $\delta \subset N \times N$ defined as a set of tuples of nodes from the process \mathbf{P} satisfying a set of well-formedness constraints
 - $e_{start} \in Start$ has no predecessor in δ
 - $e_{end} \in End$ has no successor in δ
 - $act \in Activity$ has exactly one successor and one predecessor in δ
 - $gtw \in Gateway$ has either one predecessor and several successors or several predecessors and one successor

We are also introducing a *follows* relation between any two activities of the process model.

Def: Let act_1 and act_2 be two activities in a process model. We verify whether act_2 follows act_1 using the transitive closure of the sequence flow relation: $follows(act_2, act_1) \Rightarrow act_2 \in act_1 * \delta$.

3.5.4 Naming Conventions

We introduce naming conventions to encourage consistency and clarity in the way process elements are labelled. These changes are inspired by recommended best practices in the process modelling community [SM01].

- Activities should be labelled in the form VERB-NOUN. Ex: Open Drawer
- End events should be labelled in the form NOUN-VERBed. Ex: Drug Round Finished
- An exclusive gateway does not make a decision; it just tests a condition. It should be labelled as a question, and the two outgoing branches as *Yes* and *No*. Ex: Scan Successful?
- Separate end events should be used to indicate distinct end states

Following these naming conventions will make it easier to relate the process model to the other KAOS views. For example, in section 3.7 we discuss consistency rules relating the activity labels with the object model.

3.5.5 BPMN Messages as KAOS Events

In the BPMN standard, **Messages** are used to depict some exchange of information. Messages are attached to **Message Flows** connecting participants, activities or events.

In the KAOS process meta-model messages are not considered separate objects; instead, receiving or sending a message is modelled in KAOS as an event, as shown in Figure 3.5. This language design decision reduces the complexity of the meta-model, while maintaining consistency with the BPMN standard, which also defines catching events for messages.

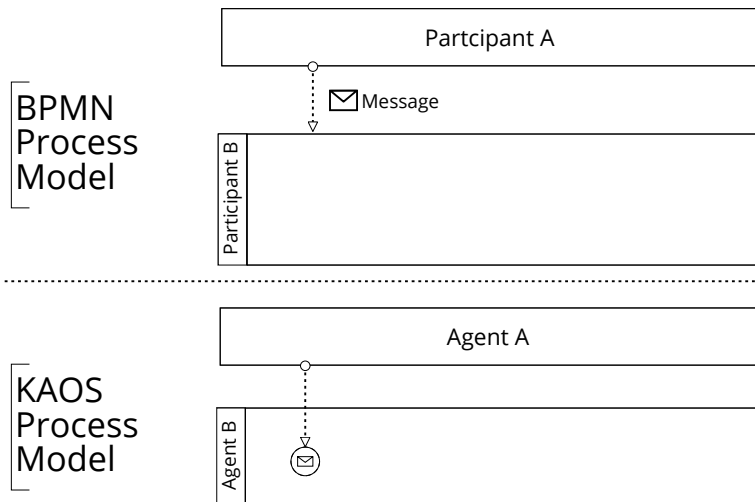


Figure 3.5: Representing Messages in KAOS

3.6 Process Annotations

We have made a conscious design choice to keep the graphical notation of the KAOS process models identical to BPMN. This is because BPMN is already an established standard to describe business process graphically[ISO13]. However, using the graphical notation on its own has some important limitations, if the process models are meant to be used for requirements elicitation.

First, the graphical notation for process models does not distinguish between a prescriptive sequence of activities and a descriptive sequence. This makes it difficult for business analysts to identify what can be changed in the workflow and what should be kept as is. This is a relevant question during projects where a new software system is also meant to support process re-engineering efforts.

Further, a graphical representation of a workflow does not explicitly define how each activity affects the system under consideration. Activity labels describe the work done during a process, but they do not show the exact changes occurring in the system as activities are completed.

Finally, the graphical notation lacks a way for business analysts to specify what are the business expectations across a number of executions of a given process. In other words, it lacks a quantitative view of the process. Consider the difference between two statements: "a process has to complete execution within 20 minutes" and "given all the executions of a given process, 96% of them should complete within

20 minutes”. The latter adds an additional layer of information.

We notably observed these limitations when analysing the WellbeingUCL project presented in Section 3.4. However, we note these are general limitations related to the expressive power of the graphical notation and are not specific to any one project.

To complement the graphical notation, we introduce a textual description that accompanies process activities, events and gateways. In the next sections, we will illustrate how annotations can be used to add more detail to both prescriptive and descriptive elements of the process model, as well as clearly distinguishing between the two types of modelling intent.

3.6.1 Activities Specification

The KAOS process view includes textual annotations allowing modellers to specify the activities’ pre and post conditions. Each activity has also a maximal allowed duration, as well as a maximal allowed delay after the previous activity has completed execution. These last two attributes make the specification of each activity more precise, by establishing upper limits for when an activity instance should start (relative to the previous element in the process flow) and when it should complete execution.

To illustrate the types of annotations we are introducing in this section, we use fragments from the Participant Registration process, developed for the WellbeingUCL survey. Figure 3.6 shows a small part of this process: the two different ways in which the participants could give consent for their data to be used. If there is a printer available in the data collection area, participants can print their registration form. This form also acts as a consent declaration so participants just need to sign. When a printer is not available, participants have to write the consent declaration by hand and then sign it. Either way, the form will also include the unique participant code - all the well-being data collected at a later stage will be linked to it. The participant registration code is automatically generated by the system, whenever a prospective participant fills in the registration web form.

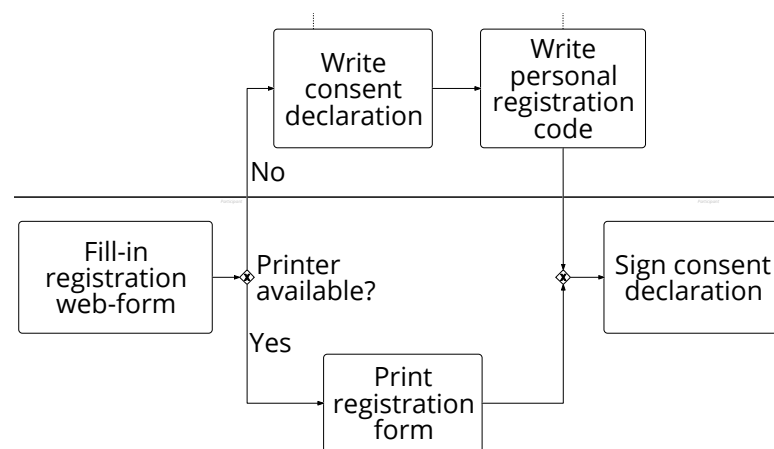


Figure 3.6: WellbeingUCL - Consent Declaration

In this fragment, the activity `Write Personal Registration Code` is defined as:

Activity	<i>WritePersonalRegistrationCode</i>
Domain pre-condition	\neg registrationCodeWritten \wedge consentDeclarationWritten
Required pre-condition for goal <i>Achieve[ParticipantIdentityKnown]</i>	registrationCodeGenerated
Required post-condition for goal <i>Achieve[ParticipantIdentityKnown]</i>	registrationCodeWritten
Maximal allowed duration	5 seconds
Maximal allowed delay	5 minutes

Formally, activities are declared by signatures over objects from the object model using linear temporal logic. The first two conditions capture the descriptive layer of the model. They simply declare the system states before and after an activity executes. The following two conditions capture the prescriptive layer of the model. They declare what additional conditions have to hold in the system, before and after the activity executes, such that a certain goal may be satisfied. Finally, the last two conditions also represent prescriptions (i.e. business expectations), but relate specifically to the start time and duration of the activity.

- A domain pre-condition characterises the states before any instance of the activity becomes active;
- A domain post-condition characterises the states after any instance of the activity completes execution;
- Required pre-conditions define those states in which an activity instance is allowed to start;
- Required post-conditions define additional conditions that must be satisfied once an instance of the activity completes;
- Maximal allowed duration defines the upper bound for the period of time for which any activity instance should be active;
- Maximal allowed delay defines the upper bound for the period of time which can pass between the system state when the previous activity has completed execution and the system state when the activity instance becomes active.

The activity annotations fulfil a number of roles.

First, they show what changes in the system, once an instance of the activity completes execution. In our example, `WritePersonalRegistrationCode` changes the value of a system state variable - `registrationCodeWritten`. In practice, the post-condition shows that once an instance of this activity completes, the registration code of the current participant will appear on his registration form.

Second, activities annotations allow business analysts to reason if two different activities have equivalent effects in the system. That is, even if the labels of two activities are different, we can show their execution semantic is the same. Consider the activities `WriteConsentDeclaration` and `Print-RegistrationForm`.

Activity	<i>WriteConsentDeclaration</i>
Domain pre-condition	\neg consentDeclarationWritten
Required pre-condition for goal <i>Achieve[LegalRequirementsSatisfied]</i>	T&C accepted
Required post-condition for goal <i>Achieve[LegalRequirementsSatisfied]</i>	consentDeclarationWritten
Maximal allowed duration	2 minutes

Activity	<i>PrintRegistrationForm</i>
Domain pre-condition	\neg registrationFormPrinted \wedge webFormFilledIn
Required pre-condition for goal <i>Achieve[LegalRequirementsSatisfied]</i>	T&C accepted
Domain post-condition	registrationFormPrinted
Required post-condition for goal <i>Achieve[LegalRequirementsSatisfied]</i>	consentDeclarationWritten
Maximal allowed duration	30 seconds

The required post-condition of `PrintRegistrationForm` clarifies that the registration form acts as a declaration of consent as well. If this weren't the case, business analysts could question whether the legal requirements for collecting data from participants can still be met, without a written consent declaration.

Third, annotations related to duration are used to compare alternative execution paths in a process. In our example, we can evaluate the time savings possible when the printer is available, compared with the process instances where the printer is not available. Then, any investment that results in better printer availability will have a traceable benefit for stakeholders - one of the high level business goals was to reduce the time required to process each participant in the WellbeingUCL survey.

Fourth, annotations clarify whether the changes in the system state brought upon by executing an activity are required for any business goal. In our example, `registrationCodeWritten` is a required post-condition, as this is necessary to establish the participant identity, which in turn is a business goal.

Fifth, activities annotations distinguish between situations where the ordering of activities has to be preserved and cases where it can be changed. This relies on the distinction between domain pre / post-conditions and required pre /post-conditions in KAOS. The former capture elementary state transitions, while the latter capture additional strengthening to ensure that the goals are met [LVL04]. This difference between domain pre-conditions and required pre-conditions is used to identify when a sequence of two activities is descriptive or prescriptive in nature:

- if the post-condition of the first activity is a required pre-condition of the second activity, the ordering should be preserved. Since the required pre-condition captures a strengthening necessary to ensure a goal is met, it means that the ordering of activities is necessary to achieve a certain goal.
- if the post-condition of the first activity is a domain pre-condition of the second activity, the ordering may be changed, as it is not required to ensure the satisfaction of any goal.

For the activity `WritePersonalRegistrationCode`, we made a conscious choice of declaring

consentDeclarationWritten part of the domain pre-condition, not a required pre-condition. This means the current ordering is not prescriptive, since no goal has been identified that would require this specific order among the two activities. A participant could equally start by writing his registration code, and only after that writing the consent declaration. Conversely, the annotation for SignConsentDeclaration defines a required ordering of activities. Specifically, the consent form can only be signed after it has been fully written (no blank signatures are allowed).

Activity	<i>SignConsentDeclaration</i>
Domain pre-condition	\neg consentDeclarationSigned
Required pre-condition for goal <i>Achieve[LegalRequirementsSatisfied]</i>	consentDeclarationWritten
Required post-condition for goal <i>Achieve[LegalRequirementsSatisfied]</i>	consentDeclarationSigned

Similarly to KAOS operations, activities can also have trigger conditions. Required trigger conditions define those states in which an activity instance should start immediately provided the domain pre-condition is true.

Analysts may explicitly define trigger conditions for an activity, using temporal logic operators and state variables. Analysts do not have to define an explicit trigger condition - they may instead specify a maximal allowed delay for an activity instance to start execution, once an instance of the previous activity in the process flow has completed. Finally, if the business analyst does not specify a trigger condition nor a maximal allowed delay, then there is an implicit obligation for the agent responsible to perform the activity to *eventually* start execution. This is in line with BPMN token based semantics. In the BPMN standard, a token moving across a sequence flow does not have any timing constraints. A token might take a long or short time to move across the sequence flow and so the time difference between the completion of one activity and the start of the next activity is not bounded.

3.6.2 Gateways Specification

Process modelling guidelines [SM01] stress that exclusive gateways are meant to evaluate a condition. The corresponding outgoing sequence flows then link to the next activities in the process, one for each possible outcome.

However, simply representing the two different ways in which the process execution can progress is not enough to capture the business intent. Consider for example the gateway that tests whether the printer is available, in the WellbeingUCL registration process:

Exclusive Gateway	<i>Printer available ?</i>
Condition	functionalPrinter \wedge \neg paperTrayEmpty \wedge \neg inkEmpty

The intent of the stakeholders is that the printer should be available in as many cases (i.e. process executions) as possible. To make explicit this prescriptive layer of a process model, we extend the annotations for exclusive gateways to include an expected ratio between the number of instances when one branch becomes active and the number of instances when the other branch becomes active.

To acknowledge the fact that some process models will be descriptive in nature, capturing the current state of a system rather than the business expectation, an exclusive gateway can also be annotated with the current ratio, in addition to the desired ratio. We give below a complete annotation for the gateway `Printer Available?`:

Exclusive Gateway	<i>Printer available ?</i>
Condition	<code>functionalPrinter \wedge \neg paperTrayEmpty \wedge \neg inkEmpty</code>
Yes branch active currently	80% of cases
Yes branch active target	95% of cases

3.6.3 End Events Specification

BPMN modelling style guidelines include advice about how to represent process end states. Specifically, it is considered best practice to indicate success and exception end states of a process with separate end events [Sil09]. In line with these guidelines, we will annotate end events as either expected end events (signifying a success state) or exceptional end events (signifying an exception state).

Def: an **Expected End Event** shows a successful execution of the process, where the end result is acceptable to all stakeholders. The result of a process must be individually identifiable and countable [SM01]. For example, in the WellbeingUCL Participant Registration process, `Participant Registration Successful` is an expected end event. One can count how many successful registrations occur each day. Moreover, this is what the stakeholders expect, the value the process brings from a business perspective: registering new participants to the survey.

Def: **Exceptional end events** represent states in which the process has completed execution in a manner that is unsatisfactory for the stakeholders. In our example, a participant registration that has to be aborted delivers no value, and so this end state should be avoided as much as possible.

End events may optionally be annotated with their current and expected occurrence rate. The values of these two can be used to reason about the performance of the current process (i.e. whether the expected behaviour is met in practice).

Figure 3.7 shows the process model for the final part of the WellbeingUCL registration process. At this stage, a WellbeingUCL volunteer responsible for data collection tries to retrieve the details linked to a participant code in order to ensure these are accurate and have been correctly saved in the database. This process model assumes the participant is present at the survey location so that he can confirm the accuracy of the retrieved data. The expected end event of the process is a successful registration.

End Event	<code>Participant Registration Successful</code>
Type	Expected end event
Occurrence Rate Currently	70%
Occurrence Rate Target	95%

In case no details are displayed for a given code, the registration should not be deemed successful - the code may be invalid, or the data may not have been saved in the database. Since in this case the participant will have to go through the registration process for a second time, the end event `Participant`

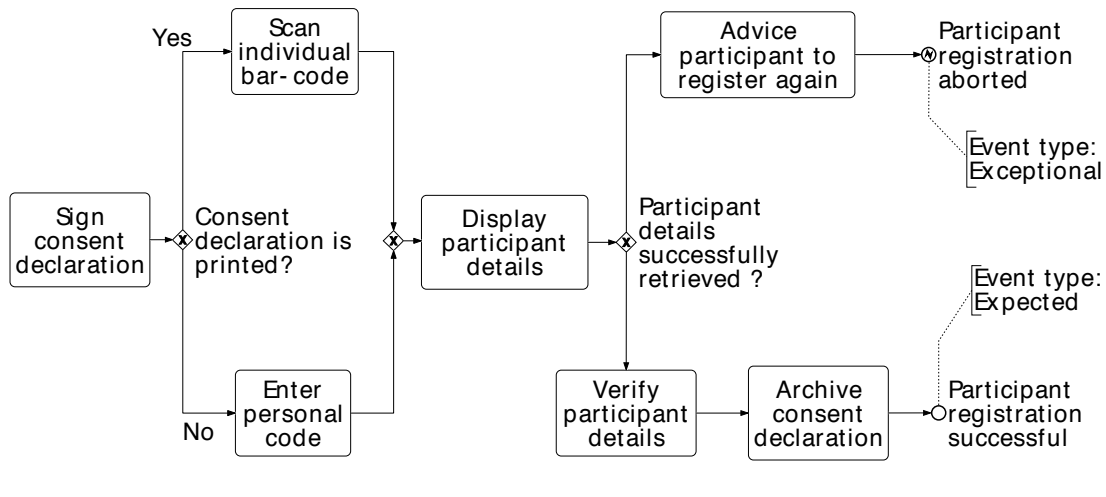


Figure 3.7: WellbeingUCL - Process End Events

Registration Aborted will be labelled as an exceptional event, to be avoided as much as possible.

3.6.4 Consistency Rules between Graphical and Textual Notation

To ensure consistency between the graphical representation of the process and the textual annotations of the process elements, we define a set of consistency rules. These rules describe how the constraints imposed by graphical connectors such as sequence flows and gateways must be reflected in the activities annotations. In each of the cases we present below, analysts will have to choose between defining a required or a domain pre-condition. Marking the post-condition of the previous activity as a required pre-condition for the following activity signifies that the order should be preserved, as it serves a goal. If the post-condition of the previous activity is just a domain pre-condition for the following activity, the ordering of the activities is not necessary for the satisfaction of any goal.

First, we consider the case of two activities connected by a sequence flow, as shown in Figure 3.8

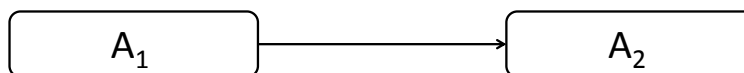


Figure 3.8: Consistency Check: Sequence of Activities

- If the domain post-condition of A_1 is a required pre-condition for A_2 then A_2 **must** follow A_1
- If the domain post-condition of A_1 is not a required pre-condition for A_2 then A_2 **may** follow A_1
- If the two activities are connected by a sequence flow, the domain post-condition A_1 is either a required or a domain pre-condition for A_2 .

Second, we consider three activities connected through a parallel divergent gateway (Figure 3.9):

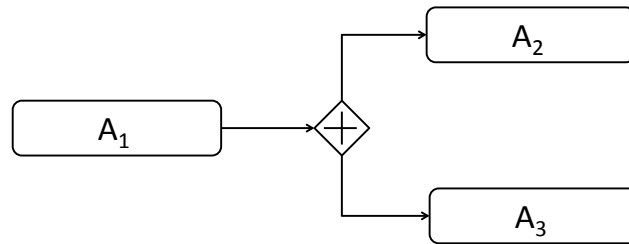


Figure 3.9: Consistency Check: Parallel Gateway

In this case, the parallel gateway is transparent and we can apply the sequence flow rule to the pairs of activities A_1 and A_2 and A_1 and A_3 respectively.

Third, we consider three activities connected through a divergent exclusive gateway (Figure 3.10):

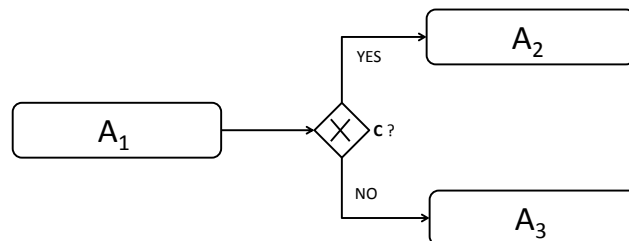


Figure 3.10: Consistency Check: Exclusive Gateway

In this case we need to consider the condition C of the gateway and A_2 and A_3 . The consistency rules are:

- If the gateway condition C is a required pre-condition for A_2 then A_2 **must** be reachable from the gateway through the 'yes' branch and not reachable through the 'no' branch
- If $\neg C$ is a required pre-condition for A_3 then A_3 **must** be reachable from the gateway through the 'no' branch and not reachable through the 'yes' branch
- If A_1 and A_2 are connected by the 'yes' branch of an exclusive gateway, the conjunction of the domain post-condition of A_1 and the condition C of the gateway represents either a required or a domain pre-condition for A_2 :

Activity	A_1	Activity	A_2
DomPre	$\neg C_1$	DomPre / ReqPre	$C_1 \wedge C$
DomPost	C_1		
Gateway	G		
Condition	C		

Fourth, we consider three activities connected by a convergent exclusive gateway (Figure 3.11):

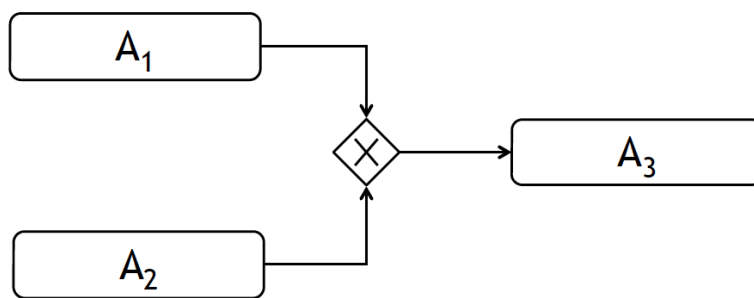


Figure 3.11: Consistency Check: Exclusive Convergent Gateway

If A_1, A_2, A_3 are connected by a convergent exclusive gateway G then the disjunction of the domain post-conditions of A_1 and A_2 represents either a required or a domain pre-condition for A_3 :

Activity	A_1	Activity	A_3
DomPre	$\neg C_1$	DomPre / ReqPre	$C_1 \vee C_2$
DomPost	C_1		
Activity	A_2		
DomPre	$\neg C_2$		
DomPost	C_2		

Fifth, we consider three activities connected by a convergent parallel gateway (Figure 3.12):

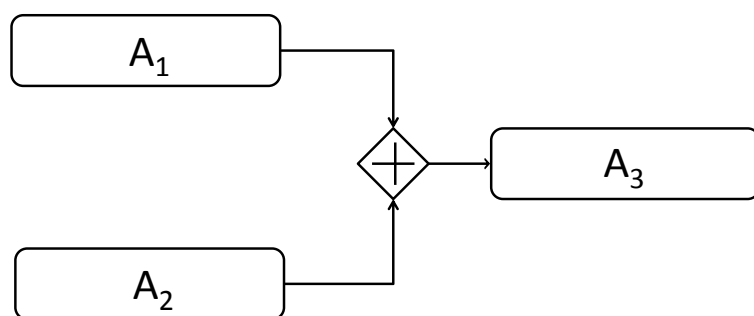


Figure 3.12: Consistency Check: Parallel Convergent Gateway

If A_1, A_2, A_3 are connected by a convergent parallel gateway G then the conjunction of the domain post-conditions of A_1 and A_2 represents either the domain or the required pre-condition for A_3 :

Activity	A_1	Activity	A_3
DomPre	$\neg C_1$	DomPre / ReqPre	$C_1 \wedge C_2$
DomPost	C_1		
Activity	A_2		
DomPre	$\neg C_2$		
DomPost	C_2		

3.7 Object Model and Process Model: Activities as KAOS Objects

When discussing the relation between the object model and the process model, there are three layers that should be considered. First, the role of process activities as entities in the domain under consideration. Second, the relation between activities labels and the object model. Third, the relation between the activities annotations and the object model. In this section, we present consistency rules applicable at each layer. Process activities are considered a specialisation of objects in the KAOS model, as shown in Figure 3.13.

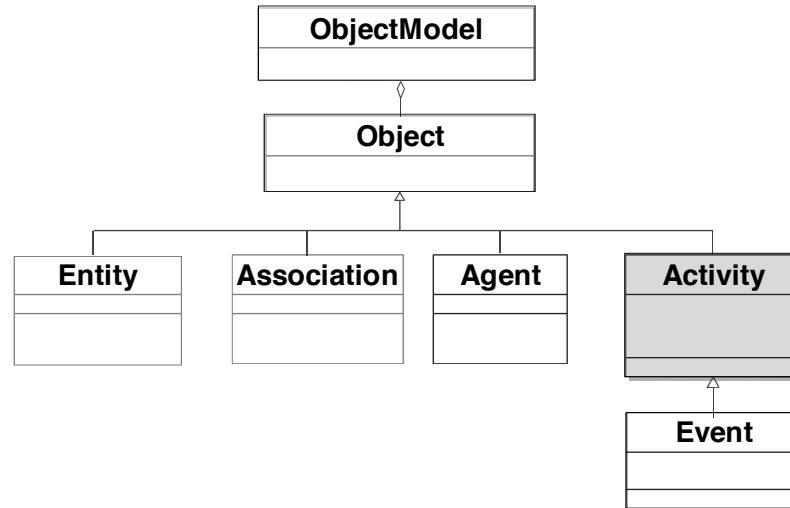


Figure 3.13: Extended KAOS Object Model

An **Activity** depicts work to be done by some agent. Activities have a series of built in attributes used to describe the activity execution cycle. A boolean attribute *active* is used to depict whether an activity instance is being performed in a given system state. The *startTime* and *endTime* are attributes used to show when an activity starts and completes execution respectively. Consequently, each instance of an activity has a *duration*: the difference between its start and end time. The effective duration of an activity instance can only be measured in the context of a process execution.

Activity	Act_i
Has	<i>active</i> : Boolean (denotes that the activity instance is currently being performed) <i>completed</i> : Boolean (denotes that the activity instance has been completed) <i>startTime</i> : Time (denotes the time an activity instance has become active) <i>endTime</i> : Time (denotes the time an activity instance has completed) <i>duration</i> : Time (denotes the time period an activity has been active)
DomInvar	An activity instance is completed if it was active and now is no longer active $\forall act : Act_i$ $act.completed \equiv \blacklozenge act.active \wedge \neg act.active$ The duration of an activity instance is by definition: $\forall act : Act_i$ $act.duration \equiv act.endTime - act.startTime$ Once completed, an activity instance cannot become active again. $\forall act : Act_i$ $act.completed \Rightarrow \square \neg act.active$

We consider events a specialisation of activity, where duration is zero. Events have a built in boolean attribute *occurs*, which shows if the event takes place in the current system state or not. For any event instance, the *occurs* attribute is true in only one system state, identified by the *occursTime* attribute.

Event	Ev_i
Has	occurs: Boolean occursTime: Time
Controlled by	ag: Ag_i
DomInvar	$\forall ev : Ev_i$ $ev.occurs \equiv ev.active$ $ev.occursTime \equiv ev.startTime$

The second type of consistency rules between the process model and the object model covers the activity labels. In section 3.5.4, we are recommending activity labels follow the pattern VERB-NOUN. To keep the process and the object model consistent, nouns used in activity labels should have a corresponding object in the object model.

The third category of consistency rules between the process model and the object model relates to the activities annotations. Since annotations on each process activity are expressed as assertions concerning state variables, these variables should also be attributes of objects in the object model, under the control of agents in the system.

3.8 Operational Model and Process Model: Operational Semantics of a Process Model

We define the execution semantic of activities by mapping each activity to a start and an end KAOS operation. The execution of an activity instance is equivalent to the application of the two corresponding operations in the system.

For every activity Act_i in the process model the operations $Start_Activity_i$ and $End_Activity_i$ are defined as follows:

Operation	$Start_Activity_i$	Operation	$End_Activity_i$
Input	act: Act_i	Input	act: Act_i
Output	act: Act_i	Output	act: Act_i
DomPre	$\neg act.active$	DomPre	act: act.active
DomPost	act.active	DomPost	act: $\neg act.active$

The operation signatures given above are common across all activities. The operations are then further specified considering each activity's pre and post-conditions, as follows:

- Any domain pre-condition of the activity becomes a domain pre-condition of the start operation
- Any required pre-condition of the activity becomes a required pre-condition of the start operation
- The trigger condition of the activity becomes the trigger of the start operation. If the business analyst has not specified an explicit trigger condition, the trigger for the start operation will follow the template: $\neg act.active \text{ S}_{=maximal\ allowed\ delay-1} (\text{DomPre} \wedge \neg act.active)$

- Any domain post-condition of the activity becomes a domain post-condition of the end operation
- Any required post-condition of the activity becomes a required post-condition of the end operation
- The trigger of the end operation is defined in relation to the activity's maximal duration following the template: $\neg \text{DomPost } S_{=\text{maximal allowed duration}-1} (\text{act.active} \wedge \neg \text{DomPost})$

To illustrate the points above, we revisit the annotation for the *WritePersonalRegistrationCode* activity and we introduce the signatures of its start and end operations.

Activity	<i>WritePersonalRegistrationCode</i>
Domain pre-condition	$\neg \text{registrationCodeWritten} \wedge \text{consentDeclarationWritten}$
Required pre-condition for goal <i>Achieve[ParticipantIdentityKnown]</i>	$\text{registrationCodeGenerated}$
Required post-condition for goal <i>Achieve[ParticipantIdentityKnown]</i>	$\text{registrationCodeWritten}$
Maximal allowed duration	5 seconds
Maximal allowed delay	5 minutes

The start and end operation are fully specified as follows:

Operation	<i>Start_WritePersonalRegistrationCode</i>
Input	$\text{act: } \text{WritePersonalRegistrationCode}$
Output	$\text{act: } \text{WritePersonalRegistrationCode}$
Domain pre-condition	$\neg \text{act.active} \wedge \text{consentDeclarationWritten} \wedge \neg \text{registrationCodeWritten}$
Required pre-condition for goal <i>Achieve[ParticipantIdentityKnown]</i>	$\text{registrationCodeGenerated}$
Required trigger for goal <i>Achieve[ParticipantIdentityKnown]</i>	$\neg \text{act.active } S_{=4\text{min}59\text{sec}} (\text{consentDeclarationWritten} \wedge \neg \text{registrationCodeWritten}) \wedge \neg \text{act.active}$
Domain post-condition	act.active

Operation	<i>End_WritePersonalRegistrationCode</i>
Input	$\text{act: } \text{WritePersonalRegistrationCode}$
Output	$\text{act: } \text{WritePersonalRegistrationCode}$
Domain pre-condition	act.active
Required trigger for goal <i>Achieve[ParticipantIdentityKnown]</i>	$\neg \text{registrationCodeWritten } S_{=4\text{sec}} (\text{act.active} \wedge \neg \text{registrationCodeWritten})$
Domain post-condition	act.completed
Required post-condition for goal <i>Achieve[ParticipantIdentityKnown]</i>	$\text{registrationCodeWritten}$

3.9 Goal Model and Process Model: Goal Satisfaction Arguments

In the KAOS framework, the purpose of developing goal models is to support requirements engineering activities. These activities include requirements elaboration, consistency and completeness checking, identifying alternative design options for the software system and selecting among these alternatives [VL01]. When defining the integration between process models and goal models in KAOS, our aim is to allow business analysts to include and use process models during these requirements engineering activities.

At the core of our proposal is the idea that processes are executed in order to satisfy business goals. It follows that process models may be used to elicit goals, domain expectations and software requirements, by enquiring what is the rationale for executing each activity. Business analysts can reason (informally or formally) about the alignment between process models and goal models. To check the two models present a consistent view over the system of interest, they should verify that each activity in the process model contributes to at least one goal and that each goal in the goal model is linked to at least one activity. Inconsistencies should be addressed by adding or removing goals or activities, in their respective models. This critical analysis facilitates requirements elaboration.

The goal and the process models also play a role when it comes to selecting among alternative design options for the system-to-be. In Section 3.10 we introduce the concept of Intentional Fragment, that explicitly relates one or more activities in the process to a common goal they contribute to. Intentional fragments allow business analysts to consider alternative system designs - each option is expressed as a set of different activities (compared to the original process) that still satisfy the functional goal of the Intentional Fragment. However, each such set of activities may have a different impact on the non-functional goals of the system. Reasoning about partial goal satisfaction is an established KAOS technique [LVL04]. Using information from the process model (such as time and cost of performing certain activities), business analysts are able to quantify the degree to which every design alternative satisfies the non-functional goals and thus select one among the available options.

Process models can also be used by business analysts for obstacle analysis [VL01]. Obstacles are conditions in the system that prevent goal satisfaction. This KAOS technique improves the requirement specification, insofar as the final specification includes mitigation strategies for the obstacles identified. Agents are responsible for starting and completing each activity in the process, by performing the respective start and end operations. Business analysts can use critical thinking to try and identify situations that would prevent agents to start or complete activities - these situations represent obstacles, as the goal linked to the activity is no longer satisfied.

To incorporate process models into the KAOS framework, our aim is to establish a sound definition of the relation between process models and goal models. To address this relation we consider three different aspects.

First, analysts can demonstrate that a business process *satisfies* a goal through the operational semantics of the process, as shown in Figure 3.14. A process model \mathbf{P} satisfies a goal \mathbf{G} if the operational semantic of that process satisfies the goal: Satisfies (\mathbf{P}, \mathbf{G}) iff $\text{OpSem}(\mathbf{P}) \models \mathbf{G}$.

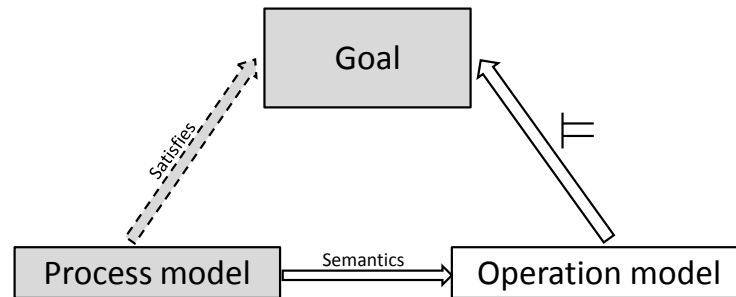


Figure 3.14: Process and Goal Models

Second, business analysts can also relate individual execution paths through the process to a goal. Since executions of the same process model may differ (as exclusive gateways activate different sequence flows), it may be that a goal is satisfied by some process instances, but not by others. That is to say, even if one can not prove that a process model in its entirety satisfies the goal, he may still prove that a certain execution path through the process ensures goal satisfaction. Similarly, there could be execution paths that explicitly contradict a goal.

Third, for each activity that has to be executed, a set of standard system requirements or domain expectations (depending on whether the activity is carried out by a software system or by a human agent) can be defined and incorporated into the requirements specification.

In the next subsections, we discuss each of these three cases. To illustrate our discussion, we use another fragment from the WellbeingUCL registration process, shown in Figure 3.15. This part of the process deals with collecting feedback from a participant in case she refuses to participate in the survey because she finds the terms and conditions unacceptable. In this case, the participant will fill in a short questionnaire, the results of which will be used to improve future wellbeing surveys.

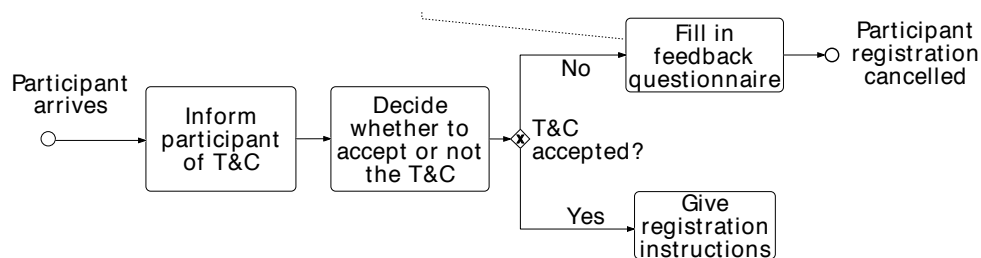


Figure 3.15: WellbeingUCL - Collect Feedback

3.9.1 Process Level Goal Satisfaction Arguments

In order to prove goal satisfaction, business analysts can try to instantiate known operationalisation patterns [LKMU08]. The semantic of goal operationalisation has been presented in Section 2.2.5. A business analyst will verify that the operations derived from the annotations of one or more activities match an operationalisation pattern for the goal under consideration.

Consider for example the goal *Achieve [ParticipantFeedbackCollected]* and the activity *Fill In Feedback Questionnaire* defined as follows:

Goal	<i>Achieve [ParticipantFeedbackCollected]</i>
Def	When a prospective participant does not accept the terms and conditions of the Well-beingUCL survey, then feedback should be collected from that participant
Formal Def	$\forall \text{participant: Participant}$ $\text{decisionMade} \wedge \neg \text{T\&CAccepted} \Rightarrow \diamond_{\leq 30min} \text{questionnaireCompleted}$

Activity	<i>FillInFeedbackQuestionnaire</i>
Domain pre-condition	$\text{decisionMade} \wedge \neg \text{T\&C accepted} \wedge \neg \text{questionnaireCompleted}$
Domain post-condition	$\text{questionnaireCompleted}$
Maximal allowed duration	5 minutes
Maximal allowed delay	10 minutes

The activity *Fill in feedback questionnaire* satisfies the goal *Achieve [ParticipantFeedbackCollected]* if the analyst proves that the start and end operations of the activity satisfy the goal, as shown in Figure 3.16.

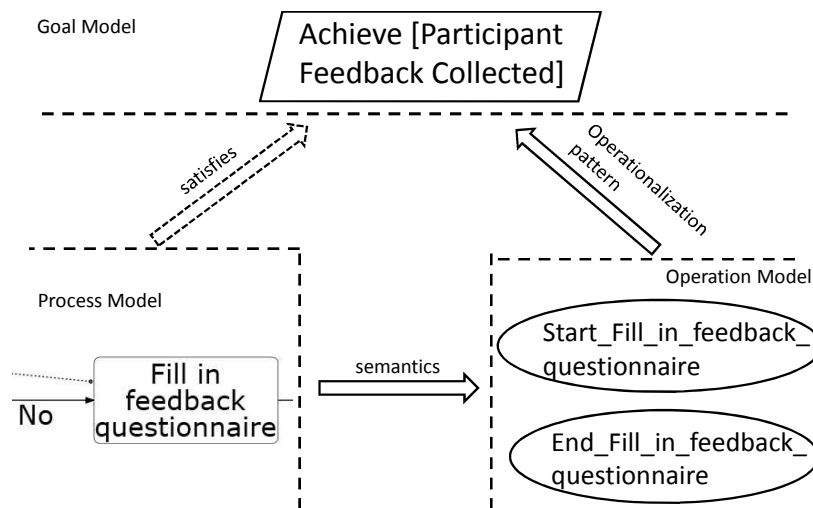


Figure 3.16: Goal Satisfaction Arguments

To prove this, he may demonstrate that the end operation of the activity matches an operationalisation pattern for the goal under consideration. In our example, the goal is a bounded achieve goal. The general form for these type of goals is $C \Rightarrow \diamond_{\leq d} T$, where T represents a target state (in our case questionnaireCompleted). The goals are operationalised as:

Operation	Op
DomPre	$\neg T$
DomPost	T
ReqTrig	$\neg T S_{=d-1} (C \wedge \neg T)$

It follows that in order to demonstrate that the activity satisfies the goal, one needs to demonstrate that the end operation of the activity matches the signature:

Operation	Op
DomPre	\neg questionnaireCompleted
DomPost	questionnaireCompleted
ReqTrig	\neg questionnaireCompleted $S_{=29:59}$ (decisionMade \wedge \neg questionnaireCompleted)

Indeed, the end operation for the activity Fill in feedback questionnaire, derived from the activity signature, is defined as follows:

Operation	End_Fill_in_feedback_questionnaire
DomPre	\neg questionnaireCompleted
DomPost	questionnaireCompleted
ReqTrig	\neg questionnaireCompleted $S_{=29:59min}$ (decisionMade \wedge \neg questionnaireCompleted)

While this is a simple example, business analysts can also apply model checking techniques on the operational model derived from the entire process [LKMU08]. However, with larger process models, this can become too computationally expensive. For this reason we introduce in Section 3.10 Intentional Fragments, so that business analysts can isolate parts of the process model that are relevant for a specific goal.

To complement these formal techniques, informal reasoning about goal satisfaction is also possible. When business analysts write the textual annotations for the process model, they can identify instances where known goals are not satisfied by the current process relying solely on the activities annotations. For example, in the wellbeingUCL survey, the goal Achieve [LegalRequirementsSatisfied] states that no data shall be collected for a participant without their consent. By annotating the process activities we discover that a process execution path where a participant first fills in the web registration form and then prints and signs the consent form contradicts this goal. Personal data is collected via the web form in the absence of a signed consent declaration.

3.9.2 Partial Goal Satisfaction Arguments

To illustrate this case, we update the previous goal definition, to state that feedback should be collected from all the prospective participants, not only from those declining to participate.

Goal	<i>Achieve [ParticipantFeedbackCollected]</i>
Def	When a prospective participant arrives, then feedback should be collected from that participant
Formal Def	\forall participant: Participant participantArrives \Rightarrow $\diamond_{\leq 45min}$ questionnaireCompleted

This goal definition now captures the real business intent, but this leads to an inconsistency with the process model.

Considering the annotations on the gateway T&C accepted?, we can derive a measure for the partial goal satisfaction - in this case only 5%. This corresponds to the 5% of potential participants that

decline to take part in the study, once they receive more information about it - with the current workflow, they would be the only ones for which feedback is collected.

Exclusive Gateway	<i>T&C accepted ?</i>
Condition	<i>termsAccepted</i>
Yes branch active currently	<i>95% of cases</i>

3.9.3 System Requirements and Domain Expectations

If we consider the process model to be prescriptive, we can also introduce consistency rules relating the goal, agent and process model.

For every activity Act_i performed by the agent Ag and preceded by activity Act_{i-1} in the sequence relation, there are three goals under the responsibility of Ag .

First, the agent should start executing the activity, once the preceding activity completes. Second, the agent should complete any activity that it starts. Third, the agent should not start the activity before the preceding activity has completed.

Goal	<i>Achieve[A_iStartedWhenA_{i-1}Completed]</i>
Def	When A_{i-1} is completed, then A_i will eventually start execution.
Formal Def	$\forall a_{i-1} : A_{i-1}$ $a_{i-1}.completed \Rightarrow \diamond(\exists a_i : A_i)a_i.active$
Responsibility	Ag

Goal	<i>Achieve [Activity$_i$CompletedWhenStarted]</i>
Def	When an instance of $Activity_i$ is active, it will eventually be completed
Formal Def	$\forall act : Act_i$ $act.active \Rightarrow \diamond act.completed$
Responsibility	Ag

Goal	<i>Avoid[A_iStartedBeforeA_{i-1}Completed]</i>
Def	A_i should not start before A_{i-1} is completed.
Formal Def	$\forall a_{i-1} : A_{i-1}, \forall a_i : A_i$ $a_i.active \Rightarrow a_{i-1}.completed$
Responsibility	Ag

These low level requirements are valuable because they facilitate obstacle analysis - for each activity, one may ask if there are any obstacles that could prevent the agent responsible for performing the activity to either start, or complete the task. For example, if an agent can not monitor when the preceding activity completes execution, it will not be able to start the execution of the next activity in the process. Additional goals of type *Avoid* can be derived when an activity is preceded by a gateway. For example, in cases where two activities are on different branches of an exclusive gateway, the system should avoid executing both of them in the same process instance.

3.10 Intentional Fragments

We introduce Intentional Fragments as a more flexible instrument that allows business analysts to explicitly relate activities in the process model to the goals they contribute to.

In this section, we discuss the rationale for introducing intentional fragments in the extended KAOS framework, we formally define the concept and then illustrate it by presenting a few examples of intentional fragments in the WellbeingUCL registration process.

3.10.1 Intuitive Definition

An intentional fragment is a collection of process elements (gateways, activities and events) that contribute to a common goal.

The idea of adding an additional layer of information to business process models, to capture the rationale for performing a given activity, may be found in the requirements engineering literature and in the business process management literature. In the former, Strategic Dependency Models have been proposed as a mean to capture the '*intentional* dimension of organisational work', by establishing relations between agents, goals and activities [YM96]. In the latter, *intention-oriented process modelling* aims to explicitly represent within the process models the goals that need to be accomplished [RP07]. Considering this previous work, the concept of Intentional Fragment thus represents another way to make the intention or the reason for performing a process activity explicit.

Business analysts can establish informal **contribution** relations between process activities and the goals they help satisfy. Within the context of a process, an activity said to be contributing to a goal may not be sufficient to satisfy the goal, but it is nevertheless required, based on the business analysts implicit domain knowledge.

These contribution relations may be validated or invalidated, when the process and the goal models become more detailed and more of the implicit domain knowledge becomes explicit. Formally, if the process **P** has been proven to satisfy a goal **G**, we say that the activity **A** contributes to **G** if by removing that activity from the process, the goal **G** will not be satisfied any more.

An intentional fragment for a given goal is composed of all the activities contributing to that goal, as well as the gateways and sequence flows linking those activities in the complete process model.

We also distinguish between *Potential* and *Justified* intentional fragments.

A **Potential Intentional Fragment** is composed of activities that **contribute** to the same goal. However, the collection of activities may not be sufficient to ensure the goal satisfaction.

A **Justified Intentional Fragment** is composed of activities that when executed, are proven to **satisfy** a given goal.

This distinction is necessary because, unless the process and the goal models are completely aligned, there may be cases where a process model does not satisfy a business goal, although it contains activities related to that goal.

To test if business analysts are able to understand and operate with the concept of intentional fragments, an evaluation was conducted in a French research laboratory [CCML⁺12]. Using semi-structured interviews, 21 people employed by the institution were asked to complete three exercises: first, identify

the goal, given a group of process activities; second, identify the process activities, given a goal; and third, identify intentional fragments, given the complete process model. This research shows that industry practitioners are able to use the concept of Intentional Fragments on real process models and identify goals, as well as areas where there is a misalignment between the process model and known business goals. The thesis will present further evaluations of use of intentional fragments in real-world projects in Chapters 5 and 6.

3.10.2 Motivation

In the previous sections we have already defined a relation between the process model and the goal model: a process model **satisfies** a goal. For this relation to hold, a business analyst needs to prove either of the following conditions:

- the process model contains one or more activities such that the operations derived for them match a goal operationalisation pattern;
- the complete operation model derived from the overall process has been proven to satisfy the goal through model checking.

However, real world process models can be incomplete, may lack annotations, can be too complex or difficult to follow. In these instances, formal goal satisfaction arguments are difficult to construct.

Intentional fragments allow a business analyst to relate parts of a process model with specific goals at an earlier stage in the requirements elicitation process, before proceeding to formal model checking.

The notion of Intentional fragments has correspondents in process modelling frameworks already used in the industry. For example, Signavio (a process modelling tool) has introduced the possibility of having several different views of the same process: in each view, one or more process elements can be hidden, while the overall structure of the process is preserved. The Signavio guidelines [Sig] recommend using process views in order to:

- help readers focus on functional parts of the process;
- distinguish between the happy path and exceptional process paths;
- show different variations of a process.

Each intentional fragment is akin to a process view - the criteria based on which activities are retained in the view is whether they contribute or not to the satisfaction of the goal under consideration.

3.10.3 Intentional Fragments in the WellbeingUCL Process

For illustration, we offer an overview of the Intentional Fragments identified for the WellbeingUCL registration process as shown in Figure 3.17. Since identifying the intentional fragments relies on establishing contribution relations between goals and activities and these relations are defined by the business analyst, it is important to stress that the view offered here is only one possibility, based on the analysts' domain knowledge.

There are two interesting cases that we briefly mention here and expand in the next chapter.

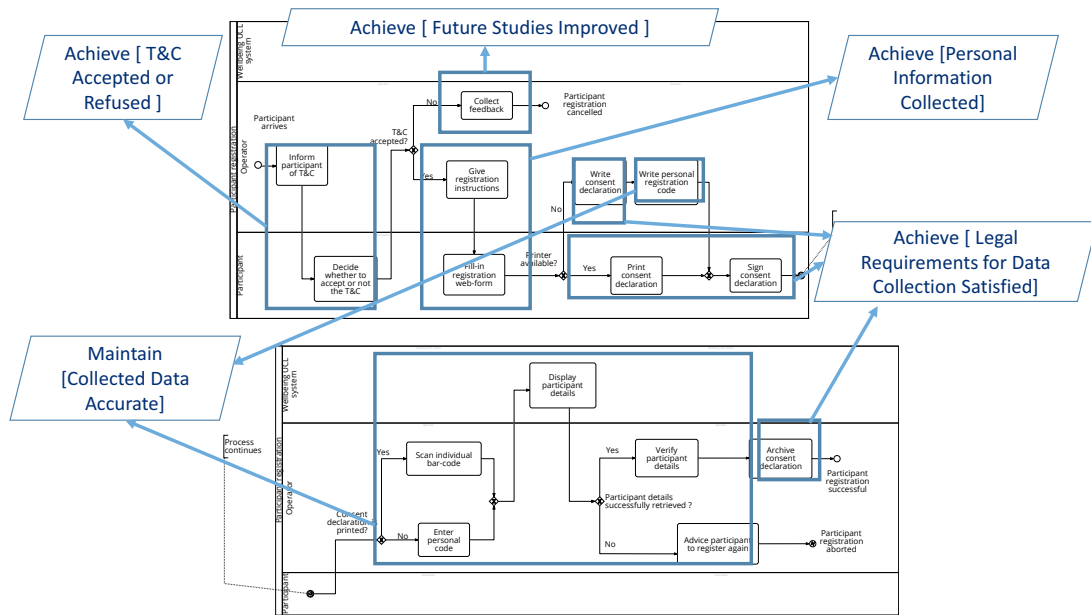


Figure 3.17: WellbeingUCL - Intentional Fragments

First, the intentional fragment for the goal `Achieve [FutureStudiesImproved]` contains only one activity. This may be a valid intentional fragment. Equally, business analysts may decide the goal is not satisfied by this activity alone, thus identifying a misalignment between the process and the goal model.

Second, the intentional fragment for the goal `Achieve [LegalRequirementsForDataCollection-Satisfied]` is composed of disparate activities. This is also a valid intentional fragment, and in many real world processes this is often the case: activities being performed at different stages of the process share a common business goal.

3.10.4 Applications of Intentional Fragments

Through the work on the WellbeingUCL project, we have identified a number of different ways in which Intentional Fragments can be used. We give here an overview, and in Chapter 5 and Chapter 6 we evaluate Intentional Fragments in two industry projects. On one hand, we consider the case where the process model and the goal models have been developed independently, possibly at separate times. Business analysts can use intentional fragments to reason about the alignment of the process and goal models early in the analysis stage of the project, to guide further development of both models towards achieving a consistent view of the system. We discuss the specific outcomes of such an analysis in Section 3.11. If there are different variants of a process (for example an as-is process model, and one or more to-be models), business analysts can identify the fragments in each variant that contribute to a specific goal. Thus, they can compare different process variants based on how well they satisfy each of the business goals.

On the other hand, we consider the case where no prior documentation exists, and the business analysts follow a goal oriented requirements engineering approach to develop and refine all the KAOS system views. In this case, business analysts can use intentional fragments to infer an initial goal model

from the business model, using the tactics we describe in Chapter 4.

3.10.5 Formal Definition

Intentional Fragments make the relation between activities and goals explicit. To define intentional fragments, we start by clarifying the structural relation between an intentional fragment and its parent process. We restrict our discussion to intentional fragments originating from single process models, although the concept of intentional fragment can be expanded to include a combination of activities from two or more processes.

We have characterised the structure of a process model in Section 3.5.3. We now define a *Potential Intentional Fragment* **IF** of a process $\mathbf{P} = (N, Start, End, \delta)$ for goal **G** as a tuple $(N', Start', End', \delta')$ such that the following criteria hold:

1. Inclusion criteria

- $N' \subseteq N$
- $Start' \subseteq N'$ is the set of nodes that have no predecessors in **IF**
- $End' \subseteq N'$ is the set of nodes that have no successors in **IF**
- $\delta' \subset N' \times N'$ is the smallest relation satisfying the following criteria:
 - if $follows(n_2, n_1)$ holds in **P**, then $follows(n_2, n_1)$ holds in **IF**
 - if $follows(n_2, n_1)$ is false in **P**, then $follows(n_2, n_1)$ is false in **IF**

2. Minimality criteria

- $(\forall n \in N'), contributes(n, G)$

3. Completeness criteria

- $(\nexists n \in N), contributes(n, G) \wedge n \notin N'$

A *Justified Intentional Fragment* **IF** of a process $\mathbf{P} = (N, Start, End, \delta)$ for goal **G** is defined as a tuple $(N', Start', End', \delta')$ such that

1. **IF** is a potential intentional fragment of **P**, as defined above
2. Completeness criteria: the execution semantic of **IF** entails **G**: $[[\mathbf{IF}]] \models \mathbf{G}$
3. Minimality criteria: there is no other **IF'** such that $[[\mathbf{IF}']] \models \mathbf{G}$ and $\mathbf{IF}' \subset \mathbf{IF}$

3.11 Alignment between Process and Goal Models

Having identified intentional fragments, even if the models have not been completely developed, business analysts can start to reason about the consistency between different system views. This type of analysis will accelerate development of the different models towards a consistent description of the system.

First, business analysts can check that for each goal identified in the goal model, there exists at least one corresponding intentional fragment. If this is not the case, analysts can decide if the process model should be further expanded to ensure the goal is satisfied, or on the contrary invalidate the goal.

Conversely, if there are process activities which are not part of at least one intentional fragment, this may indicate a superfluous activity or a goal missing from the goal model.

Finally, for each intentional fragment identified, the agent performing the process activities should be the same agent as the one responsible for the goal. If this is not the case, the analyst should reassess the responsibility assignments in the models.

3.12 Summary

Our overarching goal is to allow business analysts to evaluate and improve the alignment between the software specification and the prescribed workflows, in the context of EHR systems. In Chapter 1 we have discussed why this remains an important challenge when organisations deploy new EHR systems and Chapter 2 has informed our approach: explicitly relating a goal oriented requirements engineering framework (KAOS) with a process modelling framework (BPMN). This has several benefits: it makes process models less ambiguous; it allows business analysts to relate software requirements to process activities and process activities to business goals; it makes process models a source for new software requirements.

This chapter has presented the rationale and our proposed method to relate the two frameworks. Our approach to establish this relationship is to extend the KAOS goal oriented requirements engineering framework, adding a process model view. This decision was motivated by the fact that KAOS already provides a robust and expanding set of techniques for requirements elicitation. Furthermore, KAOS already has a formal semantic - this can be used to address current challenges with BPMN.

To develop the KAOS process view, we have taken inspiration from BPMN, but aimed to make the models more rigorous through the introduction of additional structural constraints.

In addition to the BPMN inspired graphical notation, we have also introduced a standard set of annotations for each process element. Annotations represent the way in which business analysts provide information necessary to establish the execution semantic of each element of the process model, as well as additional business expectations on the execution. The annotations are expressed in linear temporal logic and concern objects and attributes observable in the system.

In the KAOS method, four complementary models (for goals, operations, agents and objects in the system) are developed incrementally. With every change in one of the models, requirements engineers have to update the other models, following inter-model consistency rules. This ensures the four models taken together present a consistent view over the system of interest [VL01]. Since we are introducing a fifth type of model in KAOS - the process model - we present in this chapter integration rules between the new KAOS process model and each of the existing four models.

First, we integrate the process model and the object model. We introduce activities as a specialisation of KAOS objects, with a number of built-in attributes that capture the activity life cycle - whether an activity instance is active, or has already completed execution, its duration, start and end time. These built-in attributes, in conjunction with the annotations, fully describe the effects of each activity in the system.

Second, we integrate the process model and the operation and agent model. We define the execution

semantic of each activity as equivalent to the application of a start and an end operation. The signature of the two operations can be derived from the activities annotation. Agents applying the start and end operation are in effect performing the activity.

Third, we integrate the process model and the goal model. The process model can be a source of low level requirements and domain expectations, and, at the same time, it may demonstrably satisfy a goal (under any execution path). Business analysts may also evaluate a process model across a number of executions, to establish the degree in which a process satisfies quality criteria - in other words, a quantitative evaluation of the process model in regards to business goals.

Finally, we are introducing in this chapter the concept of Intentional Fragment. Intentional fragments make explicit the relationship between one or more process activities and the goal they contribute to. Intentional fragments allow business analysts to document relationships between process activities and the goals they contribute to even if the two models are not fully developed or formalised. Using the participant registration process in the WellbeingUCL study we show an example of how intentional fragments have been used in practice, to identify goals and requirements for the software system. In Chapter 5 and Chapter 6 we use action research to evaluate the application of Intentional Fragments during the requirements elicitation phase of two industry projects.

Chapter 4

Inferring Goal Models from Process Models

This chapter presents a series of heuristics applicable to process models in order to elicit goals, goal refinements, requirements and responsibility assignments. Section 4.1 presents the motivation for developing techniques to assist business analysts when developing goal models. Section 4.2 offers an overview of current methods proposed in the literature. Section 4.3 presents our approach - a semi-structured process whereby analysts gradually build a goal model by choosing from a catalogue of goal inference heuristics. Section 4.4 briefly presents the format we use to describe each heuristic, and the remaining of the chapter introduces the 12 heuristics in detail. For illustration, we continue to refer to the WellbeingUCL project.

4.1 Motivation

In our extended KAOS framework, the goal model and the business process model represent complementary views over the system under consideration. While there is an acknowledged relation between business process analysis and software analysis and design [UL13], [AMP94], BAs have difficulties adopting requirements engineering (GORE) methods and tools in their day to day practice. Goal oriented languages like KAOS are rarely used in the industry [FGZ15]. KAOS is one of the better known GORE frameworks, but novice practitioners face a steep learning curve when trying to adopt such a framework [ASM06]. Two reasons are cited for this [FGZ15]:

- it is difficult to establish correct refinement links between high level goals and software requirements;
- there is a perceived lack of methodological guidelines for inexperienced practitioners.

In this chapter, we aim to address these two obstacles. We propose that business analysts may use process models as the starting point for developing the other KAOS views (goals, agents or object models).

To provide direction in the goal elicitation process, we present a semi-structured method, that relies on business analysts selectively applying goal inference heuristics on elements of the process model. The heuristics we present in this chapter only provide guidelines into how the information implicit in a process model could be used by business analysts to formulate valid business goals. Practitioners have

to interpret the guidelines and decide when to apply each heuristic, based on their domain knowledge. Similarly, goals identified from the process model have to be validated with the stakeholders, as with any other goal elicitation technique.

The heuristics we present in this chapter can not be proven to work in all circumstances. However, they have been developed based on our experience with WellbeingUCL project, and capture rules and patterns that produced meaningful goals, in the context of that project. The heuristics have been further used in two industrial projects presented in Chapter 5 and Chapter 6. In all three cases, we have selectively applied the heuristics and constructed goal models. The resulting goal models have been validated by internal business stakeholders and have been used during the design stage of the project, to reason about software functionalities.

4.2 Related Work

In Chapter 3 we discussed related work where process models may be derived from fully specified goal models. Conversely, there exists related work addressing the challenge of deriving a goal model from a fully specified process model. [DIVGD07] and [DIVSP13] specifically propose heuristics to derive the goals from a process model, based on structural patterns in the process. Every task in the process model is represented in the goal tree, and the process itself is the highest level goal. However, these approaches do little to mitigate a common pitfall of goal modelling: confusing goals with activities and goal refinement with activity decomposition. When refining high level goals, business analysts are often guided implicitly or explicitly by the structure of the process model. Figure 4.1 illustrates this situation: the goal model has exactly the same structure as the process model, with five leaf goals (corresponding to the five activities) refining the high level goal.

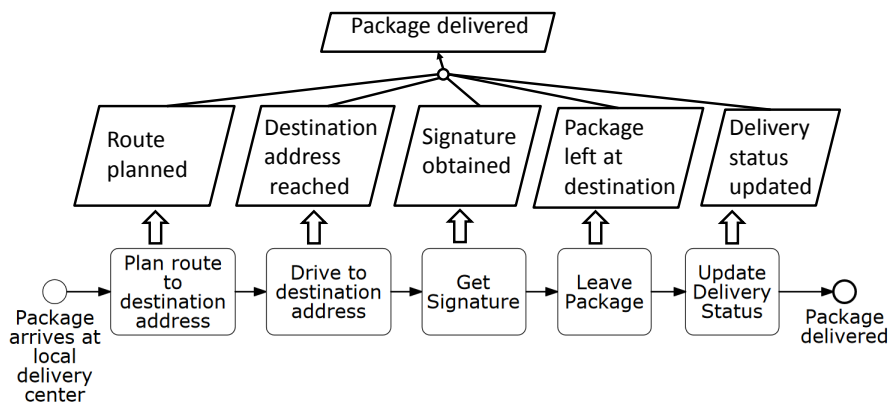


Figure 4.1: Problematic Goal Refinement from Process

However, this refinement is not representative of the real world problem domain. For example, obtaining a signature is not required in order to satisfy the goal of delivering the package. One can easily envision a scenario where a package is left at the front door and no signature is taken. Getting a signature does contribute to a goal - creating a proof of delivery - but it is not required for the delivery itself. This example illustrates that inferring goal models that simply reflect the structure of a process model are of

very little value since they merely represent the same information as that already captured in the process model.

Our approach for inferring goal models will be different and will aim to uncover the implicit rationale of activities of a process model. For example, a goal model for the package delivery process might include several high level goals and goal refinements that are not solely based on the process structure, as shown in Figure 4.2. Compared to the first example, this goal model includes several top level goals, a situation that business analysts are more likely to face in reality. We consider this refinement to be more useful because it allows business analysts to consider different options for obtaining proof of delivery.

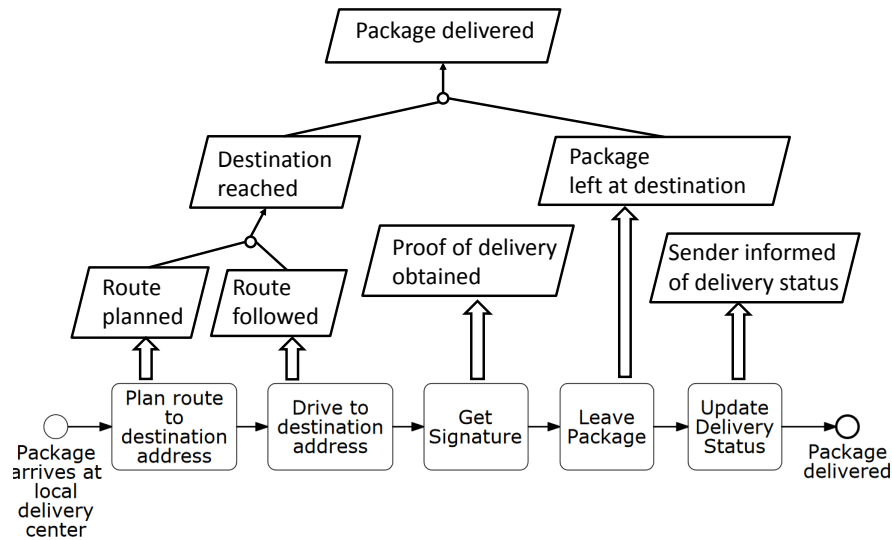


Figure 4.2: Improved Goal Refinement from Process

To guide business analysts towards meaningful and useful goal refinements, in this chapter we introduce a semi-structured approach for goal inference which is centred around intentional fragments, and not individual process activities.

4.3 Approach

The goal inference method we present in this chapter introduces a semi-structured process to infer a goal model starting from an existing process model. We describe a top down refinement process, with four main stages. Firstly, business analysts will specify high level goals for the system, taking into consideration the labels on the process gateways or the end events. Then, they will identify intentional fragments in the process model: this will yield both the main functional goals of the system, as well as non functional requirements. Next, business analysts will establish refinement links between the goals identified so far. Finally, they will refine the goals linked to intentional fragments, until each requirement can be assigned to a single agent in the system.

At each stage, business analysts have a choice of one or more specific goal inference heuristics. These heuristics (H1 to H12) are described in detail in Section 4.5, Section 4.6, Section 4.7 and Section 4.8. By applying them, business analysts will generate candidate goals. Using domain knowledge, they will then validate, modify or reject each candidate goal.

Below we briefly introduce the heuristics that can be used at each stage. In the following sections we will discuss each of them in more detail.

4.3.1 Infer Process Global Goals

The first stage of our goal inference method concentrates on identifying the high level goals of the system. Business analysts may select one or more of the available heuristics listed in Figure 4.3. The heuristics used at this stage generally do not consider process activities - the objective here is to encourage business analysts to consider the desired outcomes of the process, and not the specific work that is carried out. As such, the focus is on end event labels (H1 and H2) or the process name (H4). By definition, these are the elements that describe the outcomes of the process, and as such they are the best candidates for inferring global goals. If these elements are not labelled, business analysts can use the labels on the first and last activities of a process (H5) to generate candidate global goals. Business analysts may also use the labels on the exclusive gateways (H3) to identify high level goals of type avoid.

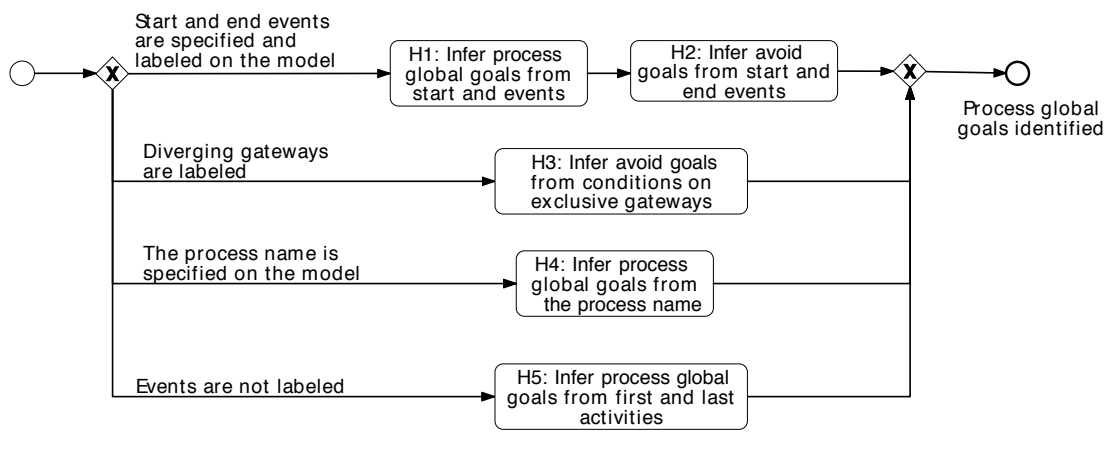


Figure 4.3: Goal Inference Heuristics: Global Goals

Identifying Avoid goals is central to a correct and complete specification of the system under consideration. Goals of type Avoid make explicit situations or events that are undesirable from a business perspective. Identifying these cases during the analysis stage of a project allows the system designers to introduce mitigation strategies, to either ensure the system prevents the undesired state, or that there is a recovery strategy in place. In turn, this makes the system more robust and predictable, as there is now an explicit behaviour that covers the scenario [VLL98].

4.3.2 Discover Intentional Fragments

Following the first stage of our goal inference method, we propose intentional fragments should then be discovered by repeatedly applying one or more of the heuristics listed in Figure 4.4.

Our approach aims to help business analysts avoid the trap of establishing one to one relations between activities and goals (a common risk for inexperienced requirements engineers). Instead, the heuristics available at this stage encourage them to reason at a more abstract level and identify a rationale (i.e. a goal) that would be applicable to a group of activities.

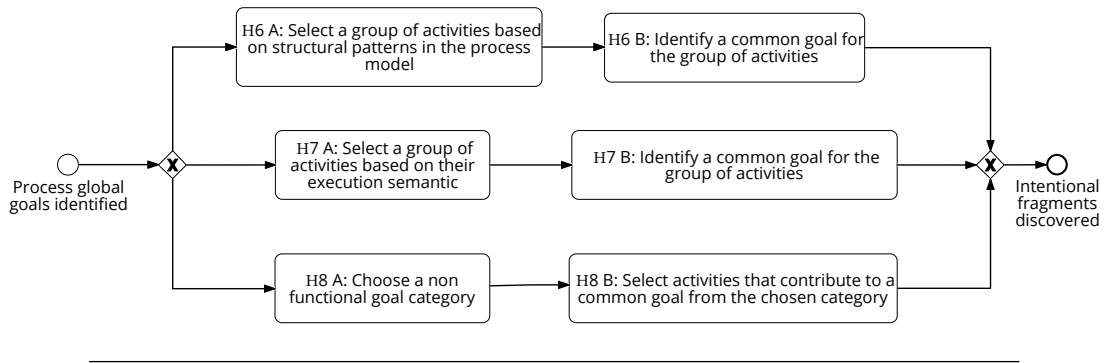


Figure 4.4: Goal Inference Heuristics: Intentional Fragments

There are two fundamental directions one can follow to discover intentional fragments. On one hand, business analysts can start by grouping two or more activities and then define a goal that would best capture their shared purpose (H6 and H7). On the other hand, business analysts can start from a catalogue of common non-functional goal classes. For each such class, analysts will then identify supporting activities (H8).

The first approach for discovering intentional fragments (starting from activities and then defining a common goal) includes two complementary heuristics. A business analyst could group activities simply based on structural patterns in the process model (H6): for example, the activities between a pair of parallel gateways will form a candidate intentional fragment. Alternatively, they can group activities based on their related semantic (H7). A starting point is looking for activity labels that refer to the same objects in the object model. If the formal execution semantic of activities has been specified through annotations, analysts could group activities based on relations among their respective pre and post-conditions. For example, a pair of activities where the post-condition of one is a pre-condition for the other will form a candidate intentional fragment - no matter their position in the process.

The second approach for discovering intentional fragments (starting from a goal and then searching for supporting activities) is illustrated by heuristic H8 in Figure 4.4. This approach has the added benefit of facilitating the discovery of *Maintain* or *Avoid* goals - analysts can start from categories of non-functional goals and select process activities that contribute to these types of goals. The KAOS framework specifies a detailed taxonomy covering different categories of goals [DVL93], including:

- Safety goals - concerned with avoiding hazardous states or maintaining safe states
- Security goals - concerned with avoiding threats to the system
- Accuracy goals - goals concerning the accuracy of the beliefs of an agent about its environment

Considering the WellbeingUCL process presented in Chapter 3, an accuracy goal for the system is ensuring the data saved about each participant is identical with the values measured by each device (weight, height, blood pressure etc.).

Similar to heuristics H2 and H3, we designed heuristic H8 in order to mitigate a tendency of business analysts new to goal modelling to solely focus on functional Achieve goals and overlook non-

functional Maintain and Avoid goals [VL09].

We also observed this tendency in the practical exercises with practitioners, organised at a French research laboratory as part of the empirical evaluation of the Intentional Fragment concept [CCML⁺12].

The output of this stage is a set of candidate intentional fragments. The heuristics described here will be applied iteratively, with two objectives in mind. Firstly, ensuring that each activity in the process is part of at least one intentional fragment (i.e. each activity has an explicit purpose). Secondly, ensuring that each goal that has been identified has an associated intentional fragment (i.e. each business goal is addressed by at least one activity).

It is possible that these two objectives are not met, pointing to a misalignment between the process and the goal model. This is an important step in the requirements elicitation process. Identifying such cases early on will help either re-engineer the process to be or re-consider some of the goals already elicited.

4.3.3 Build a Goal Model

The focus of this stage is finding refinement relations between the goals identified so far. In Figure 4.5, we introduce two heuristics that can be used to structure the goal model.

The first heuristic (H9) can be used by business analysts to refine the global goal, if this goal is a disjunction of several conditions (due to exclusive gateways that never converge). The other heuristic (H10) considers the structural relations between two intentional fragments in the process model in order to suggest the relation between the corresponding goals.

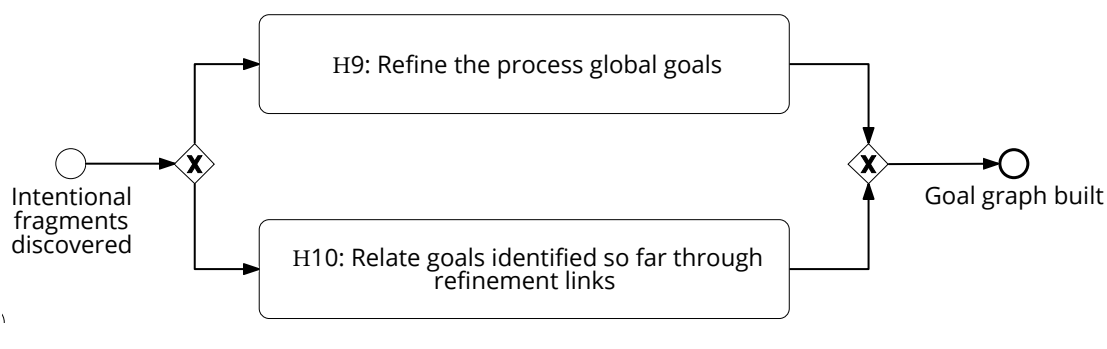


Figure 4.5: Goal Inference Heuristics: Refinement Links

As the heuristics available at this stage rely on structural patterns in the process model, business analysts should validate the goal refinements, considering any other additional sources of information and expert domain knowledge.

4.3.4 Refine the Goal Model

Business analysts will further refine the goal model until they reach individual requirements. Specifically, the goal associated to each intentional fragment should be further refined until leaf goals can be assigned to individual agents. To guide the top down refinement, structural patterns in the process model can again be used by business analysts.

We expect that at the end of the goal refinement process, each individual requirement will be related to either:

- (i) one activity or
- (ii) a group of activities performed by the same agent in an uninterrupted sequence.

This reduces the complexity of reasoning about goal satisfaction. Business analysts need only to prove that one or more activities satisfy specific leaf goals. Goal satisfaction arguments can then be incrementally built in a bottom up approach, following the refinement links present in the goal model.

However, the benefit of our approach does not lie solely in the fact that specific activities are linked to individual leaf goals. The greater benefit comes from eliciting the intermediate goals linking the process global goals to the low level requirements. These goals are essential when reasoning about the system-to-be, as they capture the business objectives embedded in a business process, while abstracting away from the details of the existing process.

4.3.5 Tool Support

We have implemented the heuristics to infer high level goals in a software tool that automates their application.

Our objective in building this tool was to demonstrate that our heuristics are defined with sufficient rigour such that their application may be automated. Furthermore, we wanted to ensure that the heuristics may be applied to a diverse range of process models and still produce meaningful goal definitions.

Implementing the tool was possible because process modelling tools such as Signavio [Sig] generate an XML representation of each diagram. This allows us to programatically access the information contained within each process diagram.

The functionalities of the tool, shown in Figure 4.6, are as follows:

- users may import a well formed process model (as XML file)
- users may select one of the available heuristic to be applied
- a goal definition is automatically generated based on the selected heuristic
- users may review and edit the goal definition

Our first objective was to demonstrate that the heuristics are well defined and are specified at a sufficient level of detail to enable their automated application. Indeed, we were able to implement the heuristics covering the process global goals and identification of possible Intentional Fragments based on the process structure. This supports our claim that the heuristics may be methodically applied by business analysts.

The tool relies on the labelling guidelines we introduced in Section 3.5.4 to produce goal definitions in natural language. We are using SimpleNLG [GR09], an open source realisation engine, to create well formed definitions.

Parsing the XML structure of the process diagram, our tool is also able to identify all instances where a certain heuristic applies.

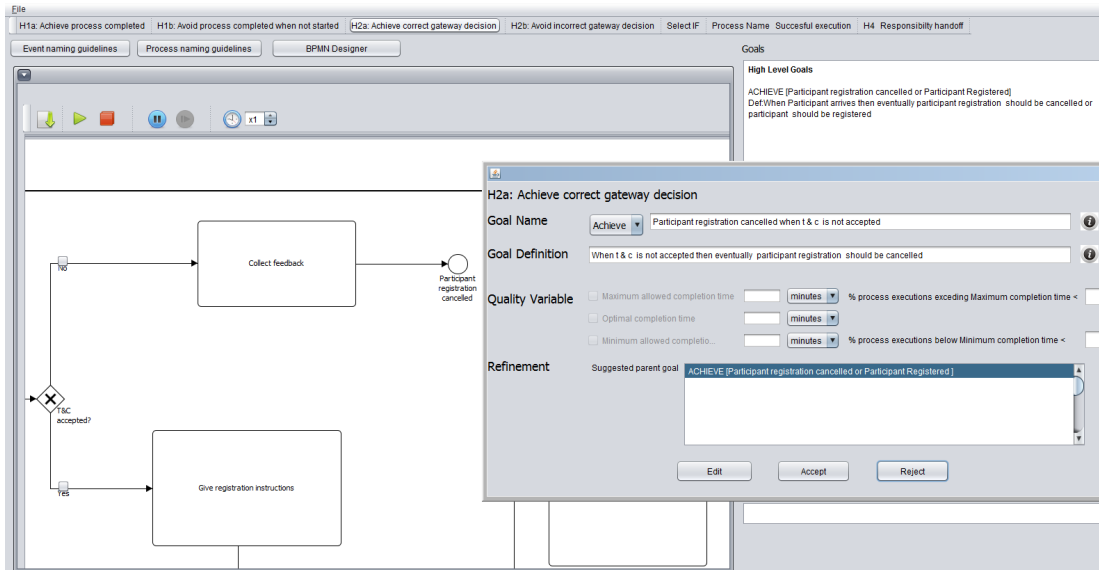


Figure 4.6: Tool Support for Goal Inference Heuristics

As expected, there are still areas where manual input is required, before a heuristics may be applied - for example, identifying exceptional end events, or specifying intentional fragments for non-functional goals. In this latter case, a business analyst has to identify activities that contribute to a goal. We also note that our focus is on early stage requirements engineering - as such, we have not implemented functionality for formal model checking.

Our second objective was to evaluate if the heuristics, once implemented in the tool, may be applied to different process models and still produce goals that are meaningful. To test this we have sourced process models related to healthcare systems from publicly available repositories [KLWW11]. We have also used existing process models developed by a UCL group of business analysts, responsible for business process re-engineering for the university. Using the tool, we have applied the goal inference heuristics H1 to H6 to the process models we had collected and discussed the resulting goals with the BA team, in a series of interviews. The feedback from the UCL business analysts demonstrated that the process global goals identified across the different process models using our method are consistent with the goals previously elicited from interviews with business stakeholders.

However, we do not aim to fully automate the process of generating a complete goal model from a process model. This is neither feasible nor desirable. For the more complex workflows included in our evaluation set, with numerous participating agents or gateways that control the process execution, the number of potential goal definitions generated by the tool grows exponentially. Rather than requesting the business analyst to review every case where a heuristic is applicable (and validate or invalidate the proposed goal), we propose that heuristics should be applied selectively by business analysts themselves. This allows them to take advantage of their domain knowledge to decide how the heuristics should be applied.

For the remainder of this thesis, we consider that business analysts will selectively apply the goal

inference heuristics, in the order that they choose, following the staged approach presented in this section.

Since the goal refinement process is driven by humans, the tool support has not been used in our case studies.

4.4 Conventions

When presenting the heuristics for inferring the goal model of a system starting from the process model, we describe each heuristic using a common template. The template is the same as the one used in the KAOS literature to describe the model elaboration tactics. Each heuristic is described by the following items:

- A *motivation* that describes the purpose and benefits
- A *pre-condition* that characterises the situations in which the heuristic can be applied.
- A *post-condition* showing the heuristic effects on the goal model and agent models, respectively.
- An *example* to show how the heuristics can be used in practice.

4.5 Heuristics to Infer the Process Global Goals

Def. A **Process Global Goal** shows a condition that should hold in the system under consideration after a process execution starts and eventually terminates. This condition should deliver value to business stakeholders.

To guide business analysts towards a correct goal model, the heuristics for process global goals are aimed at shifting the focus from process activities to process events and gateways.

4.5.1 H1: Global Goal From Start and End Events

Motivation: Goals identified through this heuristic represent the starting point when building a goal model. The heuristic captures high level business goals.

H1 focuses on capturing the process outcomes and on making explicit the business value which is delivered by a successful process execution. In line with process modelling best practices, the elements best suited to capture these outcomes are the process end events. Hence, in order to generate a definition for the global goal, this heuristic will use the labels on the process start and end events.

Figure 4.7 shows the simplified view of the WellbeingUCL registration process from which the activities and the exceptional end events have been removed. The two end events that the business analyst has categorised as expected are: *Participant Registration Cancelled* and *Participant Registration Successful*.

We draw attention on the fact that simply completing a process execution should not be considered a goal in itself. Rather, the process delivers value only because it brings about a desired outcome in the system, and that outcome represents the actual business goal that analysts should make explicit.

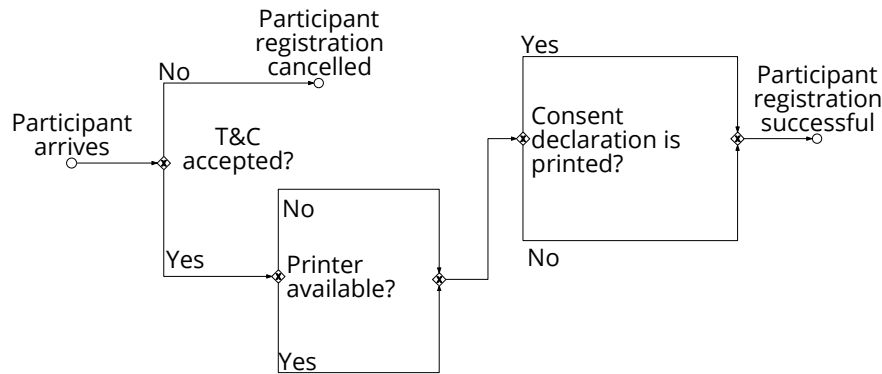


Figure 4.7: H1: WellbeingUCL End Events

Pre-condition: all the process start and end events have been identified, labelled and categorised.

In order to define the process global goals, a first condition is that all the possible end events of the process have been identified.

When constructing a process model, analysts start from the "happy path" which is the normal sequence of activities when no exceptions occur [Sil09]. The end event on the happy path is the best indication of the desired outcome from a business value perspective.

However, a complete process model will include alternative branches as well - execution paths that diverge from the happy path. Stakeholders could refer to these as workarounds, exceptional situations, unsuccessful cases, error states etc. From a process modelling perspective, all of these possible end states should be captured as distinct end events.

Each end event should then be labelled following process modelling best practices. Business analysts should ensure that event labels refer to entities from the object model.

To apply this heuristic business analysts should also distinguish between normal and exceptional end events. Using exceptional end events to capture behaviour that is out of the ordinary follows established practices in process modelling: they are akin to error end events [Sil09].

An equally important pre-condition for this heuristic is to identify and label the process start events. In many cases, the desired outcome of a process, from a business perspective, is only meaningful in a specific set of circumstances, captured by the start event. In our WellbeingUCL example, a participant should only be able to complete registration if he arrives in person at the data collection point.

Post-condition: the process global goal has been defined.

Following process modelling guidelines will often result in models with several normal end events, as the guidelines require each distinct end state be represented by a separate end event. However, we encourage business analysts to define a single process global goal, which can be refined at a later stage. This is to reinforce the fact that each process model should have one overarching business goal.

In the general case where a process model contains a combination of parallel and exclusive gateways and several normal end events, this heuristic generates a single assertion of the form $StartCondition \Rightarrow T$.

The *StartCondition* is given by the label of the start event of the process: $\langle\langle StartEventLabel \rangle\rangle$.

T is a formula with conjunctions and disjunctions of terms $\diamond e_i$ (e_i being a normal end event) that correspond to the structure of the parallel and exclusive gateways respectively. For example, for two normal end events that can be traced back to a common divergent exclusive gateway, T takes the form: $\diamond \langle\langle EndEventLabel_1 \rangle\rangle \vee \diamond \langle\langle EndEventLabel_2 \rangle\rangle$. Similarly, if two normal end events can be traced back to a common divergent parallel gateway, T takes the form: $\diamond \langle\langle EndEventLabel_1 \rangle\rangle \wedge \diamond \langle\langle EndEventLabel_2 \rangle\rangle$.

Note that we do not use the exceptional end events at this stage as these events are usually associated with risks and risk mitigation, and not with process global goals.

Example

By applying the heuristic in the WellbeingUCL example we obtain a goal defined as follows:

Goal	<i>Achieve [ParticipantRegistrationFinalised]</i>
Def	When a participant arrives, then the participant should be registered or the registration should be cancelled
Formal Def	$participantArrives \Rightarrow \diamond participantRegistration.Succesful \vee \diamond participantRegistration.Cancelled$

At this stage, without any information about the expected occurrence rate for each of the two end events, the goal definition does not capture any business preference towards either of the two possible outcomes.

This may be the case in reality, as one of the main aims of the study was simply to raise awareness and test public perceptions. On the other hand, business analyst may clarify the business expectations, and if a certain positive response rate is expected, this can be added as a quality variable once the goal is refined.

4.5.2 H2: Avoid Goals From Exceptional End Events

Motivation Business analysts should apply this heuristic in order to identify undesirable system states, which should be avoided.

The heuristic focuses on the exceptional end events of a process model. Exceptional end events have been introduced in Section 3.6.3 - they depict inconsistent, unsafe or undesired states in the system (from a business perspective). This information is made explicit in the goal model by defining corresponding goals of type *Avoid*.

Identifying these goals during requirements elicitation and including them in the KAOS goal model improves the robustness and reliability of the system-to-be. Once identified, business analysts can further refine these goals, and thus define strategies for mitigating the risk associated with exceptional end events.

Figure 4.8 shows the start and the exceptional end event of the WellbeingUCL registration process.

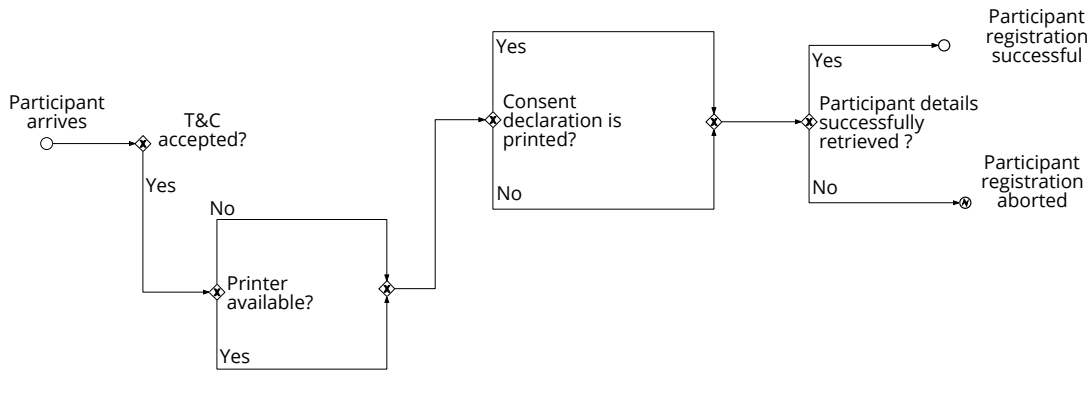


Figure 4.8: H2: WellbeingUCL Exceptional End Events

Pre-condition This heuristic may be applied in conjunction with heuristic H1, as they both operate on the same process elements: namely start and end events. Heuristic H2 can only be applied when the business analyst has identified one or more exceptional end events.

As with heuristic H1, the exceptional end events should be labelled, using objects from the KAOS object model and observable and measurable attributes of these objects.

Post-condition: High level avoid goals have been identified

In contrast with heuristic H1, this heuristic should be applied to each exceptional end event separately. We encourage this approach because exceptional end events could be quite different in nature and each associated *Avoid* goal would have a different goal refinement tree, specifying mitigation strategies.

When a process model contains an exceptional end event, this heuristic generates a single assertion of the form $StartCondition \Rightarrow \square \neg EndCondition$ based on the start and the exceptional end event labels.

Example

By applying the heuristic in the WellbeingUCL example we obtain a goal defined as follows:

Goal	<i>Avoid [ParticipantRegistrationAborted]</i>
Def	When a participant arrives, then the participant registration should never be aborted
Formal Def	$participantArrives \Rightarrow \square \neg participantRegistrationAborted$

The goal is indeed relevant because registrations that are aborted (as opposed to declined by prospective participants) can be considered a measure for how reliable the WellbeingUCL infrastructure and processes are. The current goal definition is an idealised one. To make the system more robust, business analysts can try to identify obstacles for this goal (i.e. identify a set of circumstances where a participant arrives at the data collection point and consequently his registration is aborted). Once an obstacle has been formulated, mitigation strategies may be devised.

4.5.3 H3: Avoid Goals From Exclusive Gateway

Motivation: Business analysts should apply this heuristic in order to elicit additional *Avoid* goals that further clarify the expected system behaviour.

This heuristic draws on the fact that in a process model, some of the exclusive divergent gateways may have outgoing branches that never converge again. Instead, these branches would lead to distinct end events.

From a requirements engineering perspective, it is equally important to make explicit when a certain end event should occur (through a functional *Achieve* goal) as well as when a certain end event should not occur (through an *Avoid* goal). This heuristic covers the latter concern. It is capturing the business expectation that if certain conditions (expressed through labelled gateways) have not been met during a process execution, then some of the process end events should not occur either.

Pre-condition: all the process start and end events have been identified, labelled and categorised; gateways where the execution flow diverges have been identified and labelled.

This heuristic focuses on the relation between process events and the gateways where the execution flow diverges, never to converge again.

For example, considering the process elements shown in Figure 4.7, this heuristic is not applicable to the gateway labelled *Printer available?*. The reason is that the outgoing sequence flows of that gateway do not lead to distinct end events; instead, the two branches eventually converge.

This heuristic is applicable to the gateway labelled *T&C accepted?*, as the two outgoing sequence flows in this case eventually lead to two distinct end events.

Post-condition For each exclusive gateway labelled with a guard condition whose outgoing branches lead to two normal end events e_i and e_j , with e_i occurring when the guard condition is true and e_j occurring when the guard condition is false, this heuristic generates two assertions of the form $StartCondition \wedge GuardCondition \Rightarrow \Box \neg EndCondition_j$ and $StartCondition \wedge \neg GuardCondition \Rightarrow \Box \neg EndCondition_i$. As before, the *StartCondition* is given by the label of the start event of the process, each *EndCondition* by the corresponding label of an end event and the *GuardCondition* by the label of the gateway.

Example Figure 4.7 shows the start and the normal end events of the WellbeingUCL registration process: *Participant Registration Successful* and *Participant Registration Cancelled*.

By applying the heuristic in the WellbeingUCL example we obtain a goal defined as follows:

Goal	<i>Avoid [ParticipantRegistrationSuccessfulWhenT&CNotAccepted]</i>
Def	When the participant has not accepted the T&C then the registration should never be successful,
Formal Def	$participantArrives \wedge \neg T\&C\ accepted \Rightarrow \Box \neg participantRegistrationSuccessful$

The goal is important as it reinforces the business need to be able to check whether a potential participant in the survey has accepted the terms and conditions before declaring their registration successful.

4.5.4 H4: Global Goal From the Process Name

Motivation: Heuristic H4 can be applied as an alternative to H1, in order to identify the process global goal. Following process modelling best practices, the process name should offer a good indication of the overall business purpose of the process. As such, it is also a good candidate for the process global goal.

Pre-condition: the process start event has been labelled and the process itself has been named.

This heuristic is useful in cases where the end events of the process have not been labelled. In addition, H4 should also be used if the process has too many end events. In this latter case, applying heuristic H1 leads to goal definitions that are too complex, hindering understanding. H4 provides a way to focus on the most relevant business outcome, derived from the process name.

Post-condition: the process global goal has been identified.

Heuristic H4 leads to similar goal definitions as H1, following the structure $StartCondition \Rightarrow T$. In this case, the final state T is given by the process name. In effect, the goal definition would be derived as $\langle\langle StartEventLabel \rangle\rangle \Rightarrow \diamond \langle\langle ProcessNameLabel \rangle\rangle$.

4.5.5 H5: Global Goals From the First and Last Activities

Motivation: Heuristic H5 can be applied as an alternative to H1 or H4, in order to identify the process global goal. This heuristic is less likely to produce a correct goal definition, as the labels of the last activities in the workflow may not be representative of the actual business purpose of a process.

Pre-condition: the first and last activities in the process have been identified and labelled, following the guidelines we introduced in section 3.5.4

Post-condition: the process global goal has been identified.

Heuristic H5 operates in the same manner as H1. The goal definition will again follow the structure $StartCondition \Rightarrow T$, but in this case the terms that compose the formula T are activity labels rather than event labels.

Example

We give below the goal definition a business analyst would obtain by applying H5 to the WellbeingUCL project. We stress the fact that even if the heuristic is applicable, the resulting goal definition could be rejected by the business analysts, if it is not considered representative of a business goal.

Goal	<i>Achieve [FeedbackCollectedOrConsentDeclarationArchived]</i>		
Def	When a participant is informed of the terms and conditions, then her consent declaration should be archived or her feedback collected		
Formal Def	$participantInformedOfT\&C$	\Rightarrow	$\diamond consentArchived \vee \diamond feedbackCollected$

We note that this goal definition is less informative compared to the results of alternative heuristics.

4.6 Heuristics to Discover Intentional Fragments

The first stage of our approach has dealt with the problem of correctly identifying the process global goal, as well as additional high level avoid goals. Correspondingly, the heuristics introduced so far have mainly focused on events and gateways.

In this second stage of our method, business analysts will address the next level of detail in the top down approach for inferring a goal model. The aim of this stage is to elicit new goals, that are either contributing to the global goal, or represent additional high level concerns in the system.

The heuristics applicable at this stage facilitate the discovery of intentional fragments, and through them, the discovery of new goals. Intentional fragments have been introduced in Section 3.10. In the extended KAOS framework, they represent a way to reason about the rationale of process activities - allowing business analysts to explain *Why?* is each activity necessary.

The heuristics we introduce at this stage will enable business analysts to identify intentional fragments following two complementary approaches. On one hand, starting from the process model (as the source of implicit domain knowledge) and moving towards the discovery of system goals (heuristics H6 and H7). On the other hand, starting from a catalogue of possible goals and then trying to identify activities contributing to them, if any (heuristic H8).

4.6.1 H6: Intentional Fragments From Process Structure

Motivation: business analysts should apply this heuristic in order to identify additional goals, after the process global goal has been identified.

This heuristic should be used when the process model is well-formed and structurally complex. Complexity is given by the number of participating agents, the number of gateways, the number of sequence flows crossing between pools, the number of different end events etc.

This heuristic is meant to help business analysts avoid a common pitfall when eliciting system goals: the tendency to focus on the labels of process activities and immediately transform them into goal definitions. On the contrary, heuristic H6 does not directly take into account the labels or annotations of the process elements. As such, it can be applied even when process elements are poorly or incompletely labelled (guidelines for well structured process elements have been discussed in Section 3.5 and 3.6).

Heuristic H6 proposes a set of structural patterns which business analysts should try to match against the process model. Process fragments that match these patterns are candidate intentional fragments. The set of structural patterns we present here has been informed by our own experiments, and is not exhaustive. First, we analysed the WellbeingUCL process models and observed that if we select process activities based on structural characteristics of the overall process model, in many cases we can find a common goal for these activities. For example, the activities between a pair of corresponding gateways were likely to share the same goal. Second, we used these patterns on a set of process models produced by UCL business analysts. The purpose was to check if the activities selected from the process (matching one of the patterns) consistently shared a common goal. We worked with the business analysts to validate these goals, and retained the structural patterns that helped us identify valid goals. Once candidate intentional fragments have been identified, it is the responsibility of the business analyst to define a goal for the intentional fragment, if the activities do share a common goal. As with all the other heuristics, a business analyst could find that the activities extracted from the process do not share a goal, and as a consequence an intentional fragment does not actually exist. We have stated in the previous chapter that an intentional fragment requires a validated goal.

Pre-condition: the process model follows the modelling constraints introduced in section 3.5.3.

This heuristic requires one or more of the following structural patterns:

- P1. *The process nodes immediately before and after an exclusive gateway*
- P2. *The process nodes between a pair of corresponding gateways (one divergent and one convergent)*
- P3. *The process nodes between a start event and a gateway*
- P4. *The process nodes between a divergent gateway and an end event.*
- P5. *The process nodes between a start event and a sequence flow crossing to a different pool*
- P6. *The process nodes between two sequence flows crossing back and forth between the same two pools*
- P7. *The process nodes between a sequence flow crossing from another lane and an end event*
- P8. *The process nodes connected by a sequence flow that crosses lanes.*
- P9. *A sequence of activities directly connected by sequence flows, all of them in the same lane*
- P10. *A sequence of activities connected by sequence flows and gateways, forming a loop*

Post-condition

The process nodes matching any of the patterns will form a candidate intentional fragment. Business analysts will then have to specify a common goal, if one exists, for the set of nodes. Since these patterns can be commonly found in process models, the expertise and domain knowledge of the business analyst is essential. She is expected to be able to filter through the candidate intentional fragments and identify those that capture a business goal.

The goal definition will consider the labels on activities, but there is no one single derivation rule that can be followed. One approach is to look for nouns (or synonyms) that appear in more than one activity label. This indicates the intentional fragment focuses on a specific object in the object model. Another approach is to check if the actions in the activity labels share the same semantic family. This indicates the focus of the intentional fragment is on performing a certain operation whenever necessary, during a process execution. The final goal definition will ultimately rely on the domain knowledge of the business analyst.

The intentional fragments identified through this heuristic should conform with the minimality and completeness criteria defined in Section 3.10.5. By checking against these criteria, business analysts may include additional activities in the intentional fragment, or remove some of the existing ones.

Example Searching for instances of patterns P2 and P8 on the WellbeingUCL process we obtain the two intentional fragments shown in figure 4.9.

For the first intentional fragment, which matches pattern P2, the goal can be defined as follows:

Goal	<i>Achieve [PersonalInformationCollected]</i>
Def	Demographic data should be collected for each participant to the WellbeingUCL survey

Both activity labels refer to 'registration'. The goal definition makes explicit what is meant by registration: collecting initial personal information for each participant.

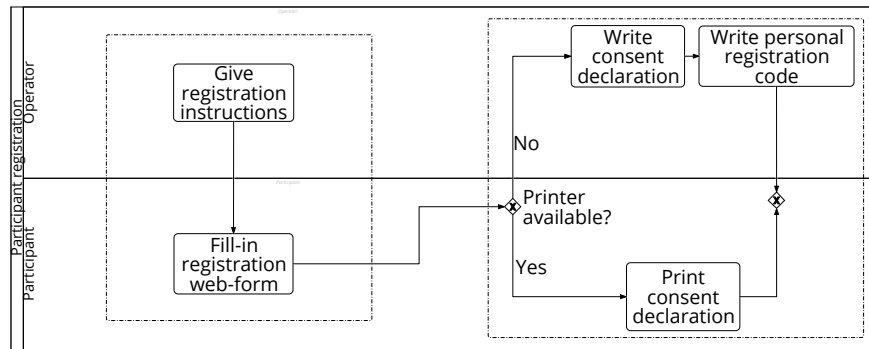


Figure 4.9: H6: Wellbeing UCL Structural Patterns

The intentional fragment based on pattern P8 uncovers a requirement around participant consent, as consent is mentioned in two of the activities' labels.

Goal	<i>Achieve [LegalRequirementsSatisfied]</i>
Def	Each participant to the survey should have a personal consent declaration

A business analyst can informally question the completeness and minimality of the intentional fragment, based on her domain knowledge. Because giving a consent declaration generally requires signing the form as well, this particular intentional fragment will have to be expanded, to include the Sign consent declaration activity. A more formal approach towards reaching the same conclusion is presented in the next heuristic, which considers the execution semantic of activities.

4.6.2 H7: Intentional Fragments From Activities Annotations

Motivation: business analysts should apply this heuristic in order to identify additional goals, after the process global goal has been identified. Heuristic H7 identifies candidate intentional fragments from annotations on the process activities.

The two heuristics, H6 and H7, complement each other. Applying H6, business analysts identify candidate intentional fragments containing process nodes that are in close proximity of each other, from a structural point of view. Applying H7, business analysts identify candidate intentional fragments containing process nodes that are dispersed throughout the model, but linked through their execution semantic.

Pre-condition: the nodes in the process model should be annotated following the guidelines presented in Section 3.6.1, with pre and post-conditions.

When writing an activity annotation, business analysts should start by writing all the post-conditions of an activity - in other words, they should describe all the changes in the system state brought upon by performing the activity. Business analysts will then question what is the benefit of each post-condition they have identified. Those post-conditions for which a purpose can be identified will become required post-conditions and the goal they contribute to should be made explicit. Business analysts will then write the activity pre-conditions, and check which pre-conditions should be classified as required pre-conditions contributing to any of the goals identified so far.

Heuristic H7 can be applied if there is at least one activity in the process whose annotations include a required post-condition for a goal.

Post-condition

Given activity A with a required post-condition for goal G , there is a candidate intentional fragment for G that contains activity A , and any other activity with a required pre or post-condition for goal G .

We note that it is not always the case that the candidate intentional fragment ensures goal satisfaction. It may be that a process model is missing activities required to satisfy the goal.

Example

As an example, we continue with the intentional fragment elicited by applying heuristic H6. This now includes the three activities between the gateways (based on structural pattern P8) as well an additional activity after the gateway, added by the business analyst based on his domain knowledge. The intentional fragment is shown in Figure 4.10.

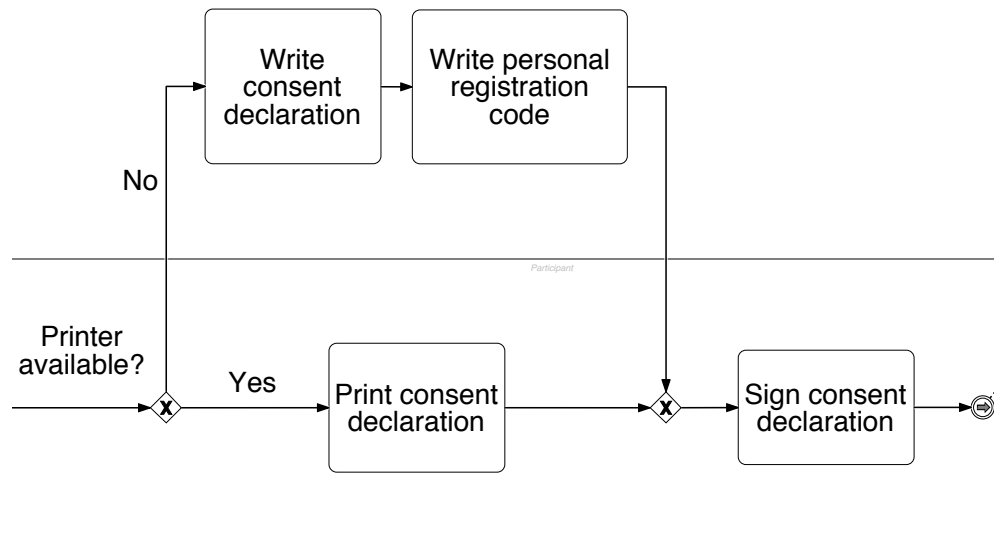


Figure 4.10: H7: WellbeingUCL Consent Declaration

We now consider the annotations of two activities, relevant for verifying the minimality and completeness of this fragment: Write Personal Registration Code and Archive Consent Declaration.

Activity	<i>WritePersonalRegistrationCode</i>
Domain pre-condition	consentDeclarationWritten \wedge \neg registrationCodeWritten
Required pre-condition for goal <i>Achieve[ParticipantIdentityKnown]</i>	registrationCodeGenerated
Required post-condition for goal <i>Achieve[ParticipantIdentityKnown]</i>	registrationCodeWritten
Maximal allowed duration	5 seconds
Maximal allowed delay	5 minutes

Activity	<i>ArchiveConsentDeclaration</i>
Domain pre-condition	participantDetailsVerified \wedge \neg consentFormArchived
Required pre-condition for goal <i>Achieve[LegalRequirmentsSatisfied]</i>	consentDeclarationSigned
Required post-condition for goal <i>Achieve[LegalRequirmentsSatisfied]</i>	consentDeclarationArchived
Maximal allowed duration	30 seconds
Maximal allowed delay	1 minutes

Based on the activities annotations, because the activity Write Personal Registration Code does not contribute to the goal *Achieve[LegalRequirementsSatisfied]* (i.e. does not have any required pre or post-conditions for that goal), it should be removed from the intentional fragment, to conform with the minimality criteria. On the other hand the activity Archive Consent Declaration has a required post-condition for the goal, so it should be part of the fragment, to conform with the completeness criteria.

The resulting intentional fragment for goal *Achieve[LegalRequirementsSatisfied]* is shown in Figure 4.11

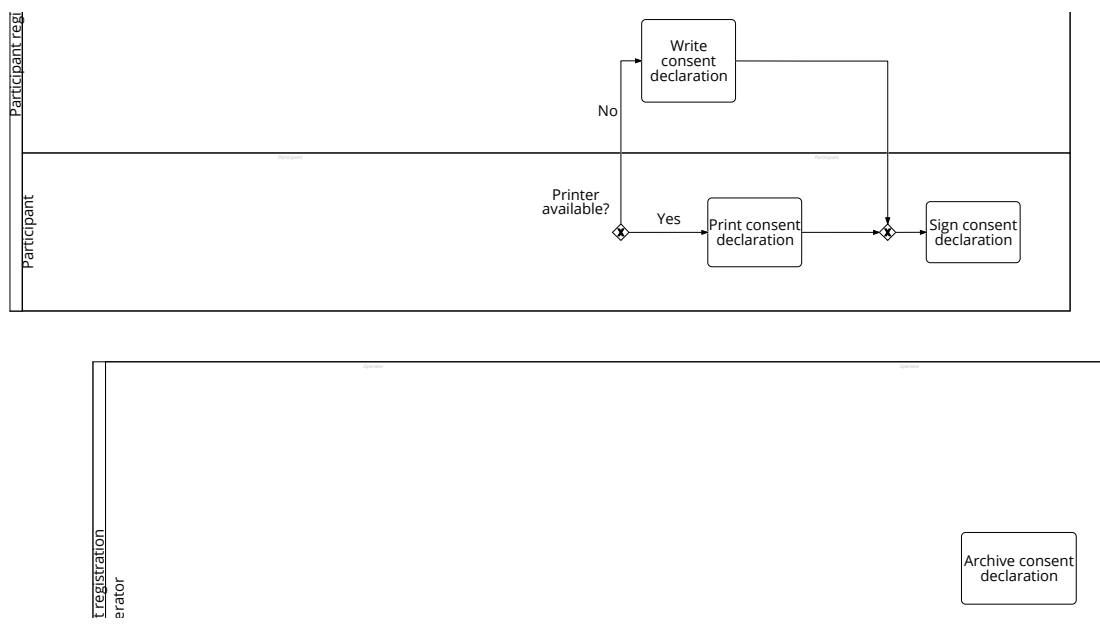


Figure 4.11: H7: WellbeingUCL Consent Declaration (Complete)

4.6.3 H8: Intentional Fragments From Non-Functional Goal Categories

Motivation: business analysts should apply this heuristic to identify additional non-functional goals.

The heuristics presented so far have focused on helping business analysts identify functional goals of the system. However, as discussed in Section 2.2.1, a KAOS goal model may also include non-functional goals. Requirements engineers can use taxonomies describing classes of non-functional goals in order to identify instances relevant for the system under consideration [CNYM12]. Most common classes of non-functional goals are:

- quality requirements related to safety, security, accuracy or performance concerns
- compliance requirements, linked to laws, regulations standards or cultural expectations
- architectural requirements, related to integration with external systems or software or hardware constraints

This heuristic is of particular importance as it helps business analysts identify additional goals, which otherwise may be overlooked. They can thus make explicit the rationale of process activities that do not contribute to functional goals, but support valid non-functional requirements.

Pre-condition: the process activities have been labelled.

This heuristic can be easily applied in cases where there are process nodes with labels that contain words within the semantic family of the non-functional goal categories: confidentiality, integrity, availability, security, compliance, interface, accuracy etc. If this is not the case, business analysts may still use the non-functional goal categories as a reference to review each process activity and verify if a contribution relation may be established.

Post-condition: all the activities contributing to the same non-functional goal category will form a candidate intentional fragment.

Example: The WellbeingUCL process contains one activity labelled *Verify Participant Details*. This points to an accuracy goal, defined as follows:

Goal	<i>Maintain [AccurateParticipantDetails]</i>
Def	The information saved for each participant in the survey should reflect their actual details

Of course, to ensure goal satisfaction, additional requirements may be necessary. In the next section we discuss how the goals identified so far can be structured into a goal model, which would then enable further refinement.

4.7 Heuristics to Build a Goal Model

The third stage in our approach deals with structuring the goals identified so far into a goal model, by establishing refinement and contribution links.

4.7.1 H9: Refining the Global Goals

Motivation: Business analysts should apply this heuristic to refine global goals.

In cases where a process model has two or more normal end events, we recall that heuristic H1 generates a single assertion of the form $StartCondition \Rightarrow T$, where T is a formulae with conjunctions and disjunctions of terms $\diamond e_i$ (e_i being a normal end event).

By applying heuristic H9, business analysts will further refine this global goal to clarify what are the precise conditions leading to each of the possible end events.

Pre-conditions: the process model contains a divergent gateway with outgoing sequence flows leading to distinct end events; the gateway and the end events have been labelled.

A pre-condition for using heuristic H9 is that H1 has already been applied. The business analyst should have validated a process global goal of the form $StartCondition \Rightarrow T_1 \vee T_2$. She should also identify and validate the guard condition C which determines the disjunction. If the gateways have been labelled, C is given by the label on the exclusive gateway that leads to the distinct end events.

If T_1 or T_2 is a formula that in turn contains a disjunction of terms, heuristic H9 will be applied recursively.

Post-conditions: The process global goal will be refined into three sub-goals.

The first goal takes the form $StartCondition \Rightarrow C \vee \neg C$. The two other goals are defined as: *Achieve* [T_1 if C] and *Achieve* [T_2 if $\neg C$].

Example By applying heuristic H1, we have identified the process global goal for the WellbeingUCL survey :

Goal	<i>Achieve</i> [<i>ParticipantRegistrationFinalised</i>]
Def	When a participant arrives, then the participant should be registered or the registration should be cancelled
Formal Def	$participantArrives \Rightarrow \diamond participantRegistrationSuccessful \vee \diamond participantRegistrationCancelled$

By applying heuristic H9, this is refined into three sub-goals, defined as follows:

Goal	<i>Achieve</i> [<i>T&C Accepted or Not Accepted</i>]
Def	When a prospective participant arrives, then she should either accept or refuse the T&C
Formal Def	$participantArrives \Rightarrow \diamond T\&C\ Accepted \vee \diamond \neg T\&C\ Accepted$

Goal	<i>Achieve</i> [<i>ParticipantRegisteredIfTermsAccepted</i>]
Def	When a participant accepts the terms and conditions, then the participant should be registered
Formal Def	$T\&C\ Accepted \Rightarrow \diamond participantRegistrationSuccessful$

Goal	<i>Achieve</i> [<i>ParticipantRegistrationCancelledIfTermsNotAccepted</i>]
Def	When a participant does not accept the terms and conditions, then the participant registration should be cancelled
Formal Def	$\neg T\&C\ Accepted \Rightarrow \diamond participantRegistrationCancelled$

4.7.2 H10: Relation Between Intentional Fragments

Motivation: This heuristic should be used to establish refinement links between goals identified so far, through the intentional fragments.

Pre-conditions: this heuristic is applicable if the business analyst has already identified a pair of intentional fragments IF_1 and IF_2 , such that $\llbracket IF_1 \rrbracket \models G_1$ and $\llbracket IF_2 \rrbracket \models G_2$ and $IF_1 \subset IF_2$.

In other words, H10 may be applied when the business analyst has identified two intentional fragments contributing to different goals, where one fragment is entirely included in the other one.

Post-condition: a refinement link is established between the two goals, with G_1 contributing to G_2 if $IF_1 \subset IF_2$. As with all the other heuristics, business analysts may reject the proposed refinement link, based on domain knowledge.

Example: We illustrate this heuristic on an intentional fragment of the WellbeingUCL process linked to data collection. The high level goal has first been shown in Figure 3.17, while the two sub-goals shown in Figure 4.12 have been identified by applying heuristics H7 and H8. We note that although the refinement links are valid in this case, the refinement is not complete - the goal model does not yet contain goals dealing with a situation where the participant details have not been recorded correctly.

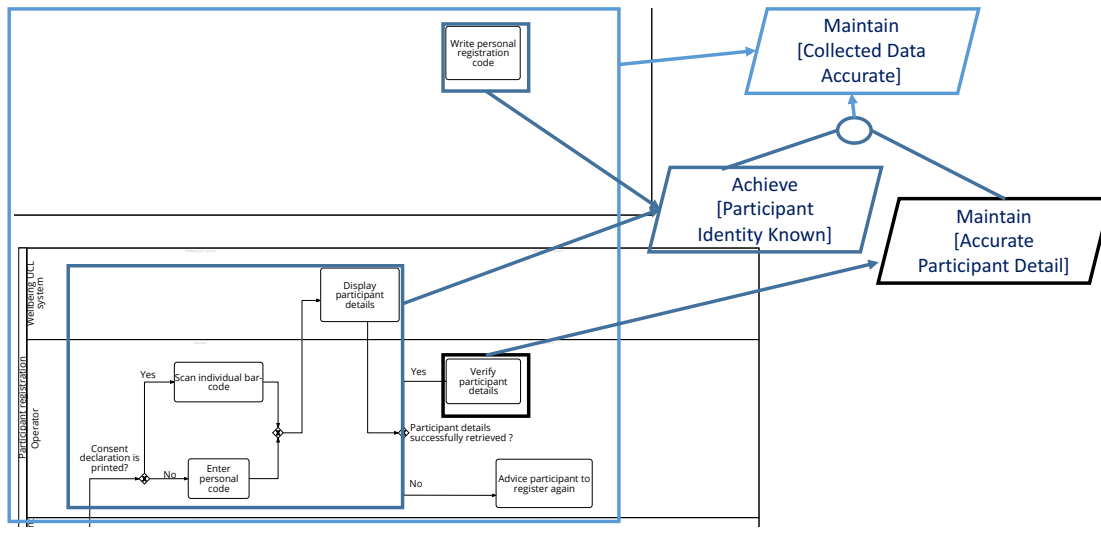


Figure 4.12: H10: WellbeingUCL Goal Refinement

4.8 Heuristics to Refine the Goal Model

The last stage of our approach includes heuristics that will help business analysts further refine the goals linked to intentional fragments. Specifically, if there are intentional fragments where at least two agents are responsible for the activities contained within, these could be further refined. Once business analysts reach the level of detail where the intentional fragments contain activities performed by only one agent, they have reached the leaf goals in the model.

Of course, the goal model may still be incomplete, but there is no additional information that could be derived from the process model. Further refinements rely on the domain knowledge of the business

analysts and on the other requirements elicitation techniques available in KAOS.

4.8.1 H11: Case Refinement of an Intentional Fragment

Motivation: this heuristic should be used when an intentional fragment has been identified, that includes alternative execution paths.

Pre-condition: for this heuristic to be applicable, business analyst should have already identified an intentional fragment that contains an exclusive gateway. The two outgoing sequence flows should fall under the responsibility of two distinct agents.

Post-condition: the goal linked to the intentional fragment will be refined using a case refinement pattern. The two sub-goals will be linked to two intentional fragments, where the activities of each fragment will be performed by one single agent. The guard condition is given by the gateway condition.

4.8.2 H12: Milestone Refinement of an Intentional Fragment

Motivation: this heuristic should be used when an intentional fragment has been identified, that includes activities performed by two different agents.

Pre-condition: for this heuristic to be applicable, business analyst should have already identified an intentional fragment where the activities composing the fragment are performed by at least two different agents, and the activities are performed in sequence.

Post-condition: the goal linked to the intentional fragment will be refined using a milestone refinement pattern. For each responsibility hand-over, signified by a sequence flow that crosses two different lanes in the process model, there will be an additional refinement of the goal. This will result in $N+1$ sub-goals, where N is the number of sequence flows crossing lanes.

4.9 Final Goal Model for the WellbeingUCL process

In Figure 4.13 we present the goal refinement of the main functional goal for the participant registration process in the WellbeingUCL project. Figure 4.14 shows additional goals and their respective refinement identified through some of the other heuristics presented in this chapter. We note that the final goal model was also influenced by our domain knowledge, our choice of how to apply the heuristics, and our own preference in naming the activity labels and the goals. This goal model is representative of the stakeholders' intents, but at the same time another business analysts following our method could have produced a model slightly different, but equally useful.

This goal model can now be used to reason about the suitability of the process model and to identify software requirements which have been overlooked. For example, a more advanced search function could reduce the number of aborted registrations caused by failure to retrieve a participant details. In addition, a functionality to edit personal details entered during registration would contribute towards maintaining accurate participant details.

The goal model also highlights incomplete goal refinements - in other words, areas where the current process model does not satisfy the intended goals. For example, the goal of cancelling a registration if the participant does not accept the terms and conditions is not fully refined in the goal models, as the process model lacks the necessary detail.

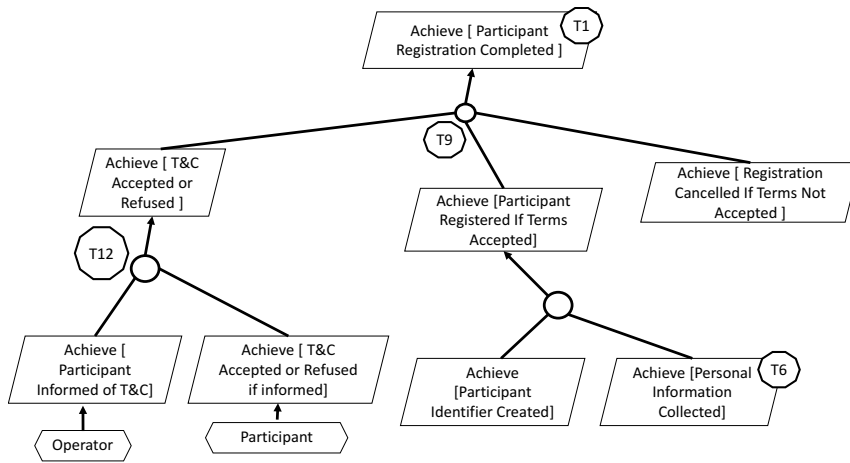


Figure 4.13: WellbeingUCL Goal Model

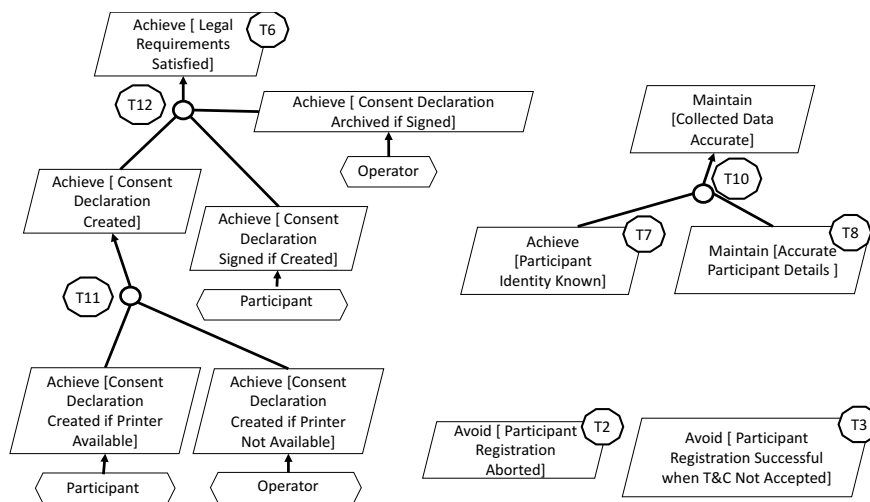


Figure 4.14: WellbeingUCL Additional Goals

4.10 Summary

In Chapter 3 we proposed that BAs will analyse the alignment between business processes and software requirements using the extended KAOS framework. The contributions presented in Chapter 4 are motivated by the fact that goal oriented requirements engineering in general and the KAOS framework in particular are not widely adopted in the industry, by business analysts.

To facilitate adoption, we have introduced in this chapter a requirements elicitation method where BAs start from process modelling (a familiar technique) and gradually build the other KAOS models (a more challenging task if attempted with no guidance).

Our goal inference method has four stages, and follows a top-down approach - business analysts start from high level goals and aim to refine these until they reach individual requirements.

At each stage, business analysts will choose one or more heuristics to apply on the process model. The heuristics provide a template used to identify elements of the KAOS goal model (goals, requirements, refinement links) from elements of the KAOS process model (events, gateways, activities).

The method is semi-structured - the decision on how to apply each heuristic lies with the analysts. We have demonstrated how tool support could automatically generate candidate goal definitions in simple cases, but a fully automated approach does not scale. Furthermore, when building a goal model, additional sources of information would be available to business analysts: the knowledge of domain experts, interviews with users, service level agreements etc. As such, process models merely represent the starting point in elaborating the goal model.

The heuristics cover a wide range of concerns. First, heuristics 1 to 5 propose different techniques to formulate high level goals. We explicitly guide business analysts away from the process activities at this stage. Instead, the heuristics concentrate on the process end events, and on the business logic (expressed through gateways) defining which event should happen under which circumstances and which events should be avoided altogether.

Second, heuristics 6 and 7 propose two complementary ways in which process activities could be grouped and their common goal be made explicit. On one hand, a process model can be divided into fragments based on structural patterns - for example, the group of activities between a pair of gateways are likely to contribute to the same goal. On the other hand, the activities composing a process model can be grouped based on their execution semantic and irrespective of their relative position in the process flow (i.e. whether the activities follow each other, or are dispersed).

Third, heuristic 8 asks BAs to consider categories of goals, from a standard KAOS catalogue, and establish if any process activity can be said to contribute to any of the goal categories. This approach has the advantage of guiding business analysts towards considering higher level, non-functional goals.

Fourth, heuristics 9 to 12 use structural patterns in the process model to propose refinement links between goals. The heuristics cover a range of situations: refining the process global goals; relating goals identified through intentional fragments; refining goals until they can be assigned to individual agents (in effect, until BAs reach the level of software requirements).

Our approach aims to prevent business analysts from creating a goal model that represents exactly

the same information contained within the process model (a one-to-one, deterministic mapping between activities and goals). This has been one of the main challenges emerging from the literature review concerning requirements elicitation from process models. Using the WellbeingUCL project, we have illustrated how the final goal model elicited using our approach has uncovered additional insight, beyond the knowledge implicit in the process model.

Chapter 5

Evaluation: Electronic Healthcare Record for Bupa

This chapter presents an evaluation of our theoretical contributions in a project aimed to deliver an EHR implementation for a health community service with over 2,600 enrolled out-patients. Section 5.1 presents the research questions we address. Section 5.2 introduces the research methodology - action research - and the rationale for choosing it. The remainder of the chapter follows the structure of action research projects. In Section 5.3 we present the organisational context, in Section 5.4 we diagnose the main challenges of the project and Section 5.5 discusses how our method was used in practice to address these challenges. Section 5.6 presents the final system, used to support the Bupa lead community service for COPD. Section 5.7 reviews the outcomes of the project, after implementation, in light of our original research questions.

5.1 Research Questions

In previous chapters we have introduced goal oriented modelling and analysis techniques applicable to business processes. Our aim was to cover a gap we have identified from the literature review: business analysts lack a rigorous way to relate business process models to software requirements. To help analysts ensure the software under development is closely aligned to the users' workflows, we have presented three specific contributions:

C1: an extension of the KAOS requirements engineering framework, to allow for business process modelling.

C2: the concept of Intentional Fragment which captures an explicit relation between fragments of a process and a specific goal.

C3: goal inference techniques to help analysts build goal models starting from process models.

To validate our method in a real life setting we use the Bupa COPD project, in which the author was directly involved. Using Action Research as a mode of enquiry and the quantitative and qualitative data collected throughout the project lifetime, we aimed to answer three specific research questions:

RQ1: How successful was the EHR deployment?

First, we analyse to what degree the EHR system, once deployed, mitigated the challenges com-

monly reported in the industry - we consider emerging workarounds, user adoption rate and the clarity of the benefit case.

RQ2: To what degree did our method contribute to the project outcomes and to its impact?

It is crucial to understand if and how the techniques we have introduced in the previous chapters can be used in a real business setting, and whether they do indeed impact the project outcomes.

To evidence how our method has contributed to the success of the project, we will discuss how the techniques introduced in this thesis have been used at different stages as the project unfolded. The outputs of these techniques (KPI definitions, new requirements) were used to take critical design decisions, as well as shape and manage the project, from initial proposition to delivery.

RQ3: How does our method fit with the organisational context within Bupa and what was the impact on the business analysts community and practice?

It is important to assess how likely it is for our techniques to be adopted and integrated into the usual practice of business analysts.

To answer this question, we will compare the practice of the business analysts working in Bupa over a two year period, since the project started. We will investigate the factors influencing the adoption of our techniques in the wider organisation, whether positively or negatively.

5.2 Research Method: Action Research

In this section we offer a brief introduction into action research and discuss the rationale for choosing action research as the evaluation framework for our contributions.

5.2.1 What Is Action Research

Action research is an approach to research which aims at both taking action and creating knowledge or theory about that action as the project unfolds. It is simultaneously concerned with bringing about change in organisations, in developing competencies in organisational members and adding to scientific knowledge [SP85]. In an action research project, theory may be generated and refined and its general application explored through reflection and quantitative and qualitative research methods [KKW10].

What distinguishes action research from ethnographies or longitudinal case studies is the role of the researcher as an agent of change. Action researchers intervene in the setting under consideration for the explicit purpose of improving the situation [Lau99].

As an agent of change, the researcher pursues two complementary goals in an action research project. One goal is to solve a practical problem within an organisation, and the second is to generate new knowledge and understanding. Striking a balance between the two is essential for the success of the action research project. To ensure we give due consideration to both of these aspects, we are following an approach for action research designed for practitioners doing action research in their own organisation [CB14]. The action research cycle we followed is composed of five steps, as shown in Fig 5.1.

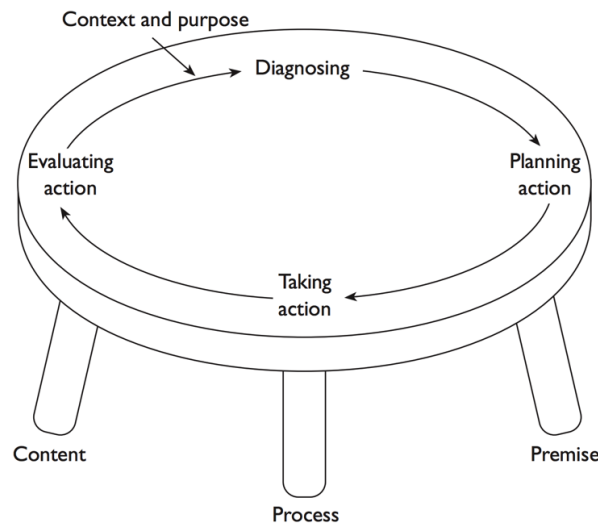


Figure 5.1: Action Research Cycle

Context and purpose: identifies the organisational and commercial drivers. At this step we present what makes this project necessary or desirable; what are the economic, political and social forces driving change; what are the internal organisational factors which should be considered; finally, what is the desired future state.

Diagnosing: specifies the issues which will be addressed through the action research project. It is a statement of scope, establishing a shared understanding among all the participants.

Planning action: defines the desired future state, as well as the required work and necessary changes that need to be implemented in the organisation.

Taking action: is the stage dedicated to implementing the solution agreed, in accordance with the overall plan. This step moves the organisation closer to the desired future stage.

Evaluation: evaluate the effects of the actions taken so far. At this stage, the participants produce insight which is used to adapt the overall plan.

While the above steps are in a similar vein to other approaches for action research, the process presented in Fig 5.1 also emphasises the need to continually reflect as we complete each step. This ongoing reflection will help us address our stated research questions. Specifically, at each step in the action research cycle we reflect on three different aspects:

- the content which is being used or produced in each step
- the process, strategies and procedures followed at each step
- the premise, including the stakeholders attitudes and behaviours, which impact each step

5.2.2 Why Action Research

The method we present in this thesis is meant to support business analysts in their day-to-day work and ultimately lead to better project outcomes. As such, it is crucial to understand if and how the techniques we have introduced in the previous chapters can be used in a real business setting, and whether they

do indeed impact the project outcomes. Moreover, it is important to assess how likely it is for these techniques to be adopted and integrated into the usual practice of business analysts.

Action research is well suited to address all these aspects. First, because action research allows us to evaluate the applicability and merits of our techniques in a real organizational context. Taking advantage of the richer evaluation framework (compared to control experiments for example), we can reflect on the impact of our techniques to the project at hand and to the organisation at large. An action researcher is committed to ensure the outcomes of the research are meaningful for the company and in the day to day practice of its employees. From this point of view, action research is problem-focused, context specific and future oriented. It draws on the practitioners' experience and frames the results into the specific context of their work [Mey00]. Thus, it mitigates common concerns in the industry related to the disconnect between academic theory and day to day practice.

Second, action research usually relies on collaboration and participation: practitioners are working together with researchers, managing the process of change and following the same organisational rules. The researcher is seen as an equal to the practitioners. Consequently, through an action research project run within an organisation we are able to evaluate the existing state of practice for process modelling and requirements specification. By engaging directly with the practitioners, we can introduce the methodology presented in this thesis as a way to support the project outcomes. As the project evolves, we can identify the situations where our techniques are applicable and investigate how they are used in practice.

Action research differs from other research methods in that researchers do not pursue generalizable results. They aim to generate knowledge based on action within one's own situation [KKW10]. Any findings from the research are generalizable only within that situation and within the context of the work and the researcher's beliefs. The dissemination of findings could be applicable to those who are interested and to other practitioners in similar circumstances. It may also be useful for those who would wish to apply the ideas and findings within similar contexts.

5.2.3 Action Research in Healthcare

Participatory action research has been increasingly used in public and community health, as a way to involve the community in identifying research priorities and foster co-education and collaboration [VAE⁺04]. An extensive body of research in the literature addresses participatory action research as a method to include and empower nurse professionals in continuing education and career development [SDJSD00], [CW92]. Likewise, many studies involve health care professionals in the process of identifying barriers to health care delivery and testing intervention approaches to address these barriers, either in a hospital setting [Fen08] or in a community service [KKW10].

5.2.4 Action Research in Software Engineering

There is an increasing number of researchers using action research in the context of software engineering projects [ST09]. There are two main reasons for this.

First, action research is regarded as "the most realistic research setting found, because the setting of the study is the same as the setting in which the results will be applied for a given organisation" [SDJ07].

Second, action research can deal with the social facets of software engineering as it puts more

emphasis on "what practitioners do than what they say they do" [ALMN99].

Action research is being applied to a wide spectrum of SE research domains [ST09]. For example, it was used to define the models, languages and stages of a methodology for the design of secure databases, as part of a project to redesign a Spanish Provincial Governments database [FMP03]. Another participatory action research study focused on developing a methodology for software maintenance. It took place within a large software maintenance company, and it was triggered by the increased volume of work due to the Y2K effect [PPR02]. Action research was utilised to understand how a RE process can be implemented successfully throughout an organisation and what are the factors contributing to its success [KVK⁺04]. Another example is using action research to investigate how creative sessions can be embedded and used during the RESCUE process for requirements engineering [MR05].

5.3 Context and Purpose for Bupa COPD System

In 2008 Bupa Home Healthcare (BHH), in collaboration with two local GPs, has set up a community service for patients with Chronic Obstructive Pulmonary Disease (COPD) living in the Somerset area. COPD is a lung disease defined by persistently poor airflow as a result of breakdown of lung tissue and dysfunction of the small airways. Its symptoms include shortness of breath, cough, and sputum production. Bupa's vision for the community service was to provide an integrated model of care leading to a reduced number of avoidable admissions, improved patient experience, measurable improvements in patients outcomes and value for money. The service was designed to integrate with the wider health and social care environment by steering patients towards pulmonary rehabilitation or oxygen therapy services as appropriate.

The service was designed around a care pathway agreed with NHS stakeholders, shown in Fig 5.2 (taken from the original tender document for the service). Patients were referred into the program by their GP, or they could self refer; they were then assessed by specialist nurses, either in a clinic or in the patients homes; after the assessment, the nurses would create a care plan for each patient, which would include referral to additional services (like pulmonary rehabilitation classes) if necessary; finally the plan was communicated to the patient and her GP and periodic visits would be scheduled, for the nurses to help patients with self management. In July 2013, 2,600 patients were accessing the COPD service.

The NHS contract for the provision of the COPD service in Somerset was set to expire in March 2014. As Bupa intended to put forward a tender for the service renewal, business stakeholders considered the opportunity to make further improvements to the service that would increase the strength of the bid. Three aspects were considered specifically: deliver improved and measurable health outcomes for the patients; deliver financial improvements to the service; demonstrate that this model of care can scale nationally.

Renewing the COPD contract was also part of a broader key priority area for BHH in 2013/2014 - to develop and deliver a Virtual Ward (VW) service to an NHS customer. A virtual ward is a concept within out-of-hospital care defined as "a hospital ward (systems, staffing, daily routines) which is delivered to patients within their own home or community setting". The model benefits patients who are frequently readmitted to hospital and typically have a long stay in hospital. This includes people with COPD,

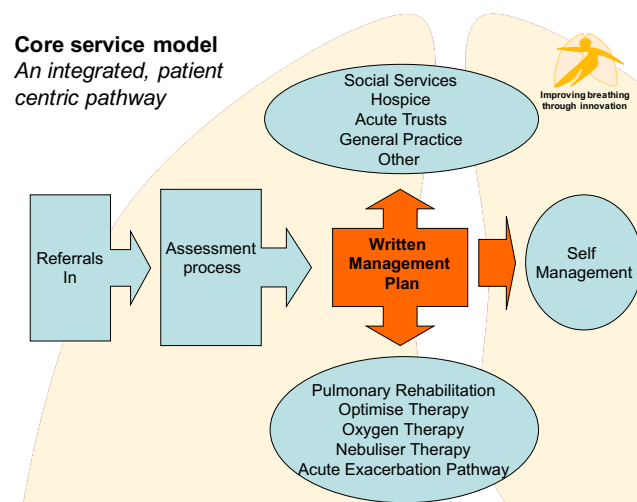


Figure 5.2: Somerset COPD Service - Care Pathway

dementia, Alzheimers, cardio vascular diseases or diabetes. In this regard, Bupa was interested to use the improved COPD service as a test bed for the wider Virtual Ward opportunity. Once proven with COPD, the same model of care could be applied across the other health conditions.

With all this in mind, a project kicked off in late 2013, bringing together internal stakeholders within Bupa, external technology providers and academic links with UCL.

5.4 Diagnosis

In this section we present the initial state of affairs, when the project was set up, that are relevant for our research questions.

First, to address **RQ1** - measuring the success of the Bupa COPD project - we present the high level objectives for the project.

Second, to address **RQ2** - the contribution of our methods to the success of the project - we present the proposed EHR system and discuss the main design challenges.

To address **RQ3** - the impact of our methods on the practice of the business analysts community - we evaluate how business process modelling and analysis had been used in Bupa before our project started.

We conclude the section by arguing how the various issues that had emerged during the diagnosis stage are linked by a common thread: a requirement for more robust process modelling and analysis techniques.

5.4.1 Business and Clinical Objectives

Bupa considered the Virtual Ward model of care to be a promising growth area. The CCGs that were awarding this type of contracts expected the tendering companies to show prior delivery experience, via successful pilots. As such, some of the project stakeholders were looking for an opportunity to pilot a virtual ward setup, to expand Bupa's delivery expertise in this area.

A number of such pilots had already been evaluated in the UK by research institutions such as the

Nuffield Trust [LVW⁺13] or Kings Health Partners [STGK]. Consequently, Bupa required a similar level of academic rigour when developing and evaluating its own model for delivering virtual ward services.

Considering these requirements, the project owners had identified a need to generate measurable evidence and a detailed case study for the new COPD service, in collaboration with University College London, which could be referenced in future Virtual Ward tenders.

In addition to this strategic business direction, some of the stakeholders were concerned with the immediate needs of the COPD service. To understand these needs, we have conducted a series of interviews with the clinical development manager and the business sponsors for the project.

The most important limitation of the service, raised both by the clinical staff and the business owners, was its lack of scalability. Specifically, certain aspects of the service delivery still relied on manual and time consuming steps. Although an IT platform was in use, it was only available inside Bupa clinics. That meant that during home visits, nurses would still have to use paper based questionnaires and take notes by hand. Moreover, the IT system in place at the time was only accessible through a Virtual Private Network, which was impacting the accessibility and responsiveness of the system. The user experience was poor, and the numerous administrative tasks were seen as a burden.

Given the limitations imposed by the existing IT infrastructure, stakeholders were in agreement that a new IT system and improved workflows were required. Their expectations were to reduce the time nurses spent on administrative tasks, deliver costs savings and improve accessibility, ultimately leading to better care and advice to patients.

From a clinical delivery perspective, by reviewing the initial contract with the Somerset Clinical Commissioning Group (CCG), we have identified the following target Key Performance Indicators for the service:

- 90% of patients wait less than 9 weeks for initial assessment (subject to patient choice)
- 60% of patients who started a pulmonary rehabilitation programme have completed it
- more than 70% of patients are satisfied with the service (measured in an annual survey)
- more than 25% of patients have a positive improvement in their St George's score post completion of the pulmonary rehabilitation programme
- minimum of 20% reduction in unscheduled care admissions for known COPD patients who are referred to the programme (compared to the 2006/07 baseline)

Based on a review of the service performance up to 2013 (against the above mentioned KPIs), we have identified three main areas for improvement.

First, the nurses have expressed a need to better manage appointment scheduling, to reduce the risk of a patient waiting more than 9 weeks for the initial assessment. Although the service was currently meeting this particular KPI, an ever present risk raised by the nurses was that patients may "fall through the net".

Second, the business owners wanted evidence to support Bupa's claims that the service has managed to reduce the number of unscheduled admissions into hospitals. As of 2013, this claim was difficult to prove without better integration to the IT systems used by the NHS.

Third, the project stakeholders wanted to improve patient satisfaction with the pulmonary rehabilitation programme, in order to improve completion rates.

5.4.2 Tolven Overview - Open Source EHR

The EHR platform chosen for the project was Tolven [Tol], an open source electronic clinical health record. Tolven was designed with a basic set of functionalities, and was meant to be extended by each organisation deploying the system with project specific add-ons.

Tolven's architecture is illustrated in Figure 5.3. Data is saved in Tolven as a collection of **Documents**. A document may capture raw data, such as a photograph or a scanned document, or it can store structured data, such as an XML document following specific clinical standards like HL7[DAB⁺06].

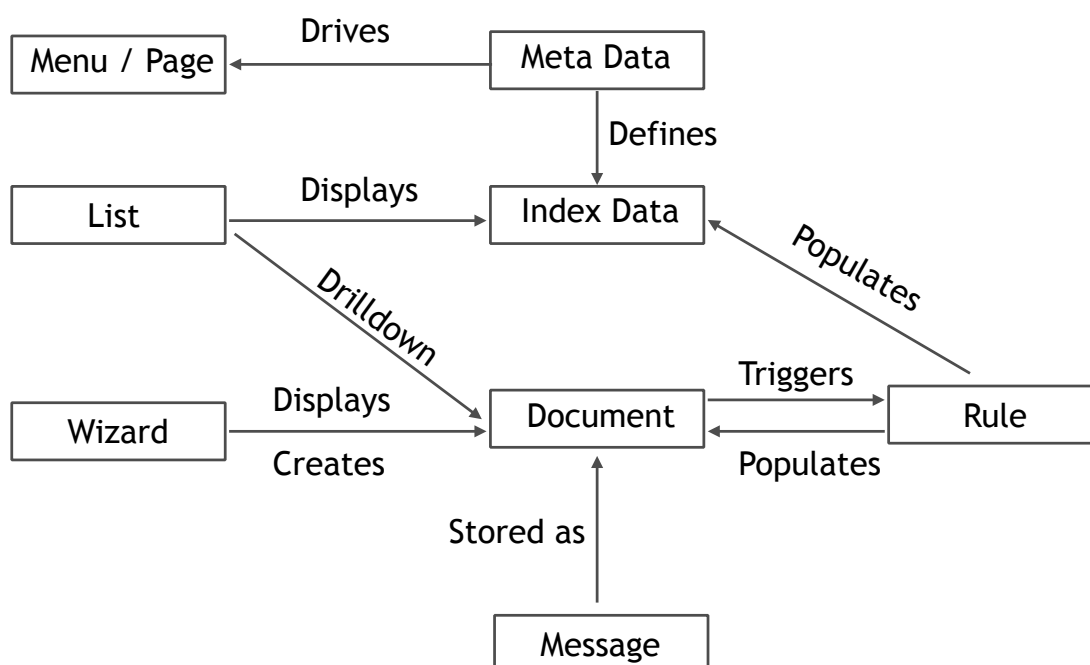


Figure 5.3: Tolven Overview

Documents can be created either by users, through data entry **Wizards**, or automatically from **Messages** received from external systems. When a new document is created a rule engine decides which **Rules** should be triggered. The rules encapsulate the logic for processing the data and triggering an appropriate action. Rules may create additional documents or they may refresh **Index Data**. Index data is extracted and derived from documents. While the documents capture the original data, the index data captures a normalised, columnar view of the information. A **List** is a pre-build view over the index data, prepared for high speed display. Meta data defines the columns required for each list.

Figure 5.4 shows standard Tolven functionalities: clinical assessments, current and recommended medication, medical history etc. The software development efforts focused on creating additional wizards for the data collection needs specific to the COPD service.

The screenshot displays the Tolven Patient Record for Emma Callanan. The interface includes a top navigation bar with tabs for Activity, Admin, Patients, Schedule, and More. Below this is a sub-navigation bar with tabs for Overview, Assessments, Orders, Problems, Allergies, Personal, Medications, Recommended Medications, Discharge Instructions, and Immunizations. The main content area is divided into several sections:

- Assessments:** A table with columns for Date, Assessment Name, and Status. Entries include COPD Assessment (Initial), COPD Referral Form, Nebuliser Assessment, Smoking Assessment (Never smoker), Smoking Assessment (Unknown if ever smoked), and Oxygen Assessment (Initial).
- Discharge Instructions:** A section with a 'New' button.
- Immunizations:** A section with a 'New' button. Entry: 02/11/2014 pneumococcal vaccine, unspecified formulation.
- Problems:** A section with a 'Document' button. Entries: [V] Personal history deep vein thrombosis (situation) - active; [D] Brain scan abnormal (situation) - active.
- Lab Results:** A section with a 'New' button.
- Results:** A section with a 'New' button.
- Medication List:** A section with a 'Review New' button. Entries: 02/11/2014 Abatacept 125mg/1ml solution for injection pre-filled syringes; 02/11/2014 Abacavir 20mg/ml oral solution sugar free.
- Appointments:** A section with a 'New' button.
- Alerts:** A section with a 'New' button. Entries: 02/11/2014 Referral to Breathlessness Mgmt; 02/11/2014 Referral to Pulmonary Rehab.
- Reminders:** A section with a 'New' button.
- Encounters:** A section with a 'New' button. Entry: 02/10/2014 COPD Program Encounter.

Figure 5.4: Tolven Patient Record

5.4.3 Process Modelling In Bupa

There was a general agreement among stakeholders that workflows are a key factor to be considered: workflow improvement was one of the main objectives identified for the project by the business owners. However, in 2013, there was not yet a consistent approach towards process modelling within Bupa.

In business oriented presentations, workflows were represented at a high level of abstraction, as seen in Fig 5.5. Their function was to identify the main actors (patient, nurse, doctor) and capabilities (scheduling, notifications, analytics) of the service and clarify how responsibilities are assigned. Although visually and syntactically they resemble a process model (due to the presence of swim lanes and actors, and the implicit sequence of steps), their semantic and purpose is different. Instead of capturing a detailed, complete and accurate description of a business process, they are instead used as a short-hand notation to describe areas of functionality for the new system and the actors involved in the delivery of each capability.

The business analysts community within Bupa followed a much more structured approach, and they were using process modelling as standard practice in their projects. In Figure 5.6 we show the model for the COPD assessment process, model that had already been created before the project start. This captures the steps taken by the nurse once a patient has been successfully enrolled into the service and he is due for his initial COPD assessment. As a result of this assessment, the patient receives a care plan and he may be further referred to additional services.

When the COPD project was initiated Bupa UK had a basic methodology to support business pro-

High Level Design

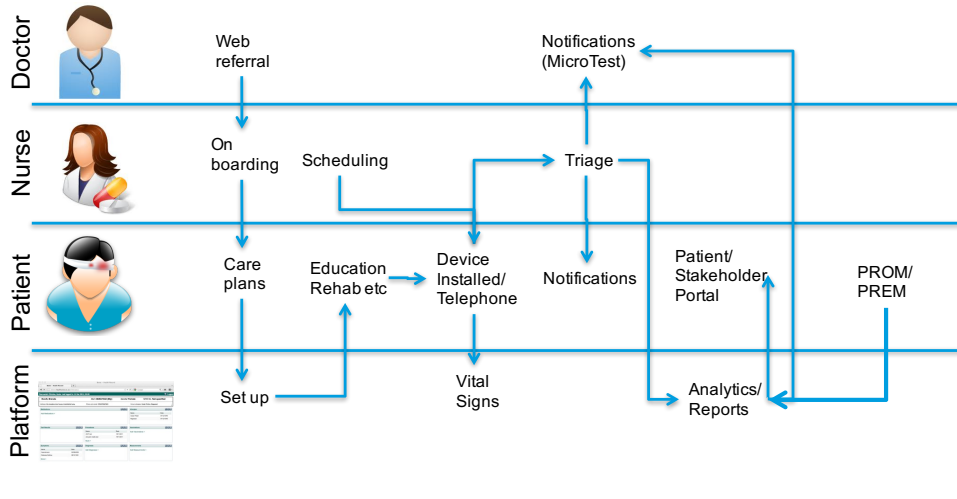


Figure 5.5: COPD proposed workflow

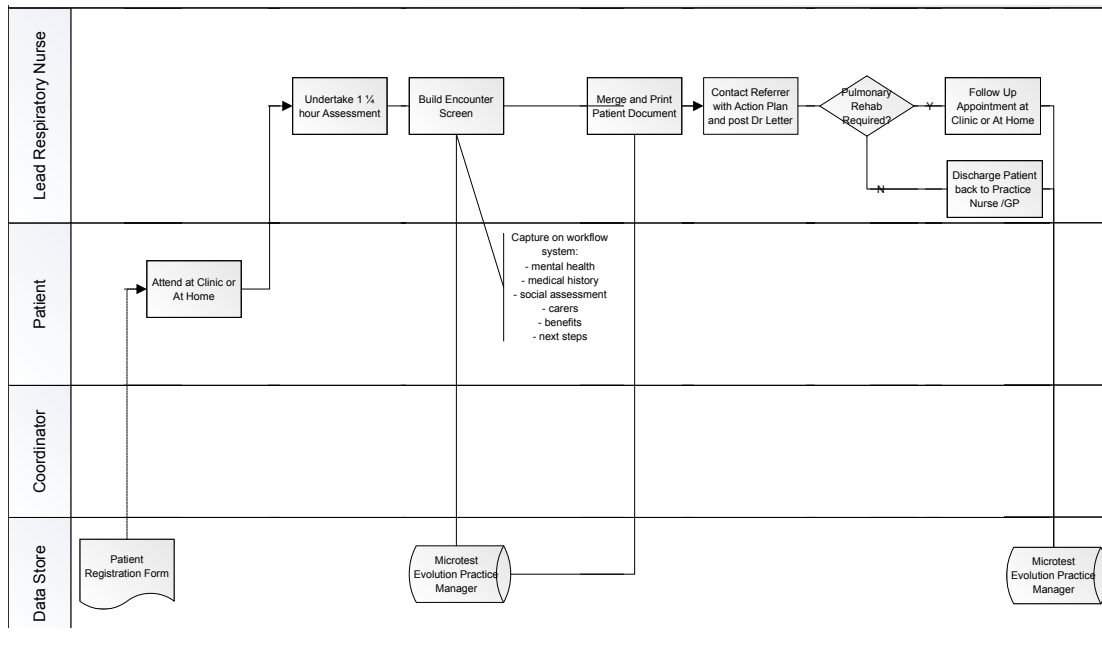


Figure 5.6: COPD Assessment Process

cess modelling. The standard modelling tool was Microsoft Visio and there was a central repository where business analysts could share their models.

From reviewing several such models, it was clear that syntactically the process models were following a common set of conventions: activities were labelled using an 'action - object' pattern, input documents and IT components were identified graphically and comments were used to capture additional details. Each process was modelled at several different levels of abstraction, to allow stakeholders to review them at the desired detail. However, only the 'happy path' was shown - the best case scenario where everything works as expected.

Semantically, there were still many inconsistencies or ambiguities. For example, potentially important information seemed to be missing: the activity labelled 'Attend at Clinic or at Home' does not give any information on how that decision is made or the effects of it. Another example of ambiguity is the activity labelled 'Contact referrer with action plan' - the referrer does not appear in the process model, so it is not clear what her role is in the process. The start and end events for the process are not shown, so it is not clear what triggers this process, or when it can be considered complete.

We also note that the model does not contain information that a business analyst may use to identify areas where the process could be improved, or how the improvements could be measured. The business goals driving this workflow are not explicit.

5.4.4 Conclusion

The requirement to model and analyse workflows was a common thread we uncovered during the diagnosis stage.

Stakeholders expected improved automation, improved patient capacity, better patient outcomes and more precise measurement of results. In light of these goals, business analysts in Bupa required a way to identify areas that could be improved in the COPD service. Conversely, they needed to establish which areas of the existing workflows could not be changed and the reasons behind that.

Bupa required a way to prove how the investment in the new technical infrastructure translated into measurably more efficient workflows or better quality of care.

Although workflow modelling was familiar to the Bupa business analysts, the methodology used for this was inconsistent and the information captured during this stage was not much used later in the project lifecycle.

Our role, as it was defined at the beginning of the project, was to manage the process modelling activities and use the techniques we have presented in this thesis to support the project team in successfully delivering on all of these aspects.

The project team included a Bupa business analyst assigned to the project, responsible for writing user stories, testing the system and training the users, a lead developer responsible for configuring and extending Tolven with functionality required by the user stories, and a project manager. The lead nurse and the administrator of the COPD service were also core members of the team, as they had the domain knowledge and could represent the interests of both nurses and patients.

5.5 Planning Action: Requirements Gathering for the System-to-Be

From the start of the project, the author worked in collaboration with a Bupa business analyst. The author was responsible for reviewing existing documentations and creating process models to capture the workflows in place at the time. He was also responsible for creating models for the future workflows. The business analyst was responsible for writing user stories and presenting these to the development team. The author worked together with the business analyst to build the benefit case for deploying a new IT system and to identify design options and present the advantages and disadvantages of each option to the project team. Unless otherwise specified, the models and analysis presented in this chapter were developed by the author, during the project.

Figure 5.7 presents the overall approach we followed. In the first stage we inferred a goal model for the COPD service - we analysed the nurses' workflows following goal elicitation techniques presented in Chapter 4. We then incorporated in the goal model additional business goals, relevant for Bupa's longer term strategy. In the second stage we used this goal model to propose changes to the nurses' workflows and identify software capabilities required to support these changes.

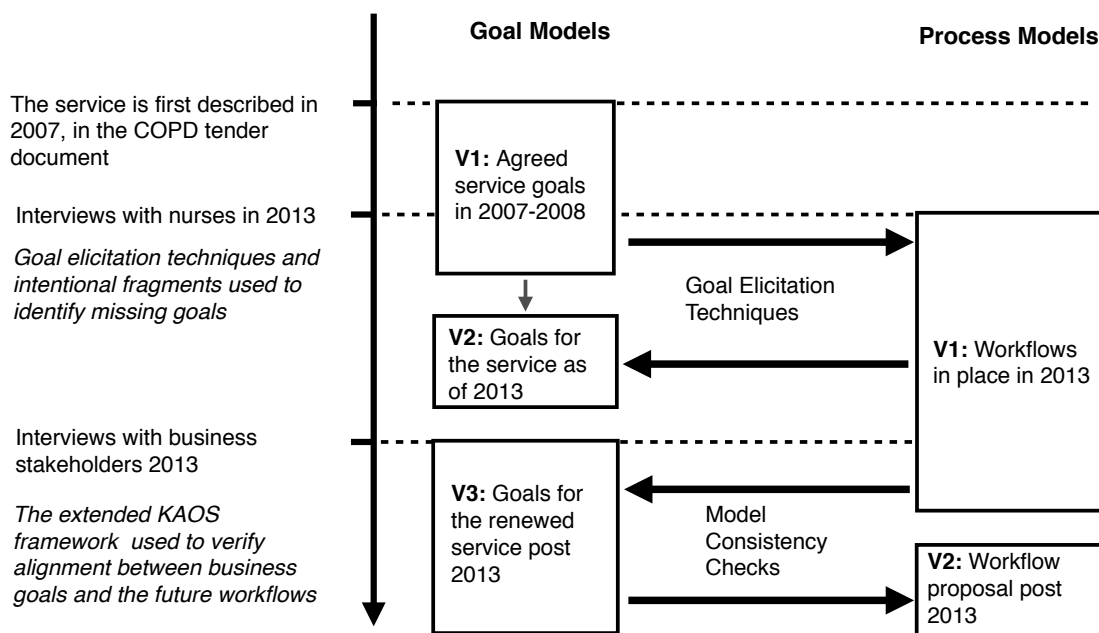


Figure 5.7: Model Development

First, the author created a goal model for the service as it was operating in 2013 using two sources of information - the COPD community service description from 2007 (when the service was first offered to COPD patients) and a process model of the nurses' workflows created before we joined Bupa. We then conducted a series of interviews with the nurses, to verify if the workflow models we had access to were still correct and to explain inconsistencies between the goals defined in 2007 (the first year in which the COPD service was offered to patients) and the workflows documented in 2013. Following

these interviews, the author created new KAOS process models that were validated by the lead nurse and the service administrator as an accurate depiction of the workflows in place at the time. On these process models, the author applied the goal elicitation method presented in Chapter 4. The resulting goal model was then used to identify areas where the nurses' workflow should be changed, the rationale for change and the benefits expected if the changes were implemented. The new workflow was presented to the project team, to the developers and to business stakeholders. The Bupa business analyst referenced the new workflow when writing user stories, to ensure the requirements communicated to the development team are consistent with the envisioned process.

5.5.1 Goal Model for the As-Is System

The first version of a high level goal model for the COPD project - shown in Figure 5.8 - was developed based on the original tender document for the service provision submitted to the Somerset Primary Care Trust in 2007 by Clinovia, a member of the Bupa group. The main clinical outcome pursued through the COPD service was to reduce the number of hospital admissions linked to COPD that could have been prevented by better management of the condition at home.

”As a main objective of the community COPD Service is to reduce avoidable admissions, strive for a minimum of 20% reduction in unscheduled care admissions for known COPD patients who are referred to the programme”

In the definition above (taken from the tender document), we note the presence of measurable performance targets, even if the goal is not formally defined.

This same document described ways for patients to be referred into the program and the type of services they will have access to (e.g. care plans, pulmonary rehabilitation classes, review of their oxygen prescription etc.). However, relations among goals were implicit - in the goal model presented in Figure 5.8 we make these refinement links explicit.

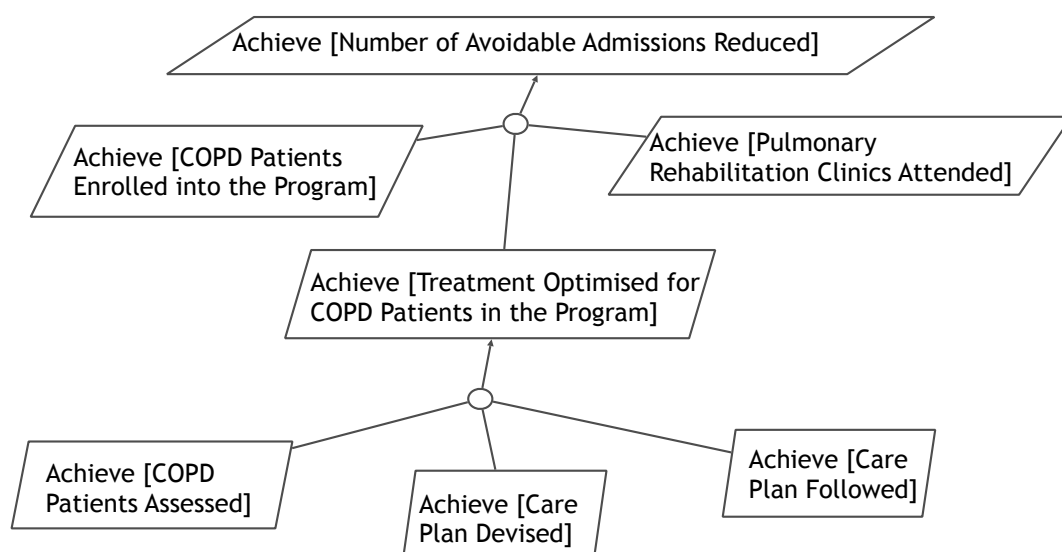


Figure 5.8: COPD Service Goal Model

The high level goals in the model are defined based on their description in the tender document as follows:

Goal	<i>Achieve [Number of Avoidable Admissions Reduced]</i>
Def	When a COPD patient is enrolled into the program, then he should have no unscheduled care admissions
Quality criterion	20% reduction in unscheduled care admissions from the 2006/07 baseline

Goal	<i>Achieve [COPD Patients Enrolled into the Program]</i>
Def	When an appropriate patient referral arrives, then the patient should be enrolled

Goal	<i>Achieve [Treatment Optimised for COPD Patients in the Program]</i>
Def	When a patient has been enrolled into the program, then the patient should follow a care plan within 7 weeks

Goal	<i>Achieve [Pulmonary Rehabilitation Clinics Attended]</i>
Def	When a patient has been enrolled into the program, then the patient should attend a Pulmonary Rehabilitation Course

To further refine the goal *Achieve [COPD Patients Enrolled Into the Program]*, we consider the Enrolment process model in place in 2013, presented in Figure 5.9.

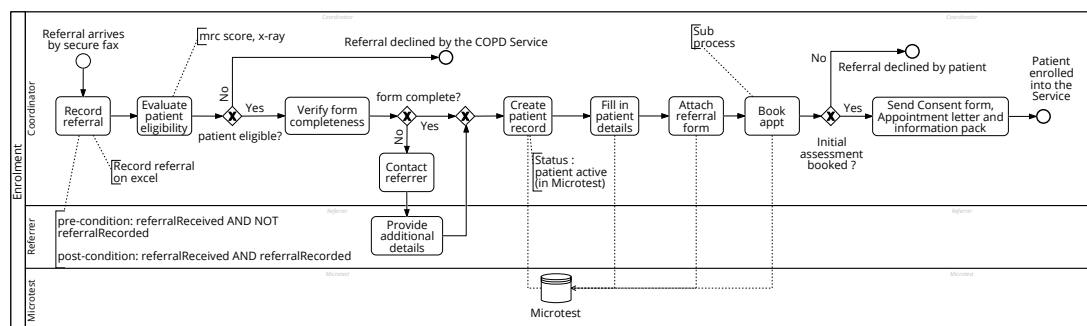


Figure 5.9: COPD Enrolment Process Model - 2013

Patients can be referred by a GP or an acute care unit, via a form sent through secure fax. Each referral is recorded on an Excel spreadsheet. The COPD service coordinator decides whether the patient meets the eligibility criteria. There are two groups of patients who will meet the criteria: those with known COPD, and those patients requiring oxygen assessment. A confirmed diagnosis from a clinician is required before patients can be enrolled into the service. Moreover, patients with coincidental medical conditions such as unstable ischaemic heart disease will not be eligible. If the referred patient is deemed eligible, the coordinator will ensure the referral form is complete and a record for the patient will be created in Microtest - the patient record system in use in 2013. The record is created by manually entering all the information from the faxed referral form. Next, the coordinator will contact the patient

and book an appointment for an initial COPD assessment. If the appointment is booked successfully, the coordinator will then send by post an information pack and a consent form. At the end of this process, the patient is enrolled into the service, and ready for his initial assessment, where a care plan will be agreed.

Next, we apply the goal inference heuristics introduced in Chapter 4 on the process model for patient enrolment. Our aim is to refine the high level goal Achieve [COPD Patients Enrolled into the Program].

H1: Global Goal From Start and End Events

Heuristic H1 applies to the start and end events of the process, presented in Figure 5.10.

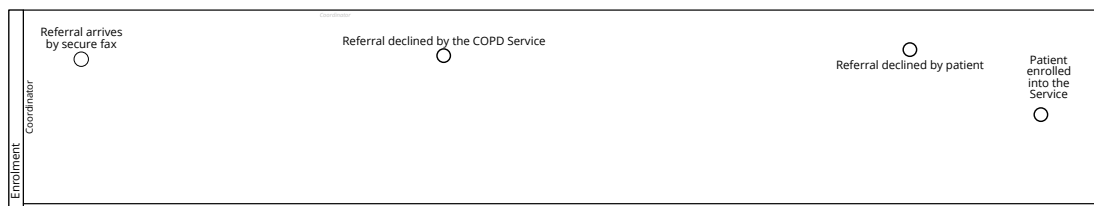


Figure 5.10: Enrolment Process Model - Heuristic H1

Since we already have the name of the high level goal, we will use heuristic H1 to generate a goal definition that is more precise.

Goal	<i>Achieve [COPD Patient Enrolled into the Program]</i>
Def	When a referral arrives, then the patient should be enrolled into the service or the referral should be declined by the patient or the referral should be declined by the service.
Observations	This definition immediately proves meaningful, as it draws attention on the fact that under certain circumstances a patient referred into the service may not be enrolled.

To understand the specific circumstances which lead to the different end states for the enrolment process, we apply Heuristic H9 to start refining the goal.

H9: Refine the Global Goals

To refine the goal Achieve [COPD Patient Enrolled into the Program] we use the conditions on the two exclusive gateways that generate distinct end events: Patient Eligible? and Initial Assessment Booked?. These elements are shown in Figure 5.11.

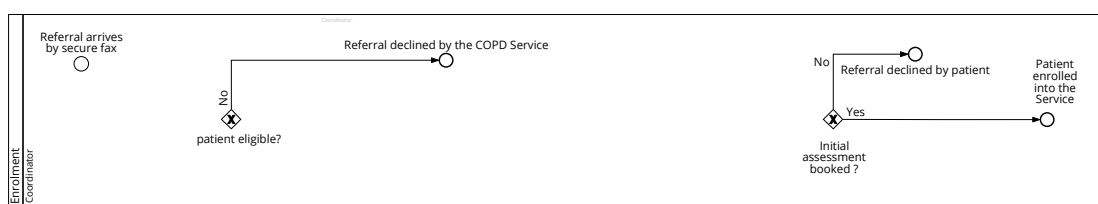


Figure 5.11: COPD Enrolment Process Model - Heuristic H9

Applying heuristic H9 leads to the goal refinement shown in Figure 5.12. Note that we apply the heuristic two times, as two different exclusive gateways generate end events.

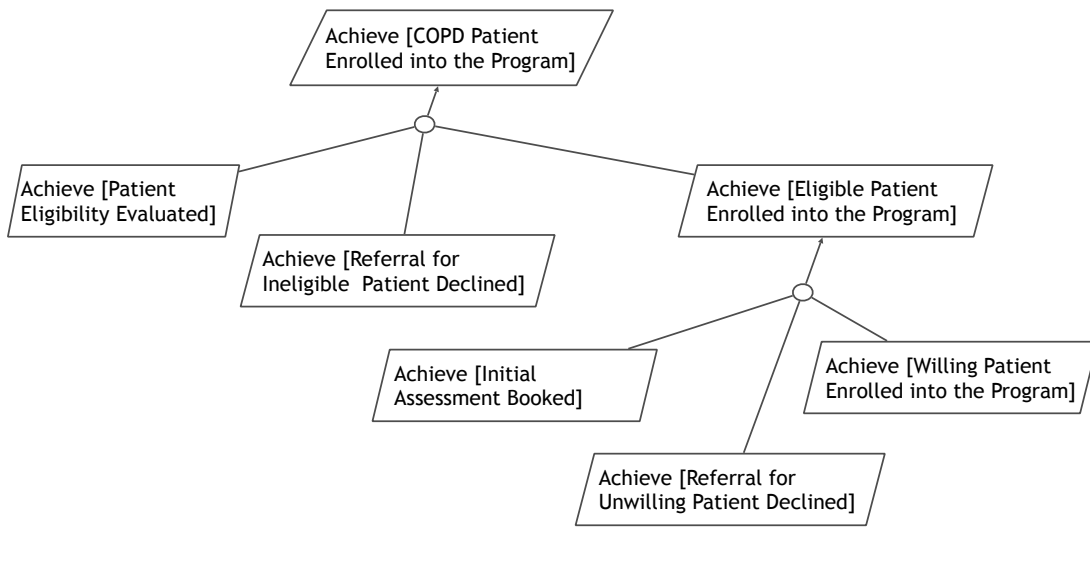


Figure 5.12: Patient Enrolment - First Goal Model Refinement

Taking into account the start events, end events and exclusive gateways in the process model, we can immediately propose a number of intentional fragments that contribute to the goals identified so far. Note that these intentional fragments will be further evaluated and refined. We give below the goal definitions, as well as the intentional fragments relevant for each goal, identified from the process structure. For each goal definition, we discuss whether it brings to light any new information, compared with the service description document.

Goal	<i>Achieve [Patient Eligibility Evaluated]</i>
Def	When a referral arrives, then the patient should be declared eligible or she should be declared ineligible
Observations	In 2013, this goal was under the responsibility of the coordinator. The goal was documented in the service description document, and the project team had access to the eligibility criteria.
Intentional fragment	

Goal	<i>Achieve [Referral for Ineligible Patient Declined]</i>
Def	When a patient is ineligible for the service, then the patient referral should be declined
Observations	This goal was also mentioned in the service description document. Furthermore, there was a requirement to notify the referrer whenever a referral was declined. In this instance, the process model was incomplete - it did not include an activity for notifying the referrer.
Intentional fragment	<pre> graph TD Start(()) --> D1{patient eligible?} D1 -- No --> End((Referral declined by the COPD Service)) </pre>

Goal	<i>Achieve [Eligible Patient Enrolled into the Program]</i>
Def	When a patient is eligible for the service, then the patient should be enrolled
Observations	We refine this goal by applying heuristic H9 for a second time.
Intentional fragment	<pre> graph TD Start(()) --> D1{patient eligible?} D1 -- Yes --> A1[Verify form completeness] A1 --> D2{form complete?} D2 -- No --> A2[Contact referrer] A2 --> D2 D2 -- Yes --> A3[Create patient record] A3 --> A4[Fill in patient details] A4 --> A5[Attach referral form] A5 --> A6[Book appt] A6 --> D3{Initial assessment booked?} D3 -- No --> End1((Referral declined by patient)) D3 -- Yes --> A7[Send Consent form, Appointment letter and information pack] A7 --> End2((Patient enrolled into the Service)) </pre>

Goal	<i>Achieve [Initial Assessment Booked]</i>
Def	When a patient is eligible for the service, then the initial assessment should be booked or refused
Observations	This goal captures one of the core clinical commitments of the service - the fact that any eligible patient referred to the service will be offered an initial assessment.
Intentional fragment	<pre> graph TD Start(()) --> D1{patient eligible?} D1 -- Yes --> A1[Verify form completeness] A1 --> D2{form complete?} D2 -- No --> A2[Contact referrer] A2 --> D2 D2 -- Yes --> A3[Create patient record] A3 --> A4[Fill in patient details] A4 --> A5[Attach referral form] A5 --> A6[Book appt] A6 --> D3{Initial assessment booked?} </pre>

Goal	<i>Achieve [Referral for Unwilling Patient Declined]</i>
Def	When the initial assessment can not be booked, then the referral should be declined
Observations	This goal captures the expectation that when a patient can not be seen for an initial assessment by a nurse, his referral will be declined. This goal definition was a first indication of how important the appointment booking functionality was: failure to agree on a booking for the initial assessment equates with the referral being declined.
Intentional fragment	<pre> graph TD Start(()) --> D{Initial assessment booked?} D -- No --> G((Referral declined by patient)) D -- Yes --> X{X} </pre>

Goal	<i>Achieve [Willing Patient Enrolled into the Program]</i>
Def	When an initial assessment is booked for an eligible patient, then she should be enrolled
Observations	This intentional fragment contains only one activity. This was identified as an area for further investigation, to ensure a complete goal refinement.
Intentional fragment	<pre> graph TD Start(()) --> D{Initial assessment booked?} D -- Yes --> A[Send Consent form, Appointment letter and information pack] A --> G((Patient enrolled into the Service)) D -- No --> X{X} </pre>

Once we have created a goal model based on the start events, end events and the gateways present in the process model, the next step is to analyse the activities. As such, we will start specifying the intentional fragments in our process model.

H6: Intentional Fragments from the Process Structure

To identify intentional fragments, we start by analysing the process structure.

One heuristic we can apply on the enrolment process model retains the activities between the divergent gateway Form complete? and the corresponding convergent gateway. The resulting goal definition is representative for an information goal, a functional goal concerned with keeping agents informed about important system states.

Goal	<i>Achieve [Referral Information Collected]</i>
Def	When the referral form is not completely filled in, then the referrer should provide the required information.
Observations	Based on the information in the process model, no IT system was contributing to this goal. The goal was identified as an area for further exploration.
Intentional fragment	<pre> graph TD G1{form complete?} -- No --> C1[Contact referrer] C1 --> G2{form complete?} G2 -- Yes --> G1 C2[Provide additional details] --> G2 </pre>

Next, we consider the chain of activities before the gateway that tests whether an initial assessment has been booked successfully, shown in Figure 5.13.

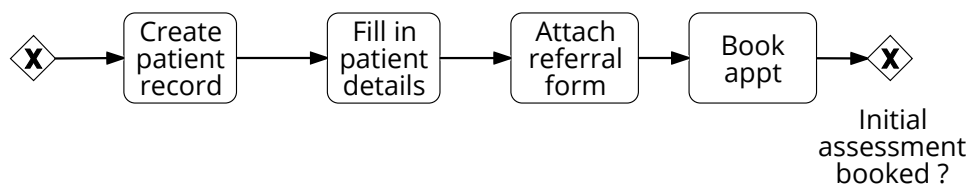


Figure 5.13: COPD Enrolment Process Model - Heuristic H6

In this case there is no single intentional fragment which captures a common goal for these activities. Instead, we can identify a number of different goals. We give their definition below.

Goal	<i>Achieve [Patient Record created on EHR Platform]</i>
Def	An electronic healthcare record should be created for each eligible patient referred into the service
Observations	One of the original commitments related to the patient care quality was that the care plans for patients enrolled into the COPD service will be managed electronically.
Intentional fragment	<pre> graph LR A1[Create patient record] --> A2[Fill in patient details] </pre>

Goal	<i>Achieve [Appointment Booked]</i>
Def	When a patient needs an appointment, then an appointment should be booked
Observations	This goal is related to the quality of care requirements for the service. There are specific clinical guidelines that further specify what are the medical circumstances when a patient needs an appointment.
Intentional fragment	

H8: Intentional Fragments for Non Functional Goals

To identify additional intentional fragments, we also verify if there are activities in the process that are required for non-functional goals. There is one such requirement, for regulatory compliance:

Goal	<i>Achieve [Original Referral Document Archived]</i>
Def	When a referral is received, then the referral document should be archived.
Observations	This goal was also flagged for further analysis, since at the time it was assigned to the human administrator.
Intentional fragment	

H10: Relation Between Intentional Fragments

We use heuristic H10 to establish refinement relations between the intentional fragments identified so far. This heuristic is applicable because we have a number of intentional fragments whose activities are also part of a larger intentional fragment - for the goal *Achieve [Initial Assessment Booked]*. The candidate goals for the refinement are shown in Figure 5.14. Heuristic H10 proposes that all four sub-goals refine *Achieve [Initial Assessment Booked]*. We also note that one of the intentional fragments - for the goal *Achieve [Referral Information Collected]* has been expanded to include an additional activity. This is an example of business analysts improving on the results generated by mechanically applying a heuristic.

At this stage, the domain knowledge of business analysts is required to validate the proposed refinement links. The decisions taken at this stage will impact the design of the future system. Based on stakeholder feedback, we clarified that only two of the goals should be part of the refinement: *Achieve [Patient Record Created]* and *Achieve [Appointment Booked]*.

The revised goal model is shown in Figure 5.15, with the new goals in dotted boxes. We also illustrate the other two possible outcomes when validating the proposed refinement links. The goal *Achieve [Referral Information Collected]* contributes to a different goal (establishing patient eligibility), while the goal *Achieve [Referral Document Archived]* remains a high level goal, as it is one of the legal requirements for the service.

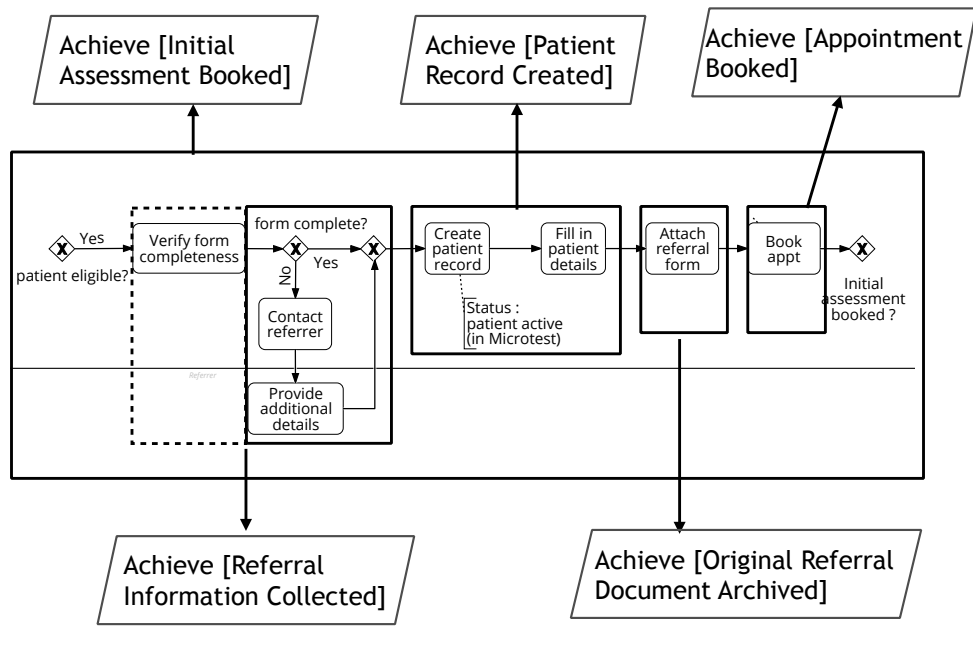


Figure 5.14: COPD Enrolment Process Model - Heuristic H10

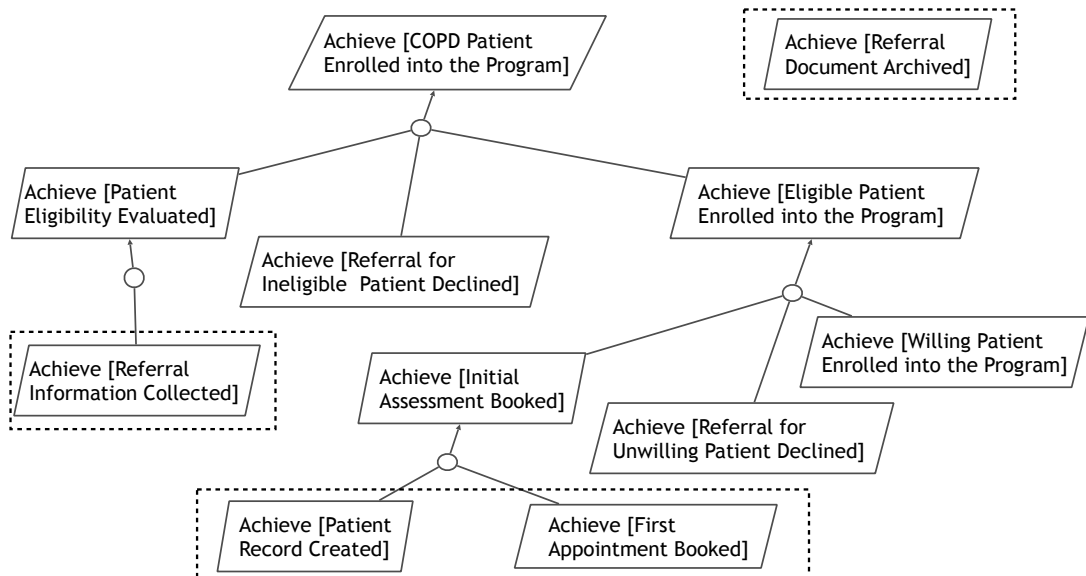


Figure 5.15: Patient Enrolment - Second Goal Model Refinement

Next, we analyse some of the activities annotations, in order to try and identify additional goals.

H7: Intentional Fragments from Activities Annotations

We use as a starting point one of the intentional fragments already identified - for the goal *Achieve [Patient Eligibility Evaluated]*. We have already established the goal *Achieve [Referral Information Collected]* contributes to this goal, as shown in Figure 5.16.

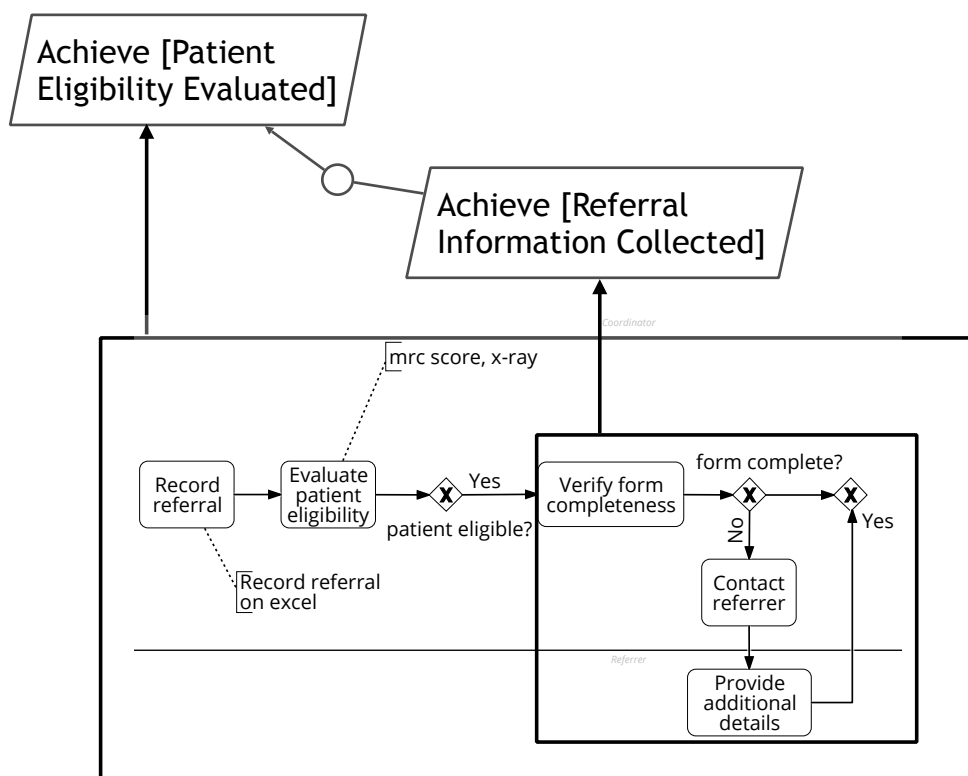


Figure 5.16: COPD Enrolment Process - Heuristic H7

The pre and post-conditions of the activities relevant for this goal have been discovered through interviews with the administrative staff, and are given below.

Activity	<i>RecordReferral</i>
Domain pre-condition	$\text{referralReceived} \wedge \neg \text{referralRecorded}$
Required post-condition for goal	referralRecorded
<i>Achieve[Service Performance Measured]</i>	

Activity	<i>EvaluatePatientEligibility</i>
Required pre-condition	$\text{referralInformationCollected} \wedge \neg \text{eligibilityDecided}$
Required post-condition for goal	$\text{eligibilityDecided}$
<i>Achieve[Patient Values Compared to Eligibility Criteria]</i>	

Analysing the activities pre and post-conditions, we can further refine the goal model. First, we discover that *RecordReferral* contributes to a new goal - *Achieve [Service Performance Measured]*.

This has been used to elicit the duration of the process activities, and to identify manual steps in the process that could be automated. For example, entering the information from the paper based referral form into the Microtest system for patient records was reported by nurses to take between five and ten minutes. There was general consensus this activity was unnecessary long because the slow response time of the current system. Another time consuming step we identified was the activity Record Referral. The administrator had to spend five minutes for each referral received by the service to record its details into a spreadsheet used for business reporting.

The second source of information has been a document detailing 47 main functions of the Tolven EHR system. The textual description of one such functionality is shown in Figure 5.20 - for creating new patient records. These functions were used to inform the design of the new system, by suggesting alternative goal refinements to the goals we have identified at this stage. For example, Tolven's ability to create new patient records from inbound messages meant we could aim to reduce the manual steps involved in creating a new patient record based on the information contained in the referral form.

11. Create and maintain patient demographics (set complies with MU)

- a. Tolven's "create new patient" function allows users to "register" patients and to add required attributes (including patient home address which will serve as encounter location for BHH)
- b. Tolven can also create lists of patients from inbound messages (normally HL7) from other systems
- c. This will need to be extended to cover locales outside the US; we have already showed the beginning of such extensions in the Bupa CPOE PoC

Figure 5.20: Tolven Functionality

We discuss next a number of changes we have introduced in the goal model.

First, we consider the situation of consent - capturing user consent was a clear requirement, but although sending the consent form was part of the patient enrolment process, the COPD service would only receive it back, signed by the patient, during their initial assessment. In Figure 5.21 we show an alternative goal refinement. The activity of sending the consent form to a patient was reported on the process model as both a time consuming task and one of the more expensive ones (because of printing and postage time and cost). We replace this with electronic capture of patient consent. Having the goal model as a reference allows us to confirm that proof of consent is not required for booking the initial assessment, so electronic consent captured at the beginning of the face to face assessment is a valid design option for the system.

Second, we consider the goal of verifying the eligibility of each patient referral. In Figure 5.22 we show the alternative goal refinement inspired by Tolven's functionality: if referrers were given the choice to create Tolven records for the patients they wanted to refer, Tolven could then automatically check whether the referral information is present and if the clinical measurements meet the eligibility criteria. The COPD service administrator would not have to spend time collecting and checking the information herself, but she would still be responsible to decline or accept the referral.

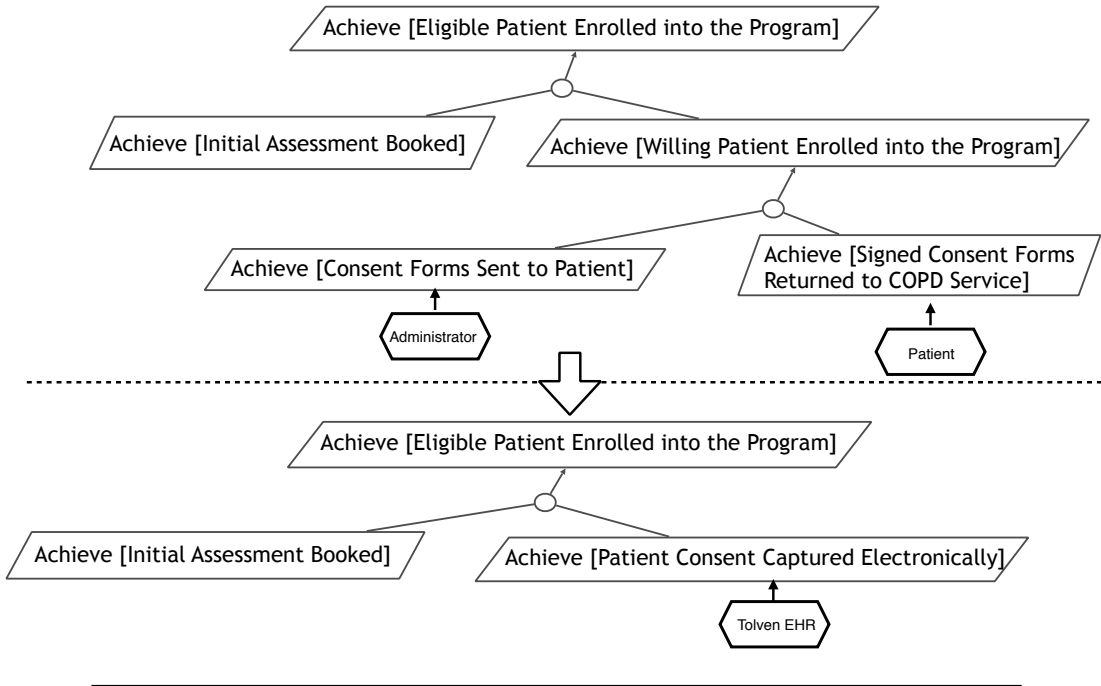


Figure 5.21: Consent in the System-To-Be

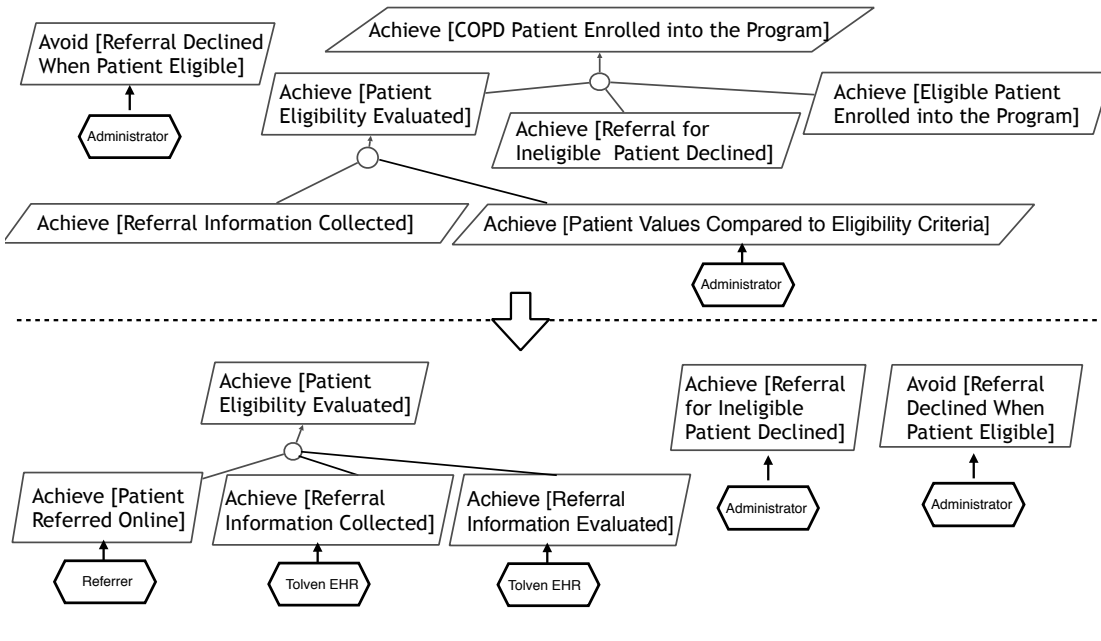


Figure 5.22: Eligibility Verification in the System-To-Be

Third, we consider the need to measure the performance of the system. From the process model for patient enrolment, we have uncovered a partial refinement of the goal *Achieve [Service Performance Measured]*. This goal was validated by business stakeholders as one of the main areas of interest. Specifically, the COPD service was expected to improve its reporting capabilities and be able to demonstrate the health impact it was delivering for the community of patients.

Figure 5.23 present an alternative goal refinement, which requires Tolven to maintain a complete log of all the patient interactions and measurements.

The performance reports for the COPD service had to be accessible to a wide range of stakeholders, both internal and external. For example, members in Bupa's leadership team wanted to know if the service meets the key performance indicators and the NHS commissioning group wanted quarterly reports showing whether the COPD service meets the contractual targets. In these cases, the beneficiaries of the reports did not require direct access to Tolven for any other tasks. A goal refinement where the reports are generated by a separate system better meets non-functional goals: it is more secure, as access to Tolven is restricted to nurses and clinical administrators and it is more cost effective. Tolven had limited reporting capabilities, and the cost of developing the reports in Tolven were greater compared to using existing business intelligence tools. Business owners for the COPD project, presented with this argument, validated our design choice.

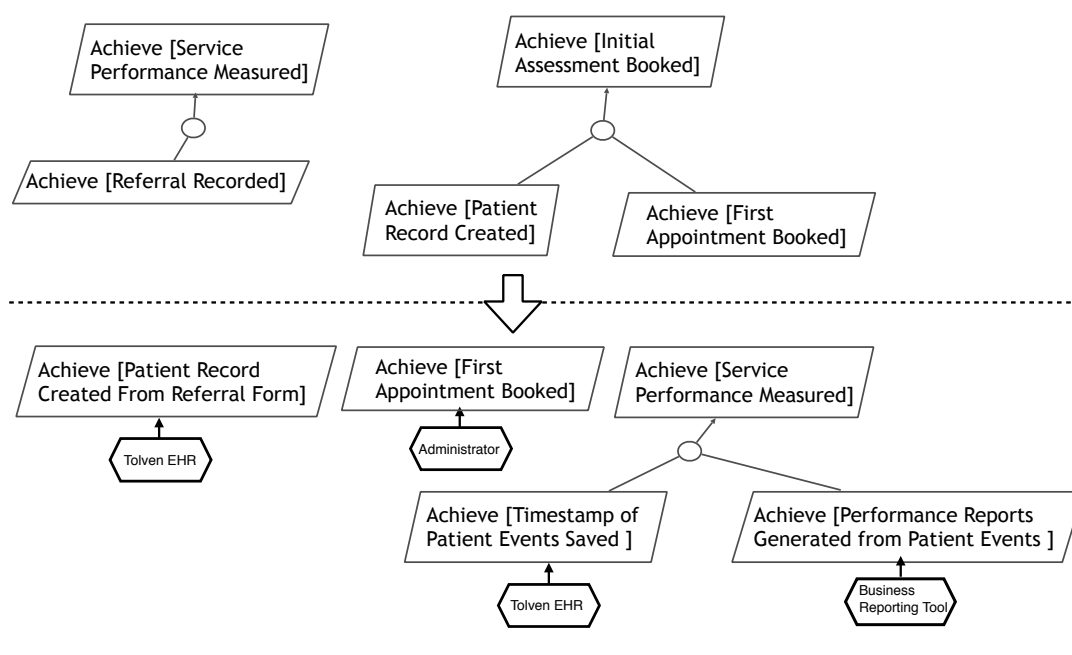


Figure 5.23: Performance Reporting in the System-to-Be

5.5.3 Process Model for the System-to-Be

Based on the new refinements in the goal model, we designed a new process model for enrolment, by following the consistency rules introduced in Section 5.2

Specifically, we ensure that for each goal in the goal model there is an intentional fragment composed of one or more activities to ensure that goal is satisfied. We also ensure that each activity in the

process model is carried out by the agent responsible for the goal.

The new enrolment process is shown in Figure 5.24. This model differs in several ways from the workflows in place at the time the project started.

During the activity Record Referral in the original process the COPD Administrator had to record details about each referral received by the COPD service on an Excel spreadsheet. This was necessary in order to prove the service processes all referrals received in a timely manner, in accordance with the KPI targets agreed with the NHS trust. This activity was replaced with Save time stamp of referral, which was performed by the Tolven EHR system. This activity was automated, as referrals were received online - thus, we could be explicit about the time saved by the administrator, following the new workflow.

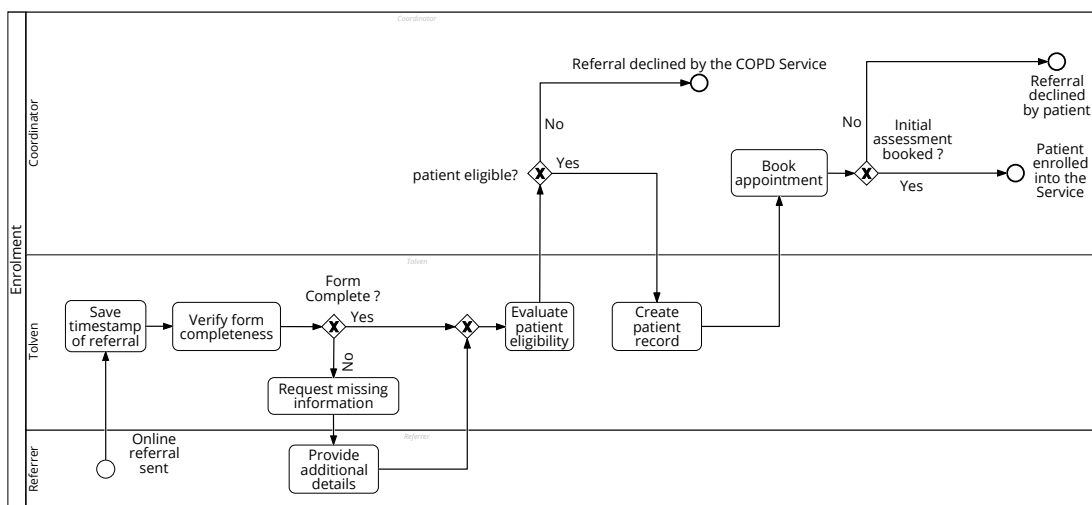


Figure 5.24: New COPD Enrolment Process

The new process ensures the satisfaction of goal Achieve [Eligibility assessed]. It prescribes that all the referral information should be collected before the decision is made. While the responsibility of the decision still lays with the administrator, the Tolven EHR system will assist by first ensuring all data is collected from the referrer and second, by pointing situations where the clinical measures of the referred patient do not meet the criteria for eligibility.

The new process also saves time by removing the activity Send Consent Form. Cost is reduced as well, as there is no need to print and send the consent paper by post. Consent will be taken electronically in the first face to face meeting between patient and nurse.

By outlining these changes at the process level, it is possible to quantify the benefits expected from the new workflow. The specific targets are given in Figure 5.25. The comparison between the process in place at the time the project started and the new process for patient enrolment has been the basis for setting these targets.

#	Benefit Description	Measure	Baseline measurement (Initiation)	Target
4	Reduced cost of printing and postage	Mailings per month	600 (i.e. 300 patients x2 mailings – copy to GP)	150 (i.e. 50% patients mailings)
5	Reduced nursing admin overhead	Admin days per week per nurse	1 day	0.5 day
6	Reduced monthly effort in reporting	Reporting days per month	5 days?	2 days

Figure 5.25: Benefit Case for the Tolven Based EHR

5.6 Taking Action: Extending Tolven for Bupa COPD Service

The final process models for the system-to-be were then used by the business analyst assigned to the project to write user stories, and then map the user stories to a requirements backlog. The requirements cover an extension to Tolven's data model, changes to the graphical interface and development of additional functionality.

The first area of functionality the new system has addressed was improving the way patient records are created from referral forms. As mentioned in Section 5.4.2, the Tolven back-end organises information as collections of documents. The format of the documents is based on the clinical standard HL7. That ensures a Tolven-based EHR can receive and send data to other clinical systems, for example those used in GP practices. To manage patient referrals, we have created a new document type (equivalent to a Referral Form). This allowed us to specify the required clinical measures for a referral form to be valid. To create a Referral document, the new COPD system offers a number of options: if the referrer is a GP, the software used in their practice may directly integrate with the Tolven back-end via a messaging mechanism. Otherwise, referrals will be created via a web front-end or a mobile app (both have been created specifically for the COPD service). Referrers would get immediate feedback if a required value has not been provided while trying to refer a patient. This functionality directly addresses one of the problems in the initial workflow - the extra activities required for the COPD administrator to deal with missing information in a referral form. Once a referral form has been created, we use the Tolven rule engine to notify the COPD administrator that she needs to establish patient eligibility. For eligible patients, the referral information is automatically used to create and populate the initial patient record. This conforms with the prescribed workflow for the system-to-be: nurses are not required to do manual input in order to create a new patient record.

The second area of functionality that has been shaped by the process and goal models centers around removing paper based communication, where possible. An example of this is replacing paper based consent with electronic consent, provided by each patient at the time of their first appointment with a

nurse. To deliver this functionality, we have developed a mobile app that is capable of both retrieving information from the Tolven database, as well as updating a patient record with new information. To obtain consent using the mobile app, the patient will sign a PDF declaration, which will be attached to her Tolven based EHR. The Tolven front end (either web based or mobile) is also used during the COPD assessments. To reduce reliance on paper forms, the clinical measurements are captured electronically and added to the patient record.

The third area of functionality covers the COPD service reporting needs. Tolven automatically records every operation performed on a document (e.g. creation, update). This built-in functionality means the data captured by the system can be used to generate reports. Nurses do not have to use Excel files to manually record performance indicators for the COPD service, such as patients' waiting time between referral and the first assessment. To deliver on this functionality, we have created an online dashboard, which queries the Tolven database in real time. The dashboard allows monthly reports to be downloaded and shared with the wider business.

5.7 Evaluating Action: Impact of the new COPD System

5.7.1 RQ1: How Successful was the EHR Deployment

The Tolven based EHR system went live in January 2014. It was used to manage the care of the 2,600 COPD patients registered into the program at the time. To evaluate its success, we consider a number of factors, that cover both the clinical and the business perspective.

From a clinical perspective, the new EHR system has gained the approval of the clinical directors responsible for the service, as a safe and more efficient way to manage patient care. By gains in efficiency we mean reducing the time spent performing administrative tasks - this was calculated based on reports from the service administrator and the nurses themselves. First, after deploying the new EHR, the nurses did not have to do manual data entry for the data collected during assessments conducted at patients' home. This has resulted in 4 hours saved each week, for each of the 6 nurses. Second, the service administrator was able to complete the monthly reports in 2 days, as opposed to 5 days. The new EHR also improved monitoring and recording capabilities for patients' clinical measures, as the data was captured electronically and encoded using a standard clinical terminology (SNOMED). The system was also capable of following a patient's status, from referral, to first assessment and then through regular annual assessments - the system was configured to notify the COPD Service administrator when a patient was due for an assessment.

Besides the formal approval for the system to go live, the medical directors also offered positive feedback for the system's capacity to inform COPD research through the detailed patient data being collected. An internal report presenting changes in the relevant clinical measures for the patients enrolled in the service has been produced and shared with internal stakeholders. The NHS Clinical Commissioning Group in Somerset responsible for the COPD service has also reviewed the new system and confirmed the service may transition to the Tolven based EHR. The system was in actual use from January 2014.

The system was also used to support other Bupa initiatives. For example, it was considered for

potential deployment in Bupa's care homes, as there was a business need to update the IT systems in use at the time. A comparison between the COPD workflows and the workflows followed within the care homes identified an important area of functionality which was not covered by the EHR: medication management. A separate project was initiated to further investigate this functionality.

From an operational perspective, the service was enhanced by ensuring the Tolven front-end is available as a mobile app. This ensured off-line access for the nurses to the electronic health records of the patients. The option to refer patients online was considered an important driver for efficiency, as creating the initial patient record was quicker. Moving from paper based communication to electronic documents had also a direct impact on the day to day cost of running the COPD service.

The Bupa business sponsors for the COPD project praised the final outcome - a new EHR and changed nurse workflows - because the operational benefits (time, cost, risk reduction) of deploying the new EHR system were clearly quantified, based on the data collected in Tolven. This provided a sound basis for both internal and external discussions. Externally, the project was selected as a finalist for the Health Service Journal award, a recognition of excellence among care providers in the UK. The project was also referenced in Bupa's entry for the European Commission's Green Paper on mobile health. Among the benefits cited for developing a mobile front-end to the Tolven EHR, we mention: faster assessments (as data is captured electronically), clinically accurate and shareable medical record, automated consent, enhanced metrics available at the point of care. Moreover, the summary report of the consultation mentions that "Several respondents drew attention to the necessity for developers to draw well-designed mHealth workflows that address users' needs".

To support adoption, several training sessions and on-going support were provided. The migration period from the old to the new system was short (two weeks in which both systems were used in parallel), as all the clinical staff were willing to start using the new EHR system. A formal sign-off from the nurses was captured in a meeting with the business sponsors.

Once the service provision relied solely on the Tolven EHR, there was no significant deviation from the prescribed workflow, with respect to the way nurses worked.

The lead nurse was responsible to collect feedback from the nurses, through weekly meetings, and observe how the system is used in practice. During the first six months after the system was live, there was no report of nurses following a different workflow.

5.7.2 RQ2: Contributions of Our Techniques

The value of using a combination of process and goal modelling to inform the design of the system has become apparent throughout the project.

First, creating the benefit case for the project (and evaluating whether it was eventually met) relied heavily on our analysis of the before and after workflows. We have documented the workflows in place at the project start and asked the nurses to evaluate the time and cost required to perform each activity. This step has proven a valuable source of new requirements, as we could identify promising areas for improvement. Even in the enrolment process, we had presented three areas where the original workflows were simply a reflection of the limited IT capabilities and not a true reflection of the business goals:

processing referral forms, capturing consent and KPI reporting. Goal modelling has been used to explore alternative designs, and then reflect these decisions both in the functionalities of the Tolven EHR and the new workflows put in place.

Being able to share with the business sponsors the reason for each change in the workflows and the specific efficiency gains to be expected from these changes was essential to ensure continued support for the project. Ultimately, we have created a compelling benefit case. We were able to demonstrate that the new Tolven EHR and associated workflows have delivered value to both patients and Bupa - this was recognised both internally and externally. The specific category our COPD project was shortlisted for in the HSJ Awards was *Improving Efficiency Through Technology*.

The second area where our use of the extended KAOS framework proved beneficial was to ensure that the nurses carry out their work and use the new system as it was intended. Specifically, we wanted to maximise adoption rate and minimise the number of reported workarounds. The system was keenly adopted by nurses, once it was made available. Discussing both the current and the proposed workflows with the nurses proved a good way to ensure they are familiar and approve of the changes that will be introduced to the way they work, once the Tolven EHR is deployed.

Third, analysing the workflows resulted in a more complete specification for the software to be. Important areas of functionality were identified early on in the analysis stage, and delivery could be planned accordingly. One such case was Tolven's capabilities to handle appointment booking. While Tolven had a built-in module for appointments, only by looking at the details of the booking workflow it became apparent there is additional software development required to support the COPD specific needs.

5.7.3 RQ3: Project Impact in the BA community

Given that many of the Bupa business analysts were contractors, we evaluate the impact of our techniques on two different levels.

First, we consider how the Bupa best practices and guidelines have changed during and after the project finished. On this aspect, BPMN has become the recommended solution for business analysts. This aligns with our proposal, since well formed BPMN models directly map to the KAOS process models.

Not only is BPMN more commonly used, but effort has been put into building the skills of the practitioners, by creating training materials and templates, as shown in Figure 5.26 and Figure 5.27. The modelling guidelines promoted across Bupa also align with the structural constraints and the labelling conventions we have proposed. This is a marked departure from the practice before our project started. Initially, there was no consistent support for process modelling. Most process diagrams were saved as Visio files, and their relation to software requirement was not explicit. The modelling conventions were not standardised across teams.

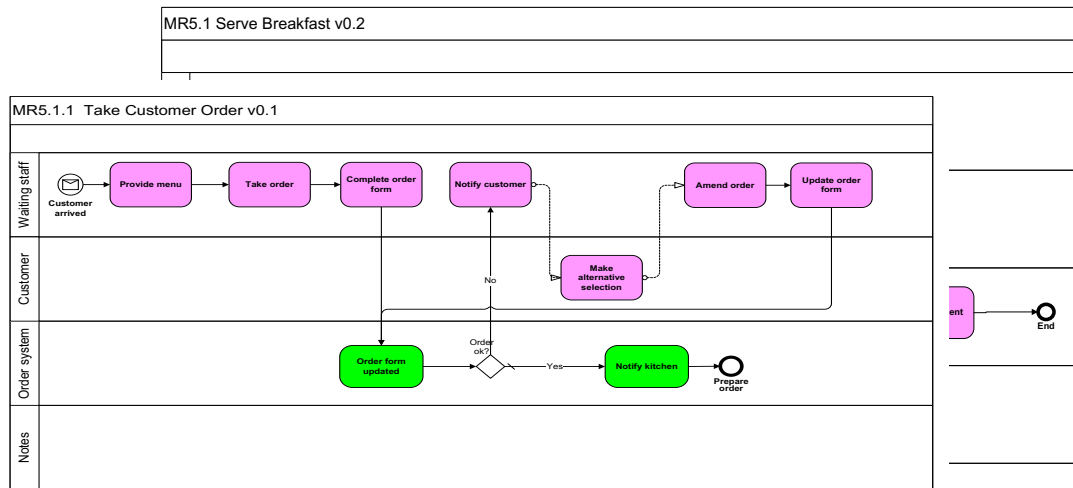


Figure 5.26: Process Modelling Examples for BAs

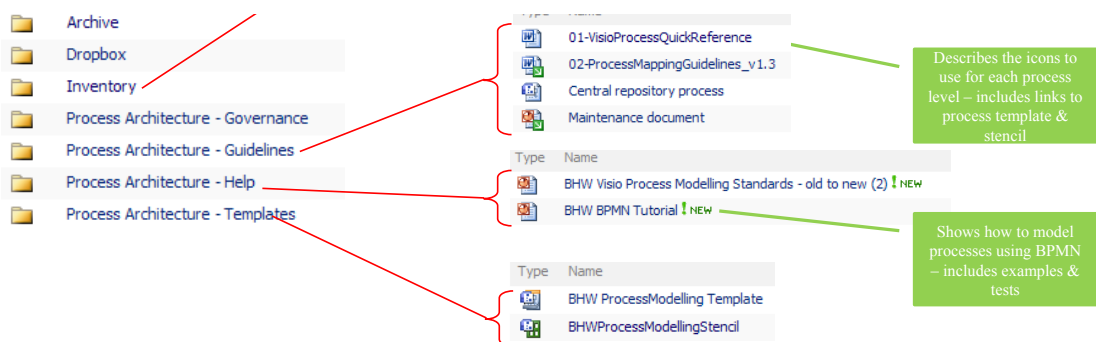


Figure 5.27: Process Modelling Training Materials

At the same time, Bupa is actively promoting a Lean approach [KVVH⁺06] towards project delivery. A dedicated team championed the use of Lean across the business, bringing both external experts and building internal capabilities. Lean was seen as core to achieving Bupa's long term targets. The extended KAOS framework is able to support the proposed Bupa Lean approach process, shown in Figure 5.28. Activities recommended for the Investigate and Improve stages have correspondents in our techniques. Specifically, the extended KAOS framework can be used to analyse process flows and develop potential solutions. Goal modelling can be used to identify risks and demonstrate new processes are performing better than before (i.e. "optimisation of new processes").

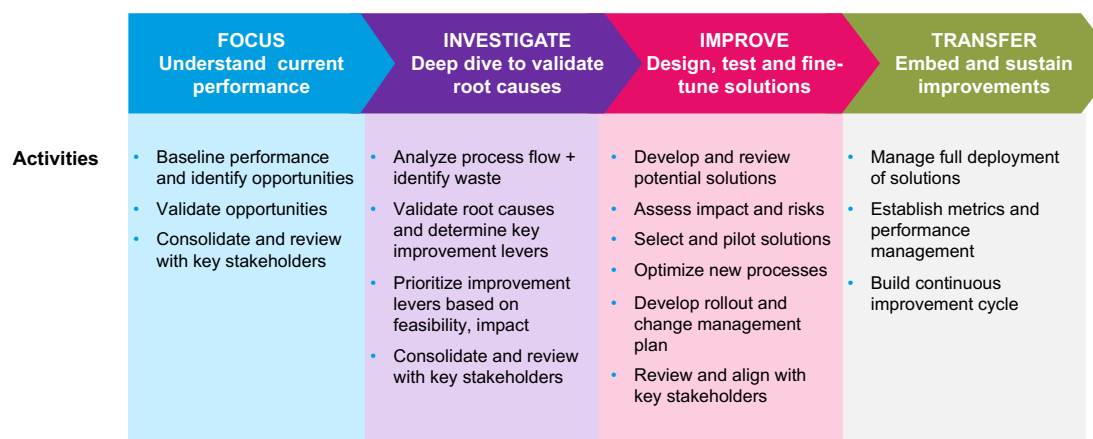


Figure 5.28: Bupa Lean Approach

5.8 Lessons Learned

In the previous section we discussed our findings specifically within the context of Bupa and the COPD project. We cover in this section a number of lessons learned which we believe to be valuable for other similar projects.

First, there is value in organising interviews with participants in the business processes, to collaboratively create process models. In our case, this meant interviews with the lead nurse and the COPD service manager. One benefit is gaining confidence that we document the *current* state of practice, and not rely on an idealised version of the process. In our case, the state of practice differed from the envisioned process described in existing documentation. Moreover, through interviews we were able to gather additional information about the nurses workflows, not immediately apparent when analysing the previously produced process models:

- what in the process should remain unchanged and why (for example, activities that directly correspond to recommended care pathways)
- what in the process should not be automated and why (for example, activities related to clinical decision making)
- what in the process is perceived as a bottleneck and what is the cause for the current state of affairs

- how much time each activity takes to complete, on average
- under what circumstances could participants deviate from the process and what is the procedure in this case

In Figure 5.19 we show one process model used in such an interview, with annotations from the COPD service manager. Thus, the process of collaboratively creating process models can be effectively used as a requirements elicitation tool - navigating the process model offers the opportunity for the process participants themselves to explain the goals and constraints shaping the workflows.

Second, the concept of intentional fragments is a useful construct when one tries to communicate to stakeholders the specific ways in which workflows have been improved, after deploying an IT system. In our COPD project, we started from an intentional fragment in the original workflow - for example taking consent when enrolling a new patient into the service - and identified the corresponding intentional fragment in the new workflow enabled by the Tolven EHR. Because there are activities in both workflows that satisfy the goal, we can demonstrate the capability hasn't been lost or overlooked in the new system. We can also articulate that in the new system that same capability is achieved through different activities, and we can compare the time and cost required to satisfy this specific requirement before and after the Tolven deployment. For the COPD project, our ability to clearly explain why the new workflow is better compared to the previous workflow has been recognised by industry peers - the COPD project has been shortlisted for a Health Service Journal award under the category Improving Efficiency Through Technology [HSJ]. Being shortlisted specifically for this category points to the strength of the argument put forward in the submission, linking technology to operational improvements (i.e. in the nurses workflows).

Third, the goal elicitation heuristics presented in Chapter 4 have been effective in building a meaningful goal model, which we have used to discuss alternative design options for the system-to-be with the wider project team. Goal modelling has not been more widely adopted within the Bupa business analyst community - one reason for this was the organisational focus on standardising existing practices, such as process modelling, before introducing new practices like goal oriented requirements engineering. However, we identified the opportunity to collaborate with the Bupa team responsible for supporting Lean practices across the organisation. We presented the KAOS framework and the process based requirements elicitation techniques to the team, and their feedback was that our techniques could be used in Lean driven projects.

Chapter 6

Evaluation: Personal Health Record for Nuffield Health

In this chapter we present the analysis and design of a Personal Health Record system, developed in partnership with Nuffield Health. Section 6.2 discusses the differences between the Bupa and the Nuffield Health projects and argues how the two evaluations complement each other. Section 6.3 presents the business need for a PHR system to support new customer journeys. We are conducting this project as action research and the following sections correspond to the action research cycle: Section 6.4 reviews the initial state of affairs, Section 6.5 describes the requirements elicitation phase based on our goal inference heuristics and Section 6.6 presents the architecture and functionalities of the final system. We review the results of the project in light of our research questions in Section 6.7.

6.1 Rationale

This thesis investigates requirements engineering methods based on process modelling that business analysts can use for EHR systems.

As discussed in Chapter 1 and Chapter 2, EHR systems are used across the full spectrum of care, by healthcare professionals, administrative staff and patients. Chapter 5 evaluates the role of our requirements engineering techniques when developing an EHR system to support delivery of a clinical service. Intended users of the system were specialist nurses and the service administrator.

However, EHR systems are increasingly used to support Digital Health products and services, available to end customers. We consider Digital Health to include online services (emails, web-sites, mobile applications, social networks) aimed to empower individuals to take action in relation to their health and wellbeing. Digital health applications are meant to facilitate coordination between providers and consumers of healthcare services, to support population health programs, chronic condition management, addressing lifestyle risk factors through behaviour change, or better access to information [Com12], [Lup13].

The process of designing and building a health record in the consumer oriented space is different from the process of building an EHR for clinical care, in several aspects.

First, the role of business analysts within the overall project delivery methodology is different - they

operate at the intersection between user centred design and agile software development [BMMW15]. In our collaboration with Bupa, which illustrates how EHRs are designed and developed in a clinical context, users of the system are clearly categorised based on their role in care provision (nurses, GPs etc.). Business analysts are able to document the users' current or future workflows, through interviews and direct observation. Business analysts then define the software requirements, prioritise and communicate them to the development team, and demonstrate the project benefits. When developing digital health products the users of the system are members of the general public with varying interests and goals. Business analysts working on such projects are not directly responsible for requirements elicitation - they do not interact with the end users. Our experience in digital health projects, including our work with Nuffield Health, was that user experience designers are the ones responsible for conducting user research and then producing target customer journeys and wireframes. This practice has been documented elsewhere in the industry as well [FSR10]. The customer journeys and wireframes are then used to produce more detailed interaction scenarios and user stories, which are then assigned to the development team.

The role of the business analysts is then to review customer journeys and wireframes and formulate requirements, while taking into consideration any other organisational goals and constraints. In other words, they need to engage with different roles in the delivery team (UX researchers as opposed to end users) and have access to different type of information (envisioned customer journeys as opposed to business processes and workflows). As such, we wish to explore whether our techniques - originally designed to assist business analysts in analysing process models in order to elicit software requirements - are still relevant in a project that combines user centred design and agile development.

Second, there are both similarities and differences between customer journeys and business processes, and as such we want to investigate whether analysing customer journeys helps business analysts shape the software specification (in the same manner as analysing process models does).

On the one hand, both customer journey maps and process models represent sequences of steps that individuals may take, in pursuit of a goal. On the other hand, customer journey maps are not as precise as process models. They hide some of the technical and organisational complexity in their focus on representing the customer perspective, and use mock-ups of the user interface to graphically represent each step in the journey. Moreover, the sequence of steps in a customer journey map will illustrate one way of achieving a goal, but not necessarily the only way. In contrast, business process models capture standard practice within an organisation or, in the case of clinical pathways, recommended practice based on medical guidelines. Consider for example a customer journey map that illustrates how visitors could interact with a website in order to book a health assessment. Every visitor could accomplish this goal by following different paths on the website. One could start from browsing through the different types of assessments, another could use a calendar to first find a suitable date, while yet another could check the locations where health assessments are available, before deciding on the type. A business analyst may still identify the minimum number of activities required to book the health assessment, but this does not mean all users are expected to follow those exact same steps, or that deviations from the envisioned

journey pose an inherent risk. Certain variations of the envisioned customer journey may be safe, as in our previous examples, while other variations should be avoided - such as paying for a health assessment before completing the booking.

Third, the customer journeys and wireframes (the output of the user experience design) abstracts the architectural complexity. In our experience with Nuffield Health, the envisioned customer journey was reliant on a number of independent software systems interacting and sharing data, as opposed to one monolithic system (as was the case with the Bupa COPD project). As such, we want to assess whether our techniques help identify the optimal way to assign software requirements to system components. This area was less explored in our collaboration with Bupa, as the EHR system was designed to support the complete range of requirements.

Given these differences, it is meaningful to evaluate how the extended KAOS framework fits in this new context of use, which is representative for many recent applications in the digital health field.

6.2 Research Questions

We have reviewed in the previous section why a second case study is necessary. In summary, we are evaluating our techniques in a different software delivery process - in Nuffield Health, the development team is using story mapping and behaviour driven development [WH12]), with different input (customer journey maps) and with a different focus (a new consumer facing digital health proposition).

Our research questions mirror the ones in Chapter 5:

RQ1: What was the impact of the PHR platform to the business and the external stakeholders?

As a health and wellbeing organisation, Nuffield Health has identified an unmet need in the industry, for a platform that service providers may use to collect, manage and analyse lifestyle data in order to provide a personalised service to customers. While this is relevant for Nuffield Health, it can equally benefit other healthcare organisations.

RQ2: To what degree did our method contribute to the project outcomes and to its impact?

The experiment presented in this chapter aims to investigate whether our techniques generate useful insight and facilitate decisions in regards to the architecture and capabilities of the PHR system. To apply our goal inference techniques, we use as input customer journey maps - while these are used by user experience designers, we have not considered them in the experiment presented in Chapter 5.

RQ3: How does our method fit with the organisational context?

We also aim to assess the applicability of our techniques in environments that follow behaviour driven development - an Agile methodology actively used in the industry [BEK18].

Given the author was a Nuffield Health employee during the project, we aim to answer these questions following the action research cycle presented in Chapter 5.

6.3 Context and Purpose for a Personal Health Record

As of 2016, Nuffield Health owns more than 110 consumer gyms spread across the UK. In early 2016, a project was initiated which aimed to deliver a digitally enabled customer journey for gym members. The

initial focus was on online gym joining, followed in 2017 by additional services, such as online class booking and buying personal trainer sessions.

As more and more interactions between Nuffield Health and its customers happen online, business owners were interested to develop a IT platform that would support and enhance the different services available to customers and would offer a connected user experience. At the same time, Nuffield's strategy was to use data led insight to offer personalised advice and support.

Figure 6.1 shows a fragment of the envisioned user experience. It concerns a hypothetical Nuffield Health customer who accesses the physiotherapy service. His experience will be enhanced through online access to his recovery plan. If the individual adds a connection to his wearable devices, his physiotherapist could adjust the intensity of the sessions, based on data available in the system. Furthermore, once the physiotherapy programme ends, his personal trainer could review all the previous data and build upon this to design future gym workouts. A customer could book these sessions and track his progress towards the goals agreed with the health experts.

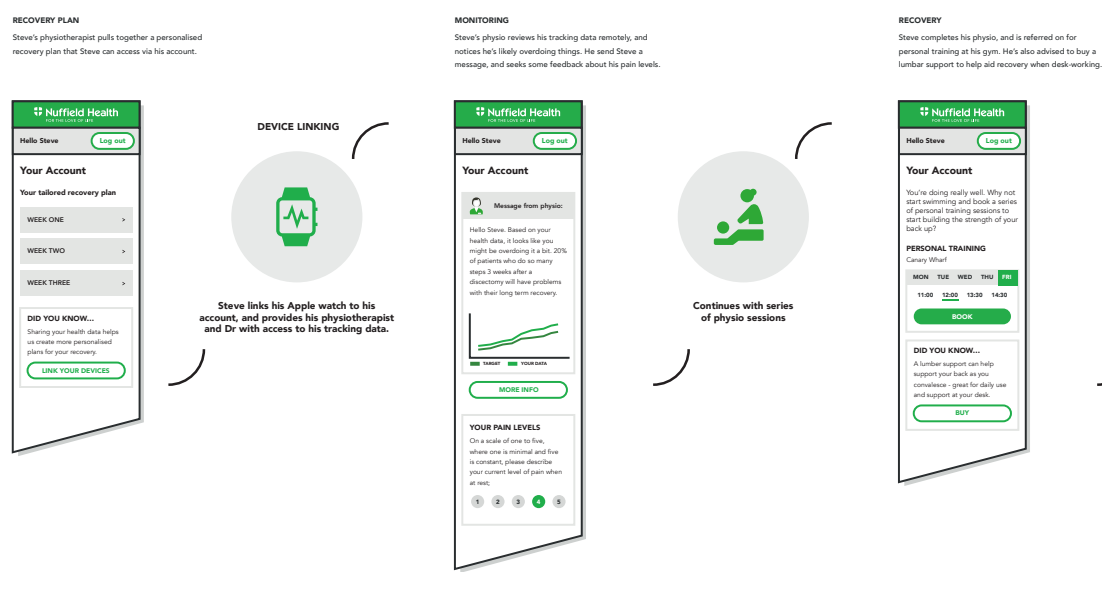


Figure 6.1: Post Physiotherapy Journey

To support this journey, there was a business need to develop a Personal Health Record platform, where data generated through customer interactions with Nuffield Health services would be saved, analysed and made available to both customers and staff, to inform a personalised journey.

The scope for the PHR was not collecting medical data such as history of hospital episodes, medication history, allergies, pathology tests or medical imaging - these are commonly managed by EHR systems. Rather, we focus on data underpinning digital health initiatives (we refer to this as lifestyle data). This includes data collected by mobile apps, smart home devices, wearables, data which is self-reported via online surveys, and data generated by consumer oriented services (gyms, weight loss programs, personal training, physiotherapy, CBT, health assessments, risk calculators etc).

The PHR platform should also allow Nuffield Health to share data it collects about consumers with the individuals themselves, in an open, standardised format. For example, one's activity in the gym (at-

tendance patterns, classes booked) or the data collected during a private health assessment. Conversely, it should allow consumers to share data with Nuffield to get a more personalised service. For example, sharing fitness data from a wearable device with a personal trainer or physio therapist, to inform a personalised exercise programme. Finally, machine learning models should be developed based on the data collected in the PHR and then used to assist customers with their individual wellbeing goals.

A collaboration between Nuffield Health and UCL has started in 2016, to design an open source PHR platform that could deliver on these high level goals.

6.4 Diagnosis

The Nuffield Health PHR project aimed to develop a platform that would integrate with existing IT systems and enable customers to have a personalised and connected experience with Nuffield Health, across the range of services available to them. The platform should support the envisioned post-physiotherapy journey, as well as other customer journeys involving Nuffield Health services.

For the first release of the system, the author acted in a business analyst capacity, and the implementation work was assigned to UCL students. A Nuffield Health data architect, as well as the digital director, were responsible for approving the design, evaluating the final system and providing control and governance during the project.

To evaluate the impact of our theoretical contributions in the design of the Nuffield Health PHR, we first review the initial state of affairs, at the point when the UCL collaboration started.

First, we present the target experience for a customer who had just finished his physiotherapy sessions in more detail, as this will be our main source of requirements. Second, we present an overview of the IT architecture in place at Nuffield Health and the technologies considered for the PHR. Third, we present the software development practices employed by the technical team. These are the three aspects which our theoretical contributions aim to connect: our interest is in how the target customer experience drives the specification of the PHR system and consequently how this KAOS based specification aligns with the software delivery process employed by Nuffield Health.

6.4.1 The Target User Experience

A high level vision has been set through the user experience diagram for customers completing physiotherapy (Figure 6.1). A customer should be able to navigate seamlessly between different Nuffield Health services, such as physiotherapy and personal training. Information held by Nuffield Health, such as the schedule of the physiotherapy sessions, should be enriched with consumer held information, like wearable device data. A consolidated view should be accessible online, presenting the individual's evolution against her targets.

The next level of detail is captured through a number of specific scenarios, presenting possible interactions between customers and digital channels owned by the organisation (for example, the Nuffield Health website, or emails sent by Nuffield Health). One such scenario is presented in Figure 6.2, covering the transition between physiotherapy and personal training. This visual representation of how a user might interact with Nuffield Health constitutes the main source of requirements at this stage. After a

physio appointment, a consumer would receive an email with several calls to action. If she decides to go to a Nuffield Health gym, she will be able to use the Nuffield Health website to book additional services, to read articles relevant to her interests, or to keep track of her fitness goals.

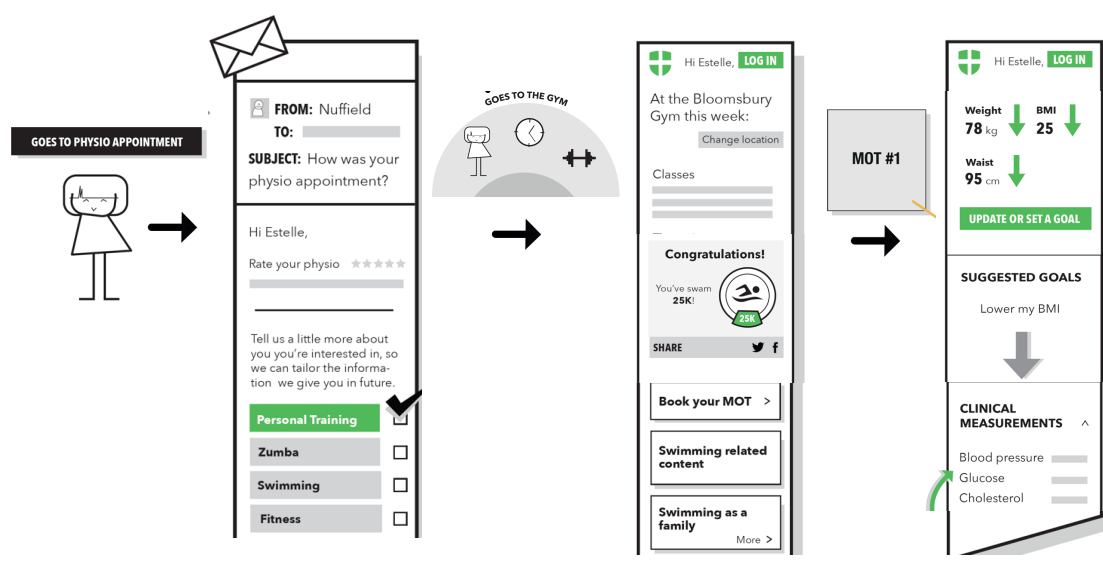


Figure 6.2: Post Physiotherapy Journey (Detailed)

Comparing this graphical representation of a user journey with the KAOS process models, there are a number of differences. This journey captures requirements from a customer perspective: users will be able to rate the physio programme, to book an MOT, to view clinical measurements etc. The business expectations on the IT systems are not explicit - functionalities are merely represented as elements on the UI. This hides complexity - for example, it is not clear under what circumstances a customer will be able to book an MOT, or what functionality is required to manage the bookings. Moreover, the customer motivations or benefits from these interactions are not explicit. For example, the user journey does not explain what is the purpose of highlighting achievements (e.g. 25k swim).

As was the case with BPMN process models, when used in isolation, this type of diagram does not provide sufficient clarity on each agent's responsibilities and motivations. It describes the behaviour of the system from a user's perspective, but hides the details of *Why* that behaviour is necessary or *What* the requirements for the software are.

6.4.2 Architecture Overview - FHIR for Fitness and Lifestyle Data

The Nuffield Health PHR platform was expected to integrate with existing IT systems, used within Nuffield Health to support service delivery. The purpose of the PHR was not to replace existing systems - rather, the PHR platform was expected to provide additional functionality required to deliver the target user journeys.

Two broad areas of functionality have been identified from the beginning of the project. First, aggregating personal health information and reporting on it, both at an individual and population level. This is illustrated in Figure 6.3. Information stored in the PHR should include data entered by individuals as well data tracked by devices. It could also include vital signs and basic test results (cholesterol, glucose

etc.). The PHR data will be linked to other datasets and ultimately create a unique customer view.

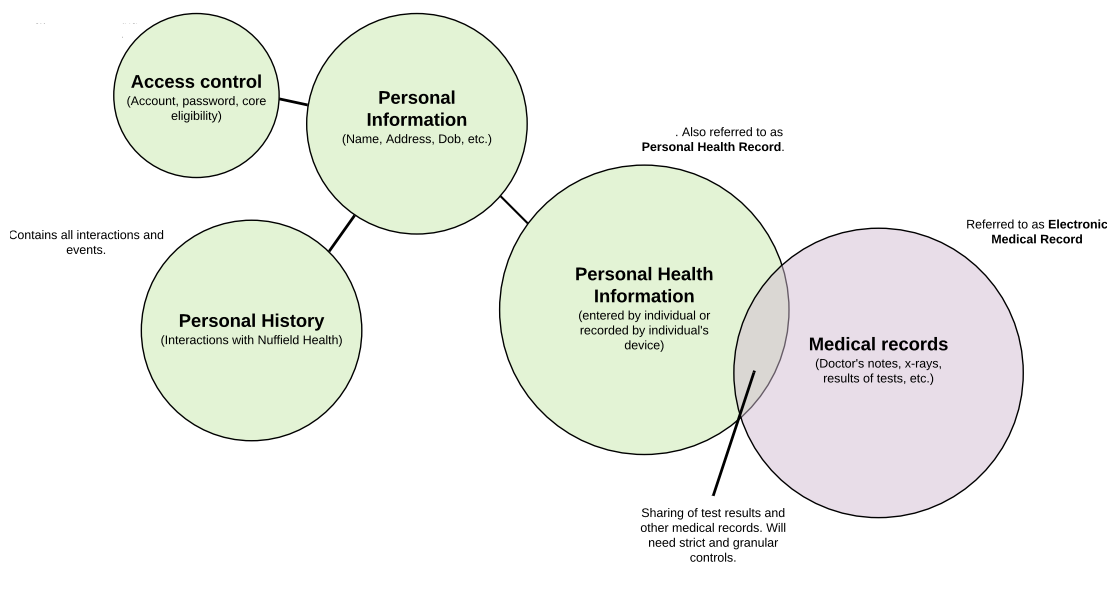


Figure 6.3: Information Architecture

There are specific challenges when managing this type of data. First, we need to account for the varying degree of reliability of the data sources: looking at a simple measure such as weight measured in kilograms - this could be self-reported, retrieved from a smart scale, or measured during a health assessment. Second, information may exist in different formats: looking at the quality of sleep - this could be captured as qualitative feedback from an individual (e.g. 'I feel rested when I wake up'), a set of numerical values (e.g. total duration 8 hours, woke up 2 times) or a proprietary format coming from a wearable device. Third, this type of information can be interrogated for varying purposes: a specific measure of interest for a specific individual (e.g. evolution of weight over time), sharing the full record of one individual with a third party (e.g. a GP), population level reporting (e.g. risk stratification based on BMI levels).

Consequently, there was a business need to identify a common data model for the lifestyle data to address these challenges.

Reviewing internal data standards used within Nuffield Health IT landscape, the Fast Healthcare Interoperability Resources (FHIR) standard [MKM⁺16] was already in use by two systems relevant for the PHR. FHIR is the latest standard released by HL7, the organisation developing standards for the exchange, integration, sharing, and retrieval of electronic health information. FHIR resources represent information models for the most common entities in the healthcare domain (e.g. Patient, Observation, Care Plan). The FHIR resources are implemented in XML or JSON, and can be retrieved via RESTful interfaces. To ensure consistency, the project team recommended the PHR platform would be build around the FHIR standard. However, at the project initiation stage, it was not clear whether FHIR can indeed support the type of data items and the analysis expected out of the PHR platform.

The second area of functionality required for this project was enabling content personalisation and

advanced analytics and machine learning, informed by data held in the PHR. Content personalisation was considered by the Nuffield digital team to be an effective way to promote long term user engagement. Since Nuffield Health already owned health information articles, the business sponsors for the project wanted the ability to recommend specific articles tailored to each individual interests. Another area of interest was the ability to offer personalised lifestyle suggestions to customers. For example, helping individuals set feasible goals, they could attain, considering their current fitness level and motivation.

From a technical perspective, there was also a need to design the infrastructure that could support data scientists to train predictive models on the data available in the PHR.

6.4.3 Software Development Processes

The Nuffield technical team was planning to use Behaviour Driven Development (BDD) [SW11] to manage delivery of digital projects. BDD is well aligned with the KAOS framework for requirements engineering. BDD focuses on specifying the behaviour of the target system, in a manner that is suitable to automated testing.

The BDD process starts from describing what should the system do (i.e. a set of behaviours). The system behaviour in relation to a specific outcome is described as an executable feature file. In Cucumber [WH12], one of the main BDD frameworks, a feature file will contain a plain text description of the feature and any relevant business rules. We give below an example for an ATM system:

Feature	Customer withdraws cash As an account owner I want to withdraw cash from an ATM so that I don't have to visit a branch.
----------------	--

A feature description will also contain one or more scenarios. A scenario describes the expected behaviour of a system, in specific circumstances. The template for writing scenarios follows a tree part structure: Given - When - Then. Under each heading, the scenario will have a number of steps (i.e. conditions that hold true in the system). In its entirety, a scenario describes a state transition: *Given* a system state, *when* an even occurs, *then* the system is in a new state.

Scenario	Withdrawal when account is in credit
Given	Given the account contains £100 And the card is valid And the dispenser contains cash
When	When the customer requests £20
Then	Then account contains £80 And £20 cash is dispensed And the card is returned

Since it puts actions and behaviours at the core of the specification, BDD as a software delivery process is closely aligned with the KAOS framework for requirements engineering.

6.5 Planning Action: Requirements Gathering

We discuss in this section the process we followed to identify a set of requirements for the Nuffield Health PHR system.

The overall objective for the PHR system was to support new types of digital interaction between Nuffield Health and its members. Previous work conducted by a digital agency working closely with stakeholders within the organisation resulted in high level customer journey maps encompassing different touch points between customers and Nuffield. Each touch point was described along three directions: the graphical interface presented to the user, the core software capabilities required during the interaction, and descriptions of how human actors interact with the software. An example is given in Figure 6.13. Our focus in the requirements elicitation phase was to convert these customer journey maps into process models and analyse them using our process based requirements elicitation techniques. The objective was to identify capabilities that fall within the scope of a Personal Health Record platform. To validate the system specification, we worked closely with the Nuffield Health software architects. We did not conduct user research ourselves, since the customer journey maps had already captured the results of prior and extensive user research.

We also had regular meetings with the members of the digital and marketing teams in Nuffield Health, as well as the external agency responsible for delivering new functionality for the Nuffield website. During these meetings, we gathered more information about work in progress at the time. For example, in 2016 - 2017 there was ongoing work on building a "Member Page" on the website, available to gym members. At the same time, the marketing team was developing a customer retention strategy for gym members, aimed at improving the rate of contract renewals. These meetings offered additional insight that helped us develop the goal model we present in this chapter.

6.5.1 Customer Journey as a Process Model for the System-to-Be

To start eliciting requirements for the Nuffield Health PHR, we have developed a process model to capture one of the scenarios considered in the Nuffield strategy: a connected user experience involving physiotherapy, gym sessions and a Health Assessment (the "Nuffield Health MOT", available to all gym members). The process model has been developed based on the illustrative user journeys presented in Figures 6.1 and 6.2.

We give below a description of the process. We also discuss if the proposed KAOS process modelling syntax is able to capture all the required information, given the different domain compared to the Bupa project (a customer facing application VS a clinical IT system). The complete process model is shown in Figure 6.4.

The process starts once a customer attends his physiotherapy session. The system should keep track of this event, which then triggers an email to the customer. As the same email contains two different calls to action (*give feedback* and *choose interests*), we use a parallel gateway with two outgoing activities. We also note that the signature of each activity is used to capture the business rule specifying when the email should be sent. Specifically, a customer should receive an email asking for his feedback no later than 24 hours after his session.

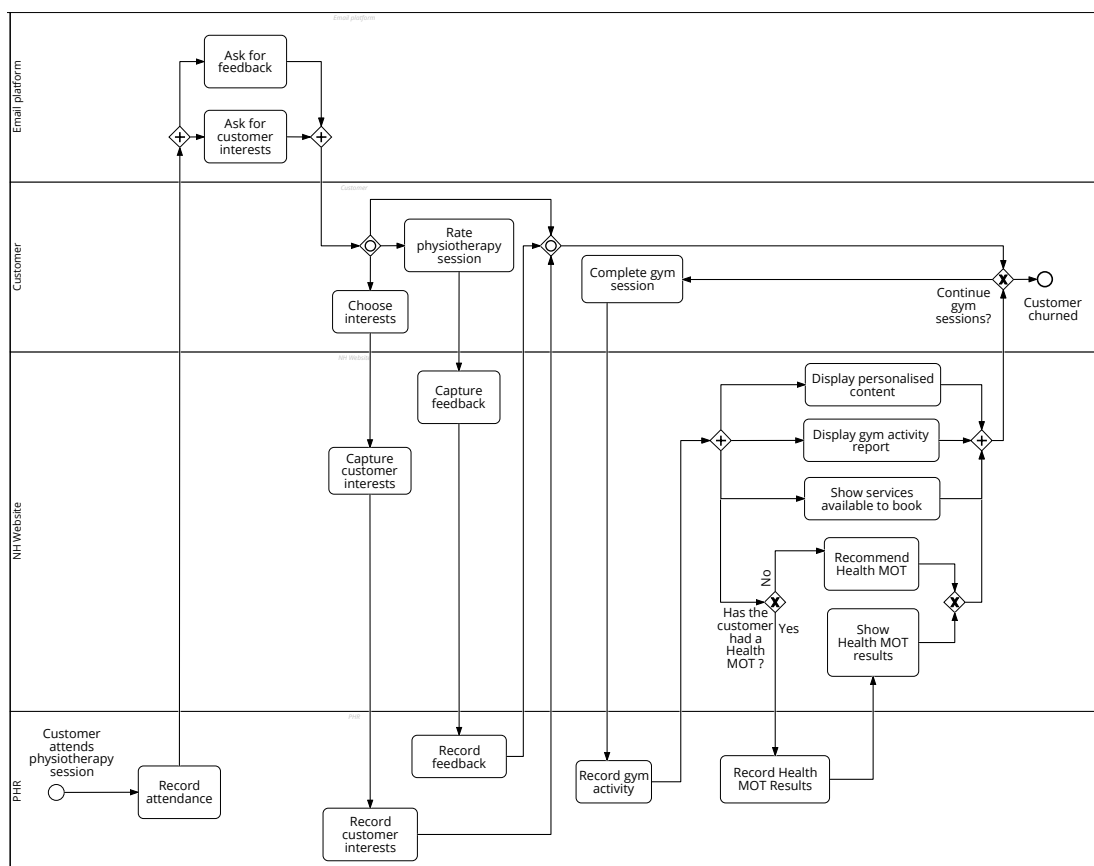


Figure 6.4: Nuffield Health PHR - Example Process Model

Activity	<i>AskForCustomerFeedback</i>
Domain pre-condition	\neg sessionRated \wedge attendanceRecorded
Required post-condition for goal <i>Achieve[CustomerSatisfactionMeasured]</i>	feedbackRequested
Maximal allowed delay	24 hours

Once the customer receives the email, there are a number of ways he can proceed. He may choose to respond to none, one or both calls to action: he may use the website to offer feedback, to choose his interests, or he may ignore both requests. If the customer does offer information, this should be captured in the PHR system. Syntactically, to describe a situation where an agent may follow any number of paths we use a different notation - inclusive gateways. Compared to exclusive gateways, they allow more than one outgoing sequence flow to be active during a process execution.

Next, the customer has the chance to continue with his gym training regime. Each time he attends the gym, the PHR should record his activity. If the customer has a health MOT while at the gym, the PHR should also record the clinical measurements collected during the Health MOT.

Based on the information collected about each customer, the website will always display a report of his gym activity when the user is logged in, as well as suggested articles and Nuffield services that match the customer's interests. There is an additional business rule that checks whether the customer has had a Health MOT and then recommends one to any gym member who has not had it yet. Otherwise, the

website should show the results of the health MOT, including lifestyle goals agreed with the physician.

Syntactically, we use a loop in the process model, to show the fact that at any point, even before his first gym session, a customer may decide to stop attending the gym - then the process will end.

6.5.2 Goal Model for the System-to-Be

In this section we present the intentional fragments identified based on this process model and discuss the resulting goal model. Our purpose is to identify specific software requirements based on the envisioned customer journey. As several systems are required to interoperate in order to deliver on the high level goals, we also explore different ways in which requirements can be assigned to agents, and how these assignments impact the overall system design. As such the final goal model is also used to establish clear responsibility assignments, to inform the system architecture and interoperability requirements.

The process model has only one end event - the `CustomerChurned` event signifies that a physiotherapy customer is no longer engaged with the gym. Churn can be defined as: the customer is not entitled to attend the gym anymore (for example, he has not renewed his gym membership) or the customer has not attended the gym for more than a set period of time (for example, he has not been to a gym for more than two months).

As with many consumer oriented services, one of the business objectives for this process is to delay the end event as much as possible. Hence, the most appropriate goal inference heuristic is H2, which yields goals of type *Avoid*, based on the start and end events of the process. The resulting goal definition is given below:

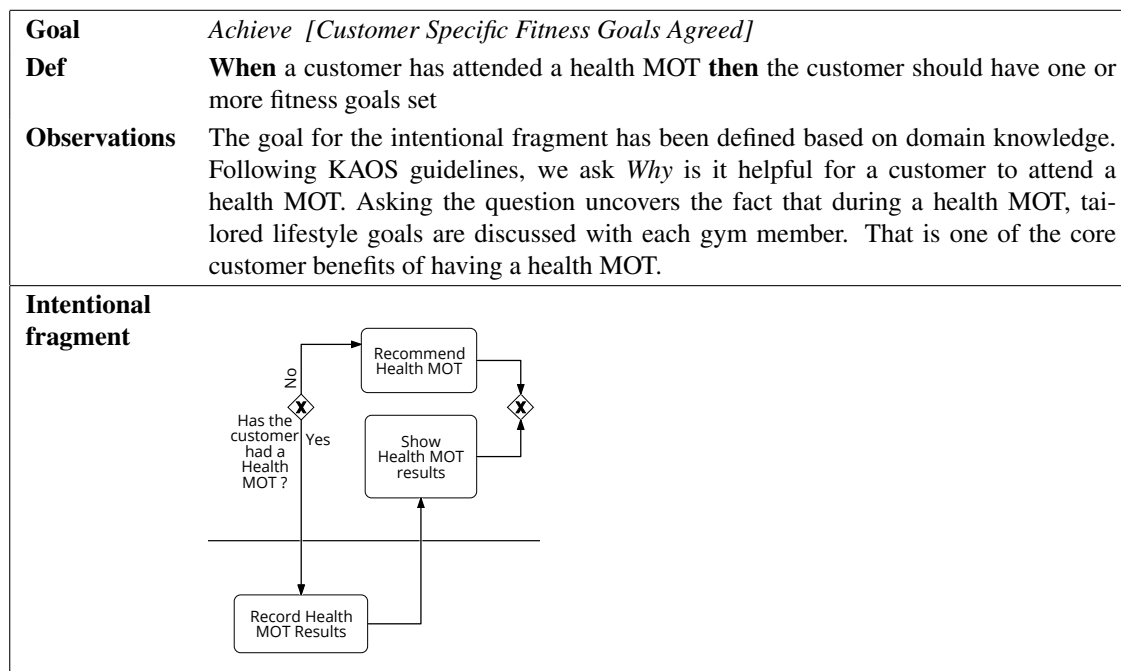
Goal	<i>Avoid [CustomerChurned]</i>
Def	When a customer attends a physiotherapy session, then the customer should not churn
Formal Def	$customerCompletesPhysiotherapy \Rightarrow \square \neg customerChurned$
Observations	This is an idealised goal, as customers will eventually churn. However, from a customer centric perspective, Nuffield Health's mission is to avoid this as much as possible.

Since the functional goal of the service is not apparent from the process model, we have to rely in this instance on domain knowledge. We define the goal *Achieve [Customers Fitness Levels Improved after Physiotherapy]* as follows:

Goal	<i>Achieve [Customers Fitness Levels Improved after Physiotherapy]</i>
Def	When a customer completes a physiotherapy programme, then he should eventually regain his fitness level
Formal Def	$customerCompletesPhysiotherapy \Rightarrow \diamond customerFullyRecovered$
Observations	Part of Nuffield Health's strategy is to help customers regain their physical fitness. As we have shown in Figure 6.1, this includes a scenario where the physiotherapist recommends his customers to attend the gym once physiotherapy completes, for them to regain their strength.

Next, we note that the process model has a large number of gateways that control the process flow, relative to the number of activities. However, there is only a pair of exclusive gateways. One of the

heuristics of heuristic H6 applies in this case - we identify an intentional fragment composed of the activities between the pair of exclusive gateways.



At this stage, we can create a high level goal model - shown in Figure 6.5.

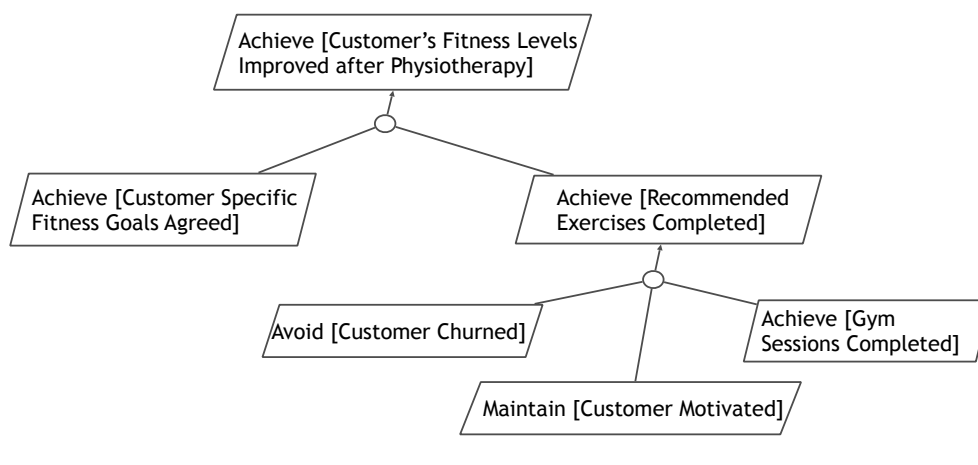


Figure 6.5: Post Physiotherapy Goals

Through interviews with domain experts, we validate that two of the goals identified so far are indeed in a refinement relation. Specifically, if a customer is to improve his fitness levels after physiotherapy, it is important he agrees specific and explicit goals with a health expert - this is equivalent to a care plan in a clinical setting. To complete the refinement of the high level goal, we introduce an additional goal:

Goal	<i>Achieve [Recommended Exercises Completed]</i>
Def	When a customer has a set of fitness goals then the customer should complete the recommended exercises

The refinement of the above goal is inspired by behaviour change theory, specifically the COM-B model [MvSW11]. In order for an individual to engage in an activity (in this case complying with the recommended exercise regime), the individual should have:

- Opportunity: they should attend the gym (Avoid [Customer Churned]).
- Capability: they should be physically able to complete their exercises (Achieve [Gym Sessions Completed])
- Motivation: they should want to complete the exercises recommended to them (Maintain [Customer Motivated])

Next, we continue applying heuristics to identify intentional fragments, to further refine the high level goals identified so far.

Considering the pair of exclusive gateways in the process and the goal Achieve [Customer Specific Fitness Goals Agreed], we can apply heuristic H11 - the case driven refinement pattern. The refinement will be based on the condition on the exclusive gateway: Has the customer had a health MOT?. To further refine the goals, we can apply heuristic H12, which introduces a milestone refinement pattern. The goal refinement derived from the process fragment is shown in Figure 6.6. The goal model clarifies that the Nuffield PHR system has the responsibility to capture the Health MOT results, including the goals, for each individual. We also observe that the goal model has an incomplete refinement for the goal Achieve [Fitness Goals Defined if Health MOT Not Done]. In practice, this means that although goal setting is an important component when trying to help individuals improve their fitness levels, the current process model does not ensure the goal is satisfied under all circumstances. It appears that customers who do not attend a health MOT will not have personalised fitness goals.

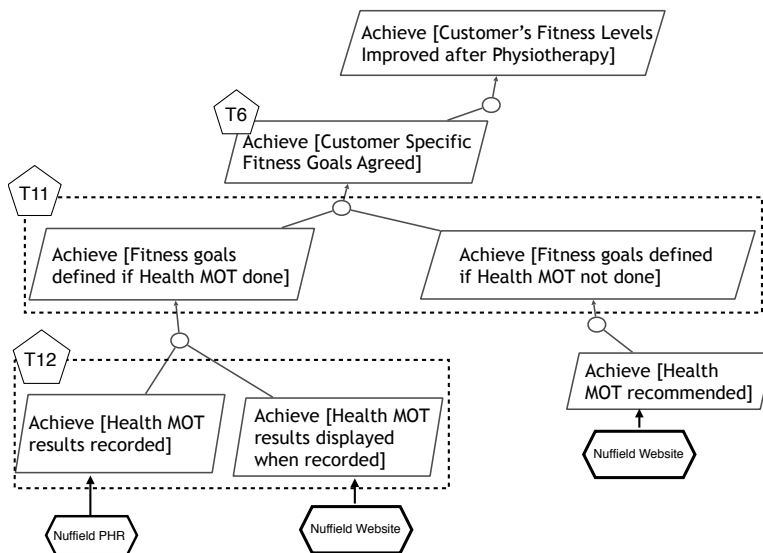


Figure 6.6: Post Physiotherapy - First Goal Model Refinement

Heuristic H6 also proposes that a closed loop in the process model is a structural pattern indicative

of an intentional fragment. The process model for the physiotherapy customer does include a loop, shown in Figure 6.7.

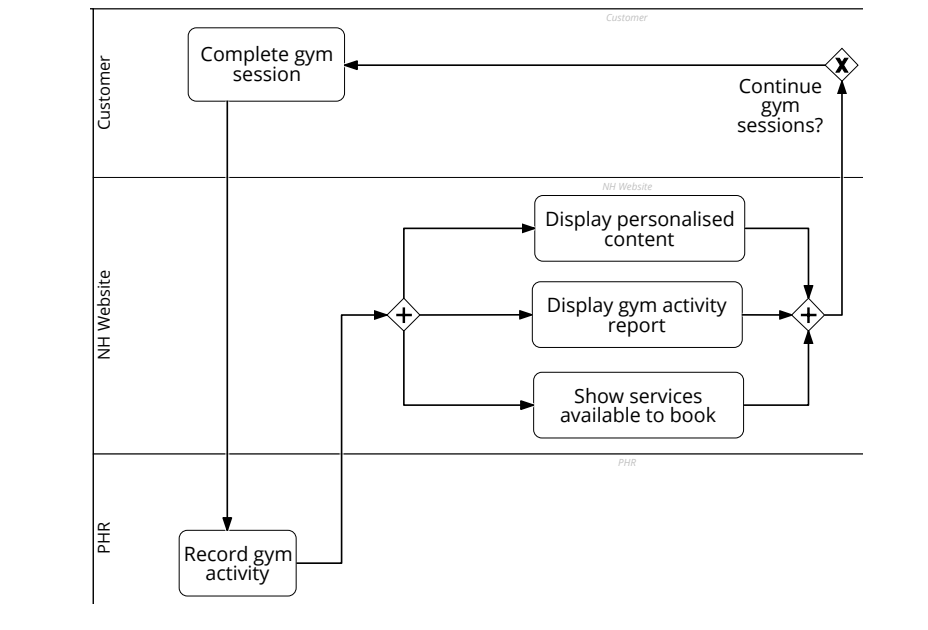


Figure 6.7: Nuffield Health PHR - Heuristic H6

Writing the pre and post-conditions for the activities in the loop shows that there is more than one goal to be considered.

We start from the activity `CompleteGymSession`. As our specific scenario concerns customers transitioning from physiotherapy to personal training, there is a pre-condition that each gym session has to be scheduled beforehand with the personal trainer.

Activity	<i>CompleteGymSession</i>
Domain pre-condition	$\text{gymSessionScheduled} \wedge \neg \text{gymSessionCompleted}$
Required post-condition for goal <i>Achieve[GymSessionsCompleted]</i>	$\text{gymSessionScheduled} \wedge \text{gymSessionCompleted}$

The next activity in the loop has as a pre-condition the post-condition of the first activity. Namely, when a customer completes a session in the gym, the PHR platform should record this activity. The benefit of this is that by monitoring each customer's physical activity, the system will be able to track their progress against their goals.

Activity	<i>RecordGymActivity</i>
Domain pre-condition	$\text{gymSessionCompleted} \wedge \neg \text{gymActivityRecorded}$
Required post-condition for goal <i>Achieve[CustomerProgressTracked]</i>	$\text{gymActivityRecorded}$

The activity `DisplayGymActivityReport` illustrates one way in which the customer progress can be presented - by showing reports on the Nuffield Health website, for logged in users.

Activity	<i>DisplayGymActivityReport</i>
Required pre-condition for goal <i>Achieve[CustomerProgressTracked]</i>	<code>gymActivityRecorded ∧ customerLoggedIn</code>
Required post-condition for goal <i>Achieve[CustomerProgressTracked]</i>	<code>activityReportPresented</code>

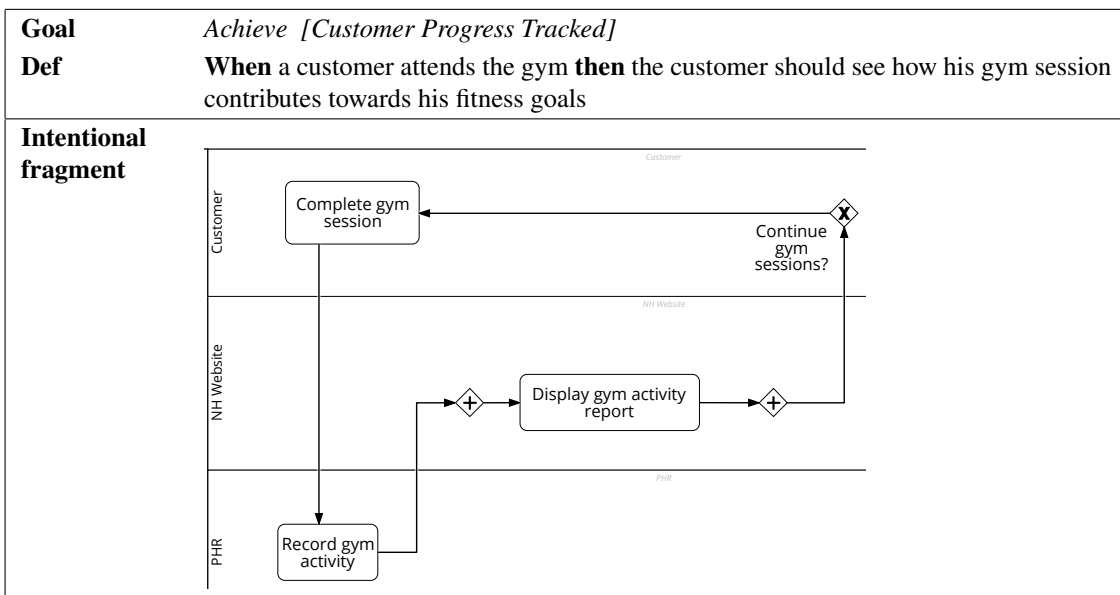
We also observe that some of the activities in the loop do not contribute to the goal identified so far - `Achieve [Customer Progress Tracked]`. For example, the activity `DisplayPersonalisedContent` shows how the customers' interests should be used to select content relevant for him.

Activity	<i>DisplayPersonalisedContent</i>
Required pre-condition for goal <i>Achieve[RelevantDigitalContentOffered]</i>	<code>customerInterestsRecorded ∧ customerLoggedIn</code>
Required post-condition for goal <i>Achieve[RelevantDigitalContentOffered]</i>	<code>personalisedContentDisplayed</code>

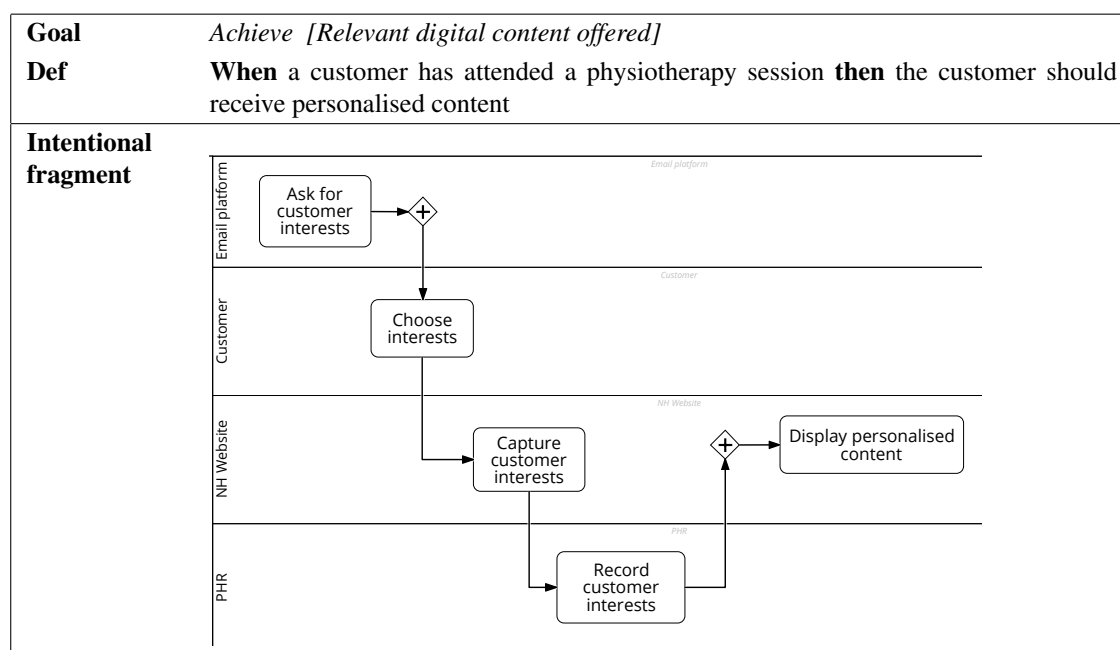
Finally, the system should also recommend additional services that might benefit customers based on their interests. The activity `ShowServicesAvailableToBook` contributes to this goal:

Activity	<i>ShowServicesAvailableToBook</i>
Required pre-condition for goal <i>Achieve[RelevantServicesRecommendedToCustomers]</i>	<code>customerInterestsRecorded ∧ customerLoggedIn</code>
Required post-condition for goal <i>Achieve[RelevantServicesRecommendedToCustomers]</i>	<code>relevantServicesDisplayed</code>

Considering the signatures of the activities, we apply heuristic H7, to identify cases when the post-condition of one activity is a required pre-condition of another activity. This yields the goal definition for `Achieve [Customer Progress Tracked]`:



Next, we search the process for other activities contributing to the goal Achieve [RelevantDigitalContentOffered], since the activity DisplayPersonalisedContent has been identified to contribute to this goal.



The goal Achieve [Relevant Digital Content Offered] can be further refined using heuristic H12: since the activities composing the fragment are performed by different agents, the goal will be refined using the Milestone refinement pattern, to show the explicit responsibilities of each participating agent.

The resulting goal model is presented in Figure 6.8. We show two different ways in which Nuffield Health aims to keep customers motivated to attend the gym. First, by presenting users up to date reports of their activity and their progress towards achieving their goals. This was illustrated in Figure 6.2 through the "Achievement" notification, when a customer covers 25km swimming. Second, the system aims to support individuals by offering relevant digital content, selected based on personal interests and goals. Educational content is another tactic commonly used in digital health and behaviour change interventions, to increase someone's determination. The goal model also shows the milestone refinements derived through heuristic H12. These refinements are typical of cases when a user interacts with a web platform: the usual agents in such scenario, a user, a front end application and a back end engine, have to collaborate such that the user's goals are achieved. In our case, considering the goal of recording a customer's interests, this is achieved by the website capturing the user input and then saving this selection into the PHR. Finally, the goal model also shows another high level goal - Nuffield's desire to raise awareness of the wide range of services available to customers, in a way that is relevant to their interests.

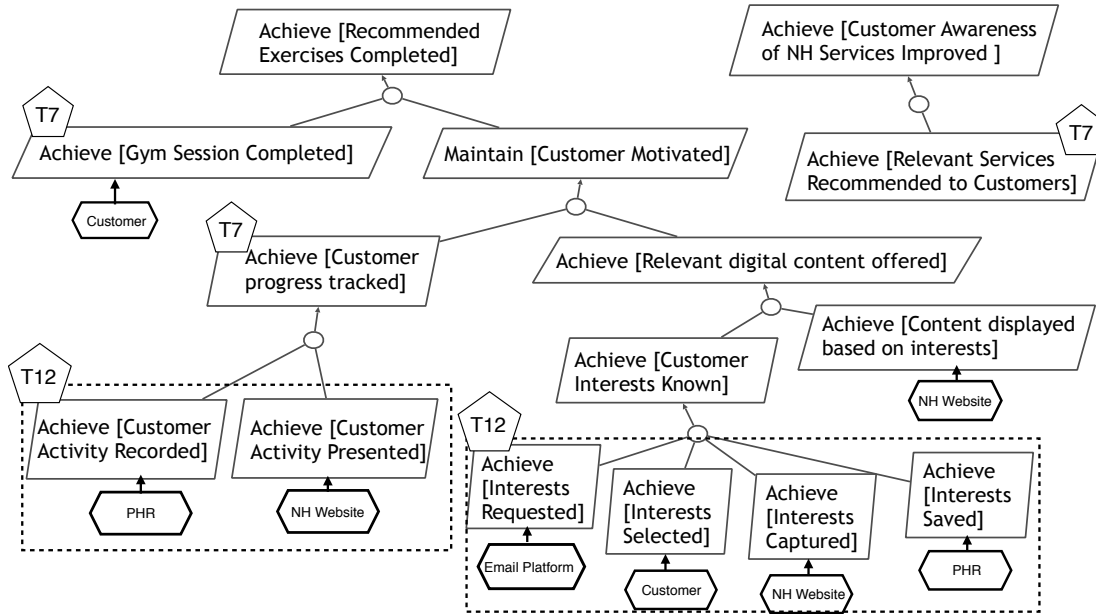
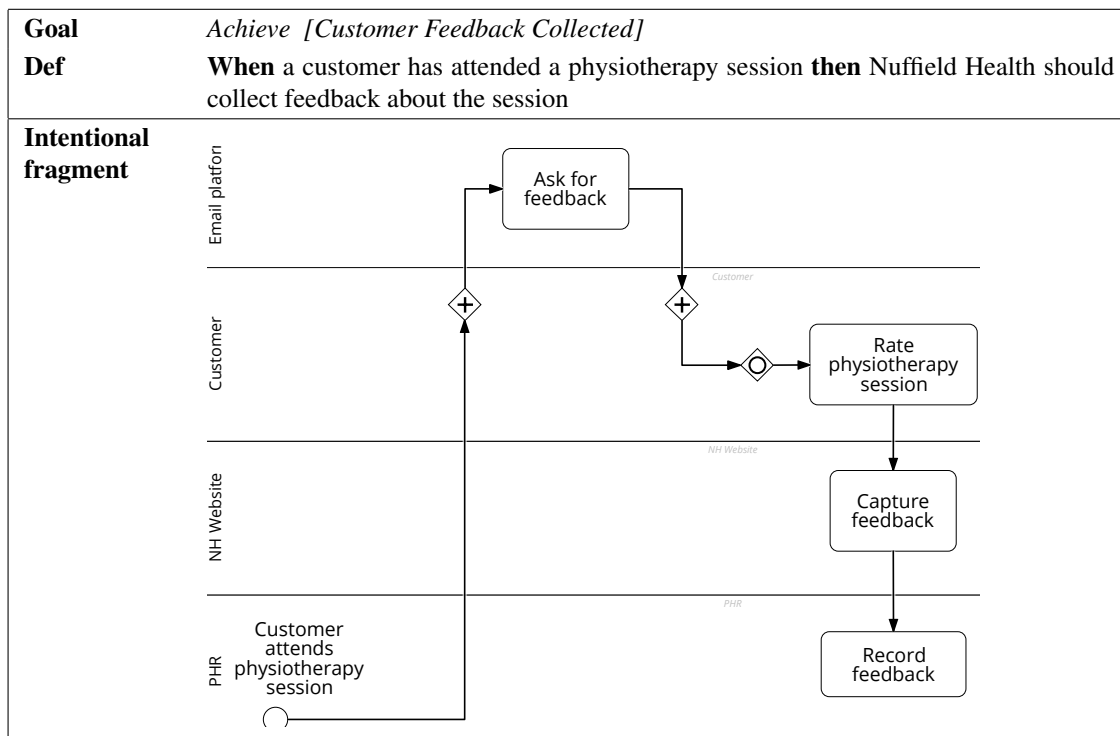


Figure 6.8: Post Physiotherapy - Second Goal Model Refinement

Next, we search for activities that are not yet part of an intentional fragment and check if they contribute to a common goal. This yields a final goal - Achieve [Customer Feedback Collected]. This is a top level goal - while it is useful for Nuffield Health to understand how happy customers are with the services provided, it does not directly impact the current service. The data is valuable for secondary analysis, to understand the relation between an individuals opinion of the service and the likelihood to remain engaged with the gym.



6.5.3 Requirements for the Nuffield PHR Platform

Based on the goal model created, the last step of the planning stage was to understand the specific requirements for the PHR platform. For this, we review the initial goal refinements and we ask:

- are the responsibility assignments correct - can agents satisfy the goals they are responsible for?
- are the goal refinements complete - do the leaf goals ensure the parent goals are satisfied?
- should the goals be further refined - do the leaf goals fall under the responsibility of a single agent?
- are there alternative goal refinements?

First we analyse the part the goal model responsible for goal setting. As mentioned before, a core functionality expected of the system was to allow customers to see the progress they make towards their tailored fitness goals. We give in Figure 6.9 a revised goal refinement. We have operated several changes, following the review process (based on the above questions). First, we note that the PHR system can not directly record the results and the goals agreed during the Health MOT. Following the KAOS unmonitorability-driven refinement pattern we split the responsibilities of the agents. The Health MOT system is the one controlling these values - this is the system used during the health assessments by the medical professionals. The Nuffield Health PHR may only monitor these values - it will read them from the health MOT system and save a copy in a local database. The values may not be changed or overwritten. The second change to the goal model is that we complete the refinement for the goal Achieve [Fitness Goals Defined if Health MOT Not Done]. The Nuffield Health PHR system is responsible to suggest fitness goals, based on data it holds about the customer. This is one area where data analytics and machine learning will help with personalised suggestions.

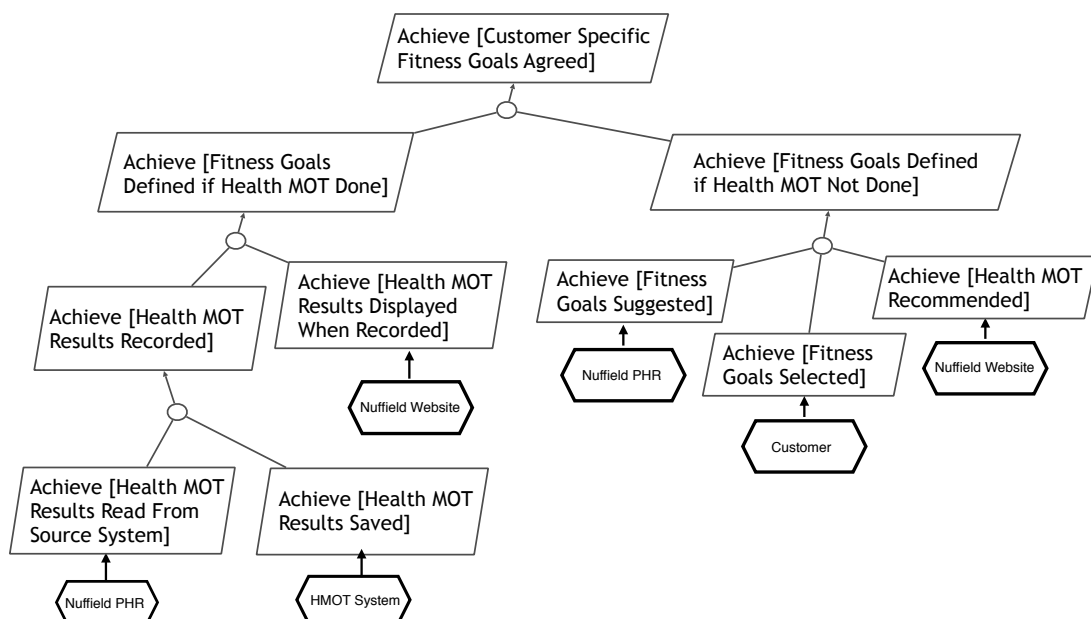


Figure 6.9: Fitness Goals Setting in the System-to-Be

Next, we review the part of the goal model that relates to tracking customer progress towards his goals. The revised goal model is shown in Figure 6.10. The first change we operate, again applying the unmonitorability-driven refinement pattern, is restricting the responsibility of the PHR platform to simply read data from the source systems. The PHR will not be responsible for creating records for customer visits to the gym, or records of someone’s physical activity. Rather, the PHR platform will be responsible to monitor the data available in the source systems and will rely on these systems for data collection. In line with the customer journey presented in Figure 6.1, we also make explicit the responsibility of the PHR to retrieve data from wearable device platforms (besides accessing Nuffield managed data like gym attendance). Another change we operate is assigning to the PHR platform the responsibility to create reports about a customer’s progress towards his goals (initially, this was assigned to the Nuffield website). That is because this type of analysis should be available across a variety of channels, it should not be tied to the website. For example, Nuffield Health could decide to send activity reports via email. As such, to ensure consistency among different digital channels, we assign this responsibility to the PHR.

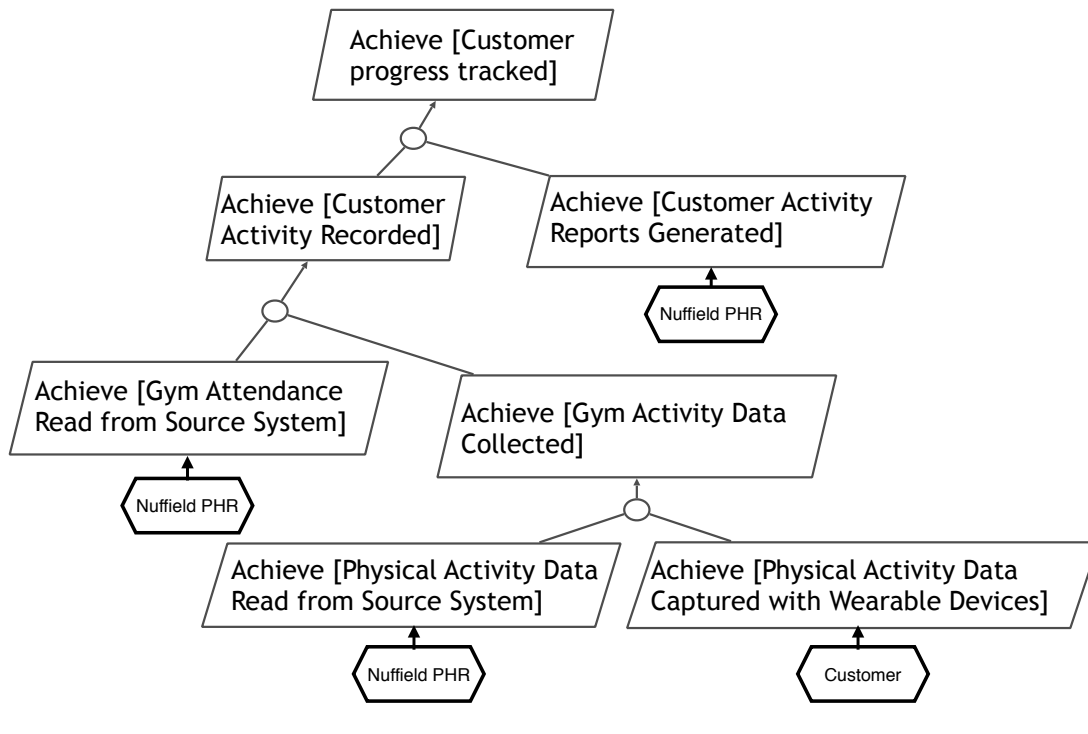


Figure 6.10: Fitness Goals Tracking in the System-to-Be

Finally, we further refine the goal Avoid [Customer Churn]. The refinement is presented in Figure 6.11. It introduces a new software agent - a dedicated system for machine learning and making predictions. While the PHR platform is still responsible to accumulate all the different data points that are predictive of churn, the machine learning system will estimate the probability of each customer to churn. Acting on these predictions - what we called a retention strategy is outside the scope of the PHR. If the strategy is limited to tailored communication based on propensity to churn, it can be delivered as an email campaign.

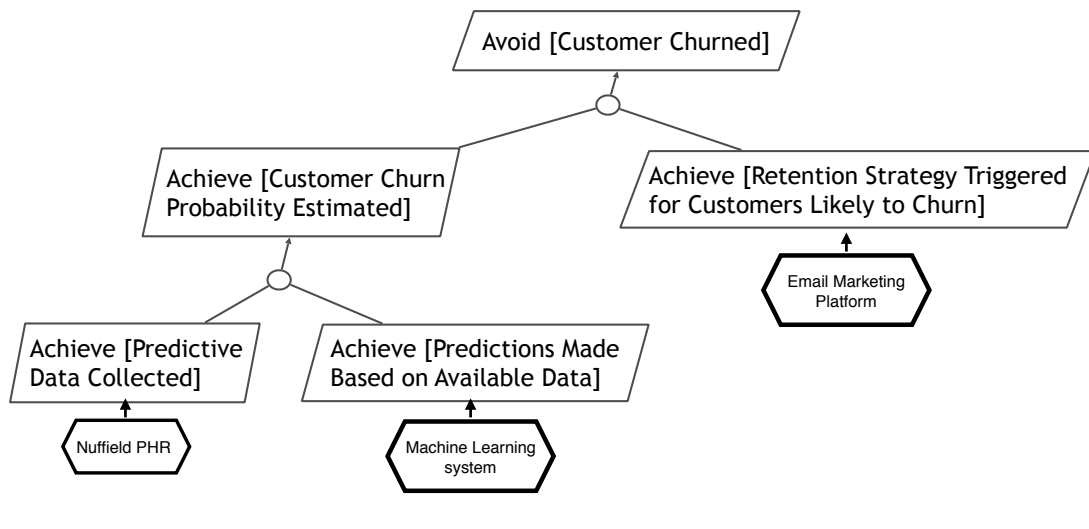


Figure 6.11: Prevent Customer Churn in the System-to-Be

6.6 Taking Action: FHIR FLI - a PHR System for Nuffield Health

In this section we discuss the PHR system that was designed and implemented in Nuffield Health, based on the requirements we identified from the post-physiotherapy journey map. We deem it relevant to briefly present the functionalities of the PHR platform and its architecture, in order to highlight the way in which requirements uncovered during the analysis stage have influenced the final system delivered for the organisation. This is to support our claim that by applying the process based goal elicitation techniques, one can identify meaningful goals and inform the design of the health record platform. The core components of the system have been open sourced under the FHIR FLI project - FHIR for Fitness and Lifestyle data Integration.

The final goal model presented in the previous section constitutes the starting specification for FHIR FLI - an open source platform that supports the exchange, storage and analysis of lifestyle data. The platform includes functionality to: recommend fitness goals to customers based on their past activity levels; track customers' progress towards their chosen fitness goals; and prevent customer churn. There are a number of architectural component, presented in Figure 6.12, required to enable these functionalities.

First, FHIR FLI has a Data Extractor component, responsible to collect data from both internal and external sources. We presented our final goal model to the software architect assigned to the project and by referencing it we were able to clarify the requirement to read data from two Nuffield Health systems: the system responsible for Health Assessment data and the system recording gym members attendance at gym sites. Besides retrieving data from these internal systems, the data extractor also retrieves information from external lifestyle data repositories. While some of the repositories expose a REST API for data retrieval (e.g. the Fitbit ecosystem), others require a mobile app to retrieve data stored locally on a mobile phone (e.g. the Apple HealthKit platform). In the first version of the system, the data extractor component had capabilities to retrieve data via a REST API, while the HealthKit integration was scheduled for a future release.

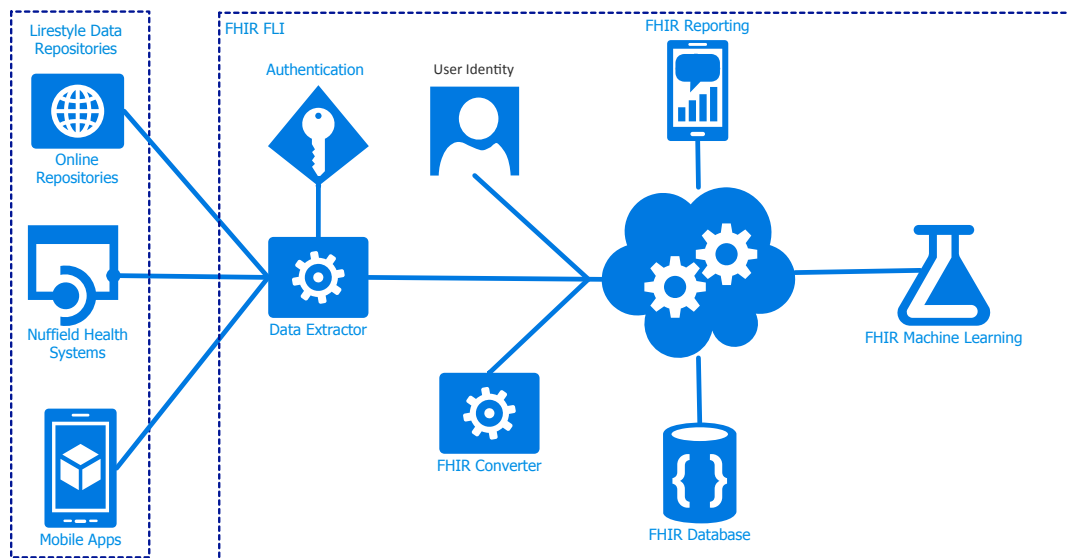


Figure 6.12: FHIR FLI Architecture

The second component in the FHIR FLI architecture is the FHIR converter - it ensures data extracted from different source systems is represented in a consistent format, following the FHIR standard. The decision to convert the data into a standard representation was again motivated by referencing the goals elicited from the customer journey map. The specific choice of the FHIR standard was motivated by additional strategic considerations. As mentioned before, FHIR was already in use by a number of Nuffield Health systems. Relevant for us, the Nuffield health assessment data was already available in FHIR format. Using FHIR as the standard format across the PHR platform presented a number of benefits. It would ensure future interoperability with external systems - for example, the NHS is building FHIR profiles for social care [FHI]. It would also be independent of any proprietary data format, so that reporting or machine learning capabilities will not be tied to specific IT systems (that are subject to change). FHIR also follows modern web development practices - since it offers granular access to data via REST APIs, it accelerates development of client facing applications. However, there is no established FHIR format for saving lifestyle data. As such, FHIR FLI relies on extensions to FHIR Observations - these extensions (called "profiles" in the FHIR terminology) cover data items commonly used in the fitness domain, such as a physical work out. In this context, the FHIR Converter component maps incoming data (in its original format) onto the new FHIR profiles for lifestyle data. Although the KAOS models have not been used to inform the structure of the FHIR profiles (i.e. the data representation), we acknowledge this area as relevant for future work.

The FHIR observations are then saved in a document database. This was the most appropriate technology choice, since FHIR observations are represented as JSON structures.

The other two components of the FHIR FLI platform - for reporting and machine learning - only need to interact with the document database - this modular design has the benefit that a client-facing application that queries and displays lifestyle information will continue to function, even if the format

of the data in the source systems changes. As an example, Nuffield Health may continue reporting on individual gym attendance, even if the system keeping track of member card swipes when entering the gym changes. The addition of a machine learning component in the PHR system is directly attributable to the goal elicitation process. We identified two goals (that could be satisfied through predictive modelling) which have not been considered at the start of the project. First, using machine learning to suggest customers fitness goals, based on their history of physical activity. Second, using machine learning to identify customers at risk of churning (i.e. customers predicted to stop going to the gym).

6.7 Evaluating Action

6.7.1 RQ1: How Successful Was the FHIR FLI Platform

The FHIR FLI platform has been designed based on target user journeys developed in Nuffield Health between 2016 - 2017. A fragment of these journeys, related specifically to the post-physiotherapy experience, has been shown in Figures 6.1 and 6.2. Through the functionalities it offers, FHIR FLI can act as the Nuffield Health PHR platform, thus enabling these journeys. Client-facing applications (e.g. the Nuffield Health website, mobile apps, emails) will provide the interface between the PHR functionalities and the end users. Because of this, we are not evaluating FHIR FLI based on usage indicators (adoption rate, feedback etc.) - these are inexorably linked to the front-end applications of which we did not have any control. Instead, we look for evidence that the project deliverables (i.e. the FHIR FLI components) have influenced current or future Nuffield Health projects, architecture guidelines or digital priorities.

The first evidence of impact is the IT department's decision to embrace the FHIR standard (and the new FHIR observations for lifestyle data developed as part of this project). The FHIR FLI platform has been instrumental in this decision, as it has demonstrated a working system built around this data standard. Moreover, the platform has demonstrated that FHIR is suitable for the types of data Nuffield Health routinely collects (for example, workout sessions in the gyms).

FHIR FLI has been evaluated by the Nuffield IT architects, and the platform was approved for internal use. It is being considered as the short term PHR platform for Nuffield Health, to serve two specific business needs: collecting fitness data from wearable devices and storing data generated through online health questionnaires. As such, it has influenced the overall strategy related to the management and use of lifestyle data within Nuffield by replacing project specific databases with a single repository, spanning across different business cases.

Delivering the FHIR FLI platform has also lead to new projects in the organisation. Specifically, Nuffield Health will build upon the reporting component of FHIR FLI to deliver an online portal for population level reporting. In terms of machine learning, a follow up project investigates customer churn rates and tries to establish if there is any correlation with their gym attendance patterns.

The FHIR FLI platform has been presented as an open source system to store and analyse lifestyle data in the wider healthcare industry. We have established collaborations with the FHIR community and international initiatives that aim to develop personal health record platforms - for example, a national PHR project in Finland [Fin], or NHS Digital in the UK.

We have also formed a community of developers interested in continuing the development of the platform. The open source FHIR FLI components and FHIR profiles for lifestyle data are published on Github.

6.7.2 RQ2: Contributions of Our Techniques

The KAOS requirements engineering method helped clarify the requirements for the PHR platform. As shown in Figure 6.13, in the initial documentation, for each web page, there was an associated capability list and a textual description of what customers and staff can do on the page. However, the visual illustrations are too specific - they present the system from the point of view of a single user. On the other hand, the list of capabilities list is too high level - it shows broad areas of functionality, without going into design details. Finally, the system requirements present only one way of delivering on an implicit goal - for example, 'aggregated benchmark from other similar patients' suggests the need for motivating customers, but does not explore other ways of achieving this.

In comparison, KAOS goal models explain the rationale behind the design of the user interface and the motivations of users interacting with the website. This allows for alternative designs to be explored. The KAOS process models help clarify the responsibilities of each agent, offering a more detailed view of how systems need to interact to achieve the high level functionality expected of them.

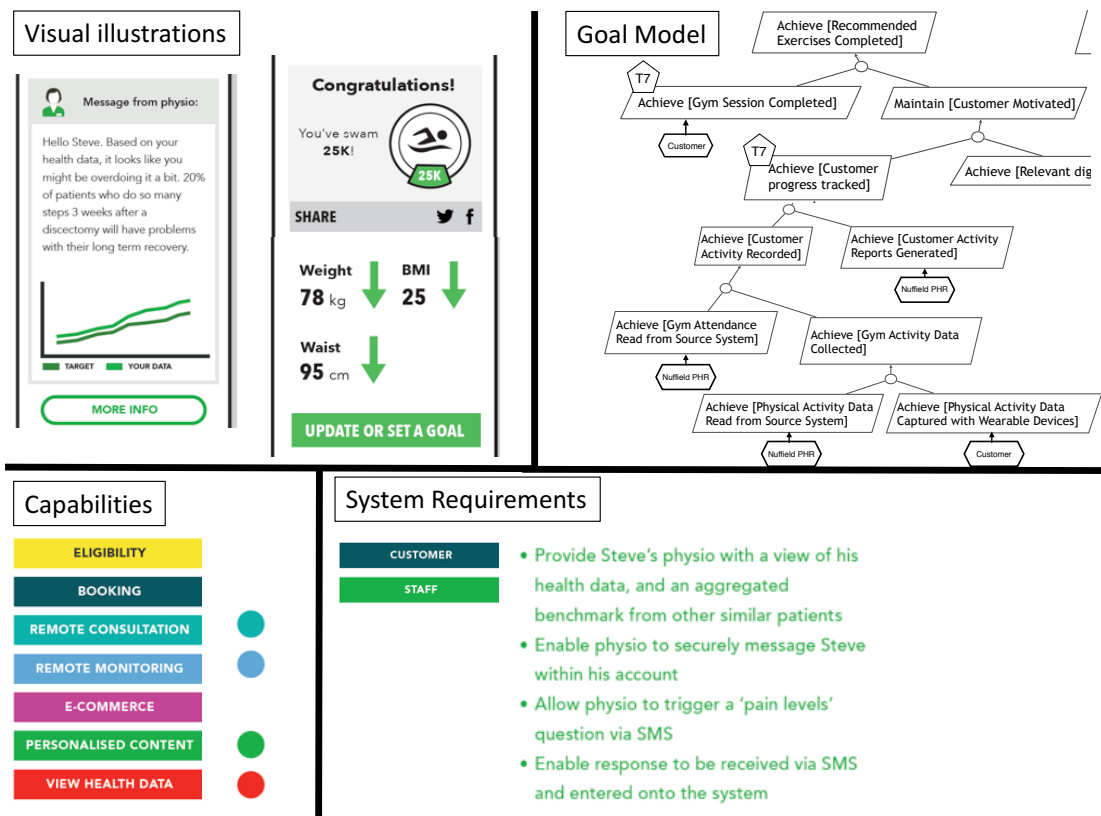


Figure 6.13: Nuffield PHR - Responsibility Assignments

In this context, the KAOS framework helped first of all to delineate areas of functionality associated with the PHR from areas of functionality for the website. By separating requirements specification

from the user interface elements, we have discovered alternative agents who could deliver some of the functionalities. An example of this is the fact that creating visualisations based on the lifestyle data has been assigned to the PHR, so that these assets (graphics) can be reused in emails or mobile apps. Clear responsibility assignments helped us define specific interoperability requirements, by asking what agents monitor and control data. Specifically, we have established the need for the PHR to integrate with the system managing the gym attendance data, something that was not in the scope of the project originally.

Second, the KAOS framework helped identify business goals that were not addressed through current systems, such as preventing churn. This was acknowledged as an important business goal, and Nuffield Health decided to pursue this aspect by adding a machine learning component to the PHR. A separate project investigates the relation between weekly physical activity and the likelihood to churn.

Thirdly, using the goal model, we have identified areas of functionality that were not completely addressed. The example for this is the goal setting functionality for each customer. Although seen as important to motivate people to attend the gym, goal setting was initially envisioned as a responsibility of the human agents: either a health expert during the health MOT or the customer when he visits the website. The goal model highlighted the fact that the software platform lacks the functionality to suggest goals without human intervention. This was included in the responsibilities of the PHR platform.

6.7.3 RQ3: Fit with the Software Delivery Process

To evaluate how well our method fits with the organisation, we discuss the relation between the KAOS framework and the software development process followed by the delivery team.

We have reviewed Behaviour Driven Design in Section 6.4.3. During workshops with the technical team, developers have shown interest in the KAOS method and how it could be integrated with the current practice.

Some similarities between KAOS and BDD are immediately apparent: practitioners can map intentional fragments to BDD feature files. The goal of an intentional fragment corresponds to the high level description of a feature, while the process fragment (the sequence of activities) correspond to the steps in the scenario descriptions.

There are also a number of differences between feature files and intentional fragments. The steps in a scenario depict one specific sequence of events, akin to one execution instance of a process model. To reflect control logic and business rules, a feature file will require additional scenarios, so as to reflect different execution paths, under different circumstances. This means business rules and requirements are more difficult to identify in a large feature file, with many scenarios. Scenarios may also be closely tied to the user interface, as feature files are used for automated testing as well.

An intentional fragment abstracts away from the implementation details, but keeps focus on the requirements, the tasks and the control logic (through gateways).

We propose that the KAOS method is best used during the early stages of analysis. It allows exploration of alternative designs for the system-to-be. Analysts may interrogate the goal or the process model. They could extend the goal model by identifying obstacles or conflicts, and they could introduce necessary changes to the workflow. Once the goal and the process model have evolved and became con-

sistent, the final intentional fragments can be translated to feature files, in the BDD software development process used by Nuffield Health. This is one area that we highlight as relevant for future work.

6.8 Lessons Learned

First, the collaboration with Nuffield Health helped us position our goal elicitation method within the wider set of development practices used in the organisation. We learned that KAOS process models complement both user journey maps and scenarios. Compared to journey maps, decision points in the process model are clearly articulated, through gateways. Also, process models present in more detail the work carried out within the organisation, while journey maps hide this complexity, focusing on the customer perspective. On the other hand, scenarios are very detailed and precise, focusing on one area of functionality - they are written based on clear expectations on the design and behaviour of the software system. Because process models capture all the work carried out within the organisation, which may serve different, possibly conflicting goals, we have successfully used them to elicit goals in the early analysis phase of a project, before design decisions are taken.

Second, during the process modelling work we faced difficulties deriving process models from the customer journey maps. The envisioned customer experience was not linear - at every step in the journey, users have many alternatives: they could jump to a different step, go back to a previous one, repeat a portion of the journey, or drop-off. To be able to represent this type of flow in the KAOS process models, we needed to expand the graphical notation, to include case based gateways and allow for loops.

Third, for the Nuffield Health modelling work we did not use annotations for process activities (i.e. we did not specify pre and post conditions for each activity) because of time limitations. Most of the goal inference techniques presented in this thesis can be used even if the activities lack annotations, and so we were still able to identify relevant goals. Further, we extended our set of heuristics for identifying intentional fragments, incorporating a new structural pattern: activities within a loop could form an Intentional Fragment.

Chapter 7

Conclusion

In this chapter we present a summary of our contributions and discuss future research directions. Section 7.1 reviews each of our four experiments, presenting in brief the rationale and the findings from each. Conversely, Section 7.2 presents possible extensions to our work, based again on our four experiments.

7.1 Summary of Contributions

This thesis has introduced and evaluated a set of requirements engineering techniques for electronic healthcare records, based on process models. The extended KAOS requirements engineering method 1) has support for rigorous process modelling; 2) provides guidance to business analysts during the requirements elicitation process, through a set of goal inference heuristics; and 3) improves the quality of the requirements engineering process for clinical EHR systems and consumer oriented PHR systems.

7.1.1 Relating Goal Oriented Requirements Engineering and Process Modelling

Chapter 3 defines our first contribution: an extension of the KAOS framework that incorporates process modelling. We consider process models to be a new KAOS view, with a graphical layer inspired by BPMN and an execution semantic inspired by the KAOS operation model. As KAOS is a rich and mature framework, Chapter 3 investigates in detail how does the new process view integrate with existing KAOS models and techniques.

Our proposal for a KAOS process view has been informed by our work on the WellbeingUCL project. The project was a collaboration between UCL, Boots and various device manufacturers. We have been involved with designing both the workflow and the supporting IT system - the aim was to develop a platform for collecting wellbeing data from willing members of the public.

The meta-model of the KAOS process view retains core elements of BPMN (activities, gateways, events, sequence flows), but departs from the BPMN standard in certain aspects. Specifically, we treat process events as a specialisation of activities, and we represent message passing as send and receive events. We introduce well-formedness rules to make the KAOS process models more rigorous, and we propose guidelines on how process elements should be labelled. We also introduce activities annotations, which describe the system state before and after an activity is executed. Activities annotations help distinguish between prescriptive and descriptive aspects in a process model. We discuss consistency rules between the graphical representation of a process model and the activities annotations.

Process activities are represented as objects in the object model. Further, each activity has a start and an end operation, consistent with the activity pre and post-conditions. The execution of an activity is thus equivalent to applying its two operations in the system. KAOS agents perform activities in order to satisfy goals. When fully annotated, formal model checking may be used to prove that a process model satisfied a given goal. This amounts to deriving the start and end operation of each activity, and then checking the resulting operational model against the goal model. The process model is also an important source of low level requirements. When the process is prescriptive, it is the responsibility of the agents to ensure each activity is performed in accordance to the prescribed flow and the constraints defined in the activities annotations.

To account for cases where the full execution semantic of the process is not available, we introduce Intentional Fragments as a semi-formal mechanism to relate process activities to the goals they help satisfy. Intentional fragments establish relations between the process view and the goal model, at an early stage of analysis when the two models are not fully developed or entirely consistent. In this sense, intentional fragments are a useful instrument to assist with requirements elicitation from process models - early in the analysis stage of a project, business analysts often face incorrect or incomplete information. With intentional fragments, they can start reasoning about the alignment between the different system views. Each process activity should contribute to a goal, and each goal identified should have activities contributing to it. By identifying cases where this is not the case, business analysts can iteratively improve the models.

7.1.2 Inferring Goal Models from Process Models

Chapter 4 describes our second contribution - a set of heuristics that business analysts can use to derive a goal model starting from a process model. By proposing these heuristics we aim to facilitate adoption of the extended KAOS framework by business analysts. Our approach starts from the process model, which is the more familiar artefact to business analysts, and guides them towards building a correct goal model.

Broadly, the goal inference heuristics we propose fall under four different categories. First, heuristics to identify the process top functional goal. We encourage a top-down refinement process, so that business analysts are not tempted to simply map each process activity to a goal. Second, heuristics to identify additional functional and non-functional goals. Under this category we present structural patterns that indicate candidate intentional fragments and also discuss how activities annotations may be used to discover intentional fragments. Third, heuristics that help business analysts relate goals into a goal model, by analysing the inclusion relationship between intentional fragments at the process level. Fourth, heuristics that guide business analysts to further refine the goal model, until each goal is assignable to one agent.

We demonstrate tool support to automate generation of the process global goals - implementing the first category of heuristics. Using the tool on a collection of process models sourced from public repositories and from UCL business analysts, we discover that with the other categories of heuristics, trying to apply them automatically and indiscriminately generates too many candidate intentional fragments. We

conclude that a better approach is to allow business analysts to selectively apply the heuristics, relying on their domain knowledge.

7.1.3 Electronic Healthcare Record for Bupa

In Chapter 5 we present the first industry case study - the Bupa Virtual Ward. The project had two main objectives. First, developing a new EHR system that would assist clinical staff in providing community based care to COPD patients in the Somerset area. The new system was planned to replace an existing software platform, and the provision of the COPD service was expected to continue throughout the transition. Second, the newly developed EHR system was meant to demonstrate a Virtual Ward model for delivery of care. Specifically, this involved helping patients with chronic diseases better manage their conditions at home, with support provided in the community, so that they avoid acute hospital episodes. We used the extended KAOS framework to assist during the analysis and design stage for the project. Process models were used to capture the workflows of the clinical staff (6 specialist nurses).

Using KAOS, we were able to elicit additional information that was not available in the process models created before our involvement. We could distinguish between an idealised workflow, and the real, day-to-day workflow of the nurses. The actual workflow was a combination of clinical guidelines that had to be followed, arbitrary design decisions that we could alter and workarounds due to the limitations of the initial IT platform. Understanding what can and can not be changed in the workflow allowed us to create a goal model for the system-to-be that both reflected the requirements implicit in the workflows and also allowed for improved workflows where possible. In writing the activities annotations we had to elicit information about the duration of activities and acceptable delays between activities. Since KAOS goals are formally defined using linear temporal logic, the information from the process annotations had been used in the formal definition of the goals. In turn this has allowed us to clearly prove how the new workflows better meet the business goals, using partial goal satisfaction arguments.

As discussed in Section 5.7.1, the EHR system developed for Bupa has been quickly adopted by the nurses. It was used to fully manage the care of the 2,600 enrolled patients, in a live environment. In the first 6 months post deployment no workarounds were observed, and the qualitative feedback collected from the nurses has been positive. The benefit case was clear, with business analysts able to demonstrate measurable improvements in the workflows, aligned with the business goals. This has led to the project being shortlisted for a Health Service Journal industry award recognising specifically improvements in efficiency through technology. The project was also used by Bupa to illustrate best practices in mobile health during external presentations, including a consultation response to the European Commission. The project has accelerated a change in the practice of Bupa business analysts. Process models have become more widely used across Bupa, and the modelling guidelines we introduced have been integrated into internal training materials. The Bupa IT team has also considered how the process analysis techniques we have used during the project can be used within the Bupa Lean methodology.

7.1.4 Personal Health Record for Nuffield Health

In Chapter 6 we present a second industry case study - the Nuffield Personal Health Record. The project aimed to design a PHR platform that could support a number of new customer journeys. The overarching

goal was to provide a connected user experience, where customers could seamlessly transition between the different services offered by Nuffield Health. The PHR platform was expected to integrate with existing Nuffield systems, and the IT architects had to identify interoperability requirements.

We have used the extended KAOS framework to elicit the requirements for the PHR platform by analysing the proposed user journeys. By building a goal model, business analysts were able to identify additional areas of functionality for the PHR platform, that were not considered at the beginning of the project. As such, the PHR system has functionality to integrate with external data sources, in addition to data held internally by Nuffield. Data is represented in a consistent manner, no matter the initial source system. The platform also includes reporting capabilities, created on top of the common information model.

The PHR platform developed for Nuffield has garnered interest from the digital health community, leading to an active open source initiative - FHIR FLI. This is being actively developed into a PHR platform dedicated to managing lifestyle data. It is thus fulfilling a need in the wider industry, where care providers are looking for ways to use lifestyle data to help people manage their risk factors and shift more resources into preventative medicine.

7.2 Future Work

The work presented in this thesis may be extended in a number of ways.

First, future work could investigate techniques for intentional fragments composition. Throughout this thesis we have explored the role of process models for requirements elicitation, in the context of EHR systems. An equally relevant area is designing new business processes, based on information elicited through goal modelling. For example, business analysts could assemble new process models, given a goal model and a library of existing intentional fragments. Using the consistency rules between the KAOS models, business analysts could then analyse the newly defined process. Further, to support the analysis of process models, the concept of Intentional Fragments may be extended. Our current definition relates goals with activities within a single process model. This could be extended in two ways. On one hand, organisations are likely to have a range of business processes, each concerned with a different aspect of the business or the services provided to customers. Therefore, there may be situations where an intentional fragment contains activities belonging to different processes. On the other hand, a business process will likely change over time, or site-specific variations may exist. Therefore, there will be situations where an intentional fragment contains more than one set of activities (each set corresponding to one process variation).

Second, the goal inference tactics may be extended to account for process variations. In the consumer digital health space, the way a consumer interacts with a system cannot be easily characterised as a single workflow - the journey of each user varies slightly. Consider for example a navigation map for a website - to reach a specific page, different users may follow different steps. Analysing where different journeys diverge and the points they converge may provide additional information about the goals of the users.

Third, we will explore how KAOS may be integrated with frameworks for clinical information mod-

elling. For example, OpenEHR [Bea02] is an open platform for electronic healthcare systems which also has incipient support for modelling sequencing of activities (as part of a care plan) and corresponding clinical goals [BB⁺05]. We are specifically interested in using the process and the goal model to inform the clinical information model employed in a system. Collecting and storing data is of prime concern in EHR systems. Consequently, the range of data collected, the format in which data is saved, how often data is collected and how quickly it can be retrieved - all these aspects have to be aligned with the business goals.

Fourth, our current work on the FHIR based Personal Lifestyle Record is under active development. Future work will focus on evaluating the suitability of a FHIR-based system for advanced predictive analytics. This is an area that has not been explored in depth within the FHIR community to date. We are also establishing collaborations with healthcare organisations, so we can further explore the role of lifestyle data in clinical care.

Bibliography

- [ACD⁺03] Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, et al. Business process execution language for web services, 2003.
- [ADW08] Ahmed Awad, Gero Decker, and Mathias Weske. Efficient compliance checking using bpmn-q and temporal logic. In *BPM*, volume 8, pages 326–341. Springer, 2008.
- [AGH⁺03] E. Ammenwerth, S. Gräber, G. Herrmann, T. Bürkle, and J. König. Evaluation of health information systems—problems and challenges. *International journal of medical informatics*, 71(2-3):125–135, 2003.
- [ALMN99] David E Avison, Francis Lau, Michael D Myers, and Peter Axel Nielsen. Action research. *Communications of the ACM*, 42(1):94–97, 1999.
- [AMP94] Annie I Antón, W Michael McCracken, and Colin Potts. Goal decomposition and scenario analysis in business process reengineering. In *International Conference on Advanced Information Systems Engineering*, pages 94–104. Springer, 1994.
- [ANS05] ISO ANSI. Tr 20514 health informatics—electronic health record definition, scope and context, 2005.
- [ASM06] Huzam SF Al-Subaie and Tom SE Maibaum. Evaluating the effectiveness of a goal-oriented requirements engineering method. In *Comparative Evaluation in Requirements Engineering, 2006. CERE'06. Fourth International Workshop on*, pages 8–19. IEEE, 2006.
- [BAM10] Saeed Ahmadi Behnam, Daniel Amyot, and Gunter Mussbacher. Towards a pattern-based framework for goal-driven business process modeling. In *Software Engineering Research, Management and Applications (SERA), 2010 Eighth ACIS International Conference on*, pages 137–145. IEEE, 2010.
- [BB⁺05] Sistine Barretto, Sistine Ann Barretto, et al. Designing guideline-based workflow-integrated electronic health records. 2005.

- [BBR11] Stephan Buchwald, Thomas Bauer, and Manfred Reichert. Bridging the gap between business process models and service composition specifications. *Service Life Cycle Tools and Technologies: Methods, Trends and Advances*, pages 124–153, 2011.
- [Bea02] Thomas Beale. Archetypes: Constraint-based domain models for future-proof information systems. In *OOPSLA 2002 workshop on behavioural semantics*, volume 105, 2002.
- [BEK18] Leonard Peter Binamungu, Suzanne M Embury, and Nikolaos Konstantinou. Maintaining behaviour driven development specifications: Challenges and opportunities. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 175–184. IEEE, 2018.
- [Ber01] Marc Berg. Implementing information systems in health care organizations: myths and challenges. *International journal of medical informatics*, 64(2):143–156, 2001.
- [BMMW15] Manuel Brhel, Hendrik Meth, Alexander Maedche, and Karl Werder. Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology*, 61:163–181, 2015.
- [CAA⁺11] K. Cresswell, M. Ali, A. Avery, N. Barber, T. Cornford, S. Crowe, et al. The long and winding road: An independent evaluation of the implementation and adoption of the national health service care records service (nhs crs) in secondary care in england, 2011.
- [CAH⁺08] B. Chen, G.S. Avrunin, E.A. Henneman, L.A. Clarke, L.J. Osterweil, and P.L. Henne-man. Analyzing medical processes. In *Proceedings of the 30th international conference on Software engineering*, pages 623–632. ACM, 2008.
- [Car13] Evellin Cristine Souza Cardoso. Towards a methodology for goal-oriented enterprise management. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2013 17th IEEE International*, pages 94–103. IEEE, 2013.
- [CB14] David Coghlan and Teresa Brannick. *Doing action research in your own organization*. Sage, 2014.
- [CCML⁺12] Mario Cortes-Cornax, Alexandru Matei, Emmanuel Letier, Sophie Dupuy-Chessa, and Dominique Rieu. Intentional fragments: bridging the gap between organizational and intentional levels in business processes. In *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*, pages 110–127. Springer, 2012.
- [CLM⁺00] A.G. Cass, AS Lerner, E.K. McCall, L.J. Osterweil, S.M. Sutton Jr, and A. Wise. Littlejill/juliette: a process definition language and interpreter. In *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, pages 754–757. IEEE, 2000.

- [CNYM12] Lawrence Chung, Brian A Nixon, Eric Yu, and John Mylopoulos. *Non-functional requirements in software engineering*, volume 5. Springer Science & Business Media, 2012.
- [Com12] European Commission. ehealth action plan 2012–2020. innovative healthcare for the 21st century. *Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions. Brussels, 6.12. 2012*, 2012.
- [CW92] MJ Craft and JA Willadsen. Interventions related to family. *The Nursing Clinics of North America*, 27(2):517–540, 1992.
- [DAB⁺06] Robert H Dolin, Liora Alschuler, Sandy Boyer, Calvin Beebe, Fred M Behlen, Paul V Biron, et al. HI7 clinical document architecture, release 2. *Journal of the American Medical Informatics Association*, 13(1):30–39, 2006.
- [DDDGB08] G. Decker, R. Dijkman, M. Dumas, and L. García-Bañuelos. Transforming bpmn diagrams into yawl nets. *Business Process Management*, pages 386–389, 2008.
- [DDO08] R.M. Dijkman, M. Dumas, and C. Ouyang. Semantics and analysis of business process models in bpmn. *Information and Software Technology*, 50(12):1281–1294, 2008.
- [DLRVL09] C. Damas, B. Lambeau, F. Roucoux, and A. Van Lamsweerde. Analyzing critical process models through behavior model synthesis. In *Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on*, pages 441–451. IEEE, 2009.
- [DIVGD07] Jose Luis De la Vara González and J Sanchez Diaz. Business process-driven requirements engineering: a goal-based approach. In *Proceedings of the 8th Workshop on Business Process Modeling*, 2007.
- [DLVL06] Christophe Damas, Bernard Lambeau, and Axel Van Lamsweerde. Scenarios, goals, and state machines: a win-win partnership for model synthesis. In *Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 197–207. ACM, 2006.
- [dIVSP08] J. de la Vara, J. Sánchez, and Ó. Pastor. Business process modelling and purpose analysis for requirements analysis of information systems. In *Advanced Information Systems Engineering*, pages 213–227. Springer, 2008.
- [dIVSP13] Jose Luis de la Vara, Juan Sánchez, and Oscar Pastor. On the use of goal models and business process models for elicitation of system requirements. pages 168–183, 2013.
- [DP09] Ken Decreus and Geert Poels. Mapping semantically enriched formal tropos to business process models. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 371–376. ACM, 2009.

- [DVL96] R. Darimont and A. Van Lamsweerde. Formal refinement patterns for goal-driven requirements elaboration. *ACM SIGSOFT Software Engineering Notes*, 21(6):179–190, 1996.
- [DVL93] A. Dardenne, A. Van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of computer programming*, 20(1-2):3–50, 1993.
- [EEM95] Khaled El Emam and Nazim H Madhavji. A field study of requirements engineering practices in information systems development. In *Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on*, pages 68–80. IEEE, 1995.
- [Fen08] Wendy Fenton. Introducing a post-fall assessment algorithm into a community rehabilitation hospital for older adults. *Nursing older people*, 20(10):36–39, 2008.
- [FGZ15] Samuel A Fricker, Rainer Grau, and Adrian Zwingli. Requirements engineering: best practice. In *Requirements Engineering for Digital Health*, pages 25–46. Springer, 2015.
- [FHI] Social care domain message specification. <http://data.developer.nhs.uk/fhir/sc-fhir/sc-fhir-1-0-RC4/Chapter.1.About/index.html>. Accessed: 2017-09-12.
- [Fin] Fhir implementation guides. <http://www.kanta.fi/en/web/ammattilaisille/fhir-oppaat>. Accessed: 2017-09-12.
- [FLM⁺04] Ariel Fuxman, Lin Liu, John Mylopoulos, Marco Pistore, Marco Roveri, and Paolo Traverso. Specifying and analyzing early requirements in tropos. *Requirements Engineering*, 9(2):132–150, 2004.
- [FMP03] Eduardo Fernández-Medina and Mario Piattini. Designing secure databases for ols. In *Database and Expert Systems Applications*, pages 886–895. Springer, 2003.
- [FSR10] Jennifer Ferreira, Helen Sharp, and Hugh Robinson. Values and assumptions shaping agile development and user experience design in practice. In *International Conference on Agile Software Development*, pages 178–183. Springer, 2010.
- [GR09] Albert Gatt and Ehud Reiter. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, ENLG '09, pages 90–93, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [Har16] P Harmon. A bptrends report: The state of business process management. 2016.
- [Hee06] R. Heeks. Health information systems: Failure, success and improvisation. *International journal of medical informatics*, 75(2):125–137, 2006.

- [HK10] Richard J Holden and Ben-Tzion Karsh. The technology acceptance model: its past and its future in health care. *Journal of biomedical informatics*, 43(1):159–172, 2010.
- [HSJ] Health service journal awards 2014 - shortlist. <https://www.hsj.co.uk/home/hsj-awards-2014-shortlist-revealed/5074739.article>. Accessed: 2018-06-06.
- [HSN08a] Kristiina Häyrynen, Kaija Saranto, and Pirkko Nykänen. Definition, structure, content, use and impacts of electronic health records: a review of the research literature. *International journal of medical informatics*, 77(5):291–304, 2008.
- [HSN08b] Kristiina Hyrinen, Kaija Saranto, and Pirkko Nyknen. Definition, structure, content, use and impacts of electronic health records: A review of the research literature. *International Journal of Medical Informatics*, 77(5):291 – 304, 2008.
- [IGRR09] Marta Indulska, Peter Green, Jan Recker, and Michael Rosemann. Business process modeling: Perceived benefits. pages 458–471, 2009.
- [IRRG09] M. Indulska, J. Recker, M. Rosemann, and P. Green. Business process modeling: Current issues and future challenges. In *Advanced Information Systems Engineering*, pages 501–514. Springer, 2009.
- [ISO13] ISO. Information technology object management group business process model and notation, 2013.
- [KG06] G. Koliadis and A. Ghose. Relating business process models to goal-oriented requirements models in kaos. In *Advances in Knowledge Acquisition and Management*, pages 25–39. Springer, 2006.
- [KK97] Peter Kueng and Peter Kawalek. Goal-based business process models: creation and evaluation. *Business Process Management Journal*, 3(1):17–38, 1997.
- [KKW10] Elizabeth Koshy, Valsa Koshy, and Heather Waterman. *Action research in healthcare*. Sage, 2010.
- [KL99] Vagelio Kavakli and Pericles Loucopoulos. Goal-driven business process analysis application in electricity deregulation. *Information Systems*, 24(3):187–207, 1999.
- [KLWW11] Matthias Kunze, Alexander Luebbe, Matthias Weidlich, and Mathias Weske. Towards understanding process modeling—the case of the bpm academic initiative. In *International Workshop on Business Process Modeling Notation*, pages 44–58. Springer, 2011.
- [KVH⁺06] Henk Koning, John PS Verver, Jaap Heuvel, Soren Bisgaard, and Ronald JMM Does. Lean six sigma in healthcare. *Journal for Healthcare Quality*, 28(2):4–11, 2006.

- [KVK⁺04] Marjo Kauppinen, Matti Vartiainen, Jyrki Kontio, Sari Kujala, and Reijo Sulonen. Implementing requirements engineering processes throughout organizations: success factors and challenges. *Information and Software Technology*, 46(14):937–953, 2004.
- [KWTK08] Ross Koppel, Tosha Wetterneck, Joel Leon Telles, and Ben-Tzion Karsh. Workarounds to barcode medication administration systems: their occurrences, causes, and threats to patient safety. *Journal of the American Medical Informatics Association*, 15(4):408–423, 2008.
- [Lau99] Francis Lau. Toward a framework for action research in information systems studies. *Information Technology & People*, 12(2):148–176, 1999.
- [LD09] Dave A Ludwick and John Doucette. Adopting electronic medical records in primary care: lessons learned from health information systems implementation experience in seven countries. *International journal of medical informatics*, 78(1):22–31, 2009.
- [LEF01] Hallvard Lærum, Gunnar Ellingsen, and Arild Faxvaag. Doctors’ use of electronic medical records systems in hospitals: cross sectional survey. *Bmj*, 323(7325):1344–1348, 2001.
- [LKMU08] E. Letier, J. Kramer, J. Magee, and S. Uchitel. Deriving event-based transition systems from goal-oriented requirements models. *Automated Software Engineering*, 15(2):175–206, 2008.
- [LR05] Liette Lapointe and Suzanne Rivard. A multilevel model of resistance to information technology implementation. *Mis Quarterly*, pages 461–491, 2005.
- [LR06] Liette Lapointe and Suzanne Rivard. Getting physicians to accept new information technology: insights from case studies. *Canadian Medical Association Journal*, 174(11):1573–1578, 2006.
- [LR07] R. Lenz and M. Reichert. It support for healthcare processes—premises, challenges, perspectives. *Data & Knowledge Engineering*, 61(1):39–58, 2007.
- [LR12] Matthias Lohrmann and Manfred Reichert. Modeling business objectives for business process management. In *S-BPM ONE—Scientific Research*, pages 106–126. Springer, 2012.
- [Lup13] Deborah Lupton. The digitally engaged patient: Self-monitoring and self-care in the digital health era. *Social Theory & Health*, 11(3):256–270, 2013.
- [LVL02] Emmanuel Letier and Axel Van Lamsweerde. Deriving operational software specifications from system goals. *ACM SIGSOFT Software Engineering Notes*, 27(6):119–128, 2002.

- [LVL04] E. Letier and A. Van Lamsweerde. Reasoning about partial goal satisfaction for requirements and design engineering. In *ACM SIGSOFT Software Engineering Notes*, volume 29, pages 53–62. ACM, 2004.
- [LVW⁺13] Geraint Lewis, Rhema Vaithianathan, Lorraine Wright, Mary R Brice, Paul Lovell, Seth Rankin, and Martin Bardsley. Integrating care for high-risk patients in england using the virtual ward model: lessons in the process of care integration from three case sites. *International journal of integrated care*, 13, 2013.
- [LYM07] A. Lapouchnian, Y. Yu, and J. Mylopoulos. Requirements-driven design and configuration management of business processes. In *Business Process Management*, pages 246–261. Springer, 2007.
- [Mey00] Julienne Meyer. Qualitative research in health care: Using qualitative methods in health related action research. *BMJ: British Medical Journal*, 320(7228):178, 2000.
- [MK08] I. Markov and M. Kowalkiewicz. Linking business goals to process models in semantic business process modeling. In *Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE*, pages 332–338. IEEE, 2008.
- [MKM⁺16] Joshua C Mandel, David A Kreda, Kenneth D Mandl, Isaac S Kohane, and Rachel B Ramoni. Smart on fhir: a standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*, 23(5):899–908, 2016.
- [MNVDA07] Jan Mendling, Gustaf Neumann, and Wil Van Der Aalst. Understanding the occurrence of errors in process models based on metrics. In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, pages 113–130. Springer, 2007.
- [MR05] Neil Maiden and Suzanne Robertson. Integrating creativity into requirements processes: Experiences with an air traffic management system. In *13th IEEE International Conference on Requirements Engineering (RE'05)*, pages 105–114. IEEE, 2005.
- [MRR10] Jan Mendling, Jan Recker, and Hajo A Reijers. On the usage of labels and icons in business process modeling. *International Journal of Information System Modeling and Design (IJISMD)*, 1(2):40–58, 2010.
- [MRvdA10] Jan Mendling, Hajo A Reijers, and Wil MP van der Aalst. Seven process modeling guidelines (7pmg). *Information and Software Technology*, 52(2):127–136, 2010.
- [MvSW11] Susan Michie, Maartje M van Stralen, and Robert West. The behaviour change wheel: a new method for characterising and designing behaviour change interventions. *Implementation science*, 6(1):42, 2011.

- [ODTHVdA06] Chun Ouyang, Marlon Dumas, Arthur HM Ter Hofstede, and Wil MP Van der Aalst. From bpmn process models to bpel web services. In *Web Services, 2006. ICWS'06. International Conference on*, pages 285–292. IEEE, 2006.
- [OMG11] OMG. Business process model and notation (bpmn), 2011.
- [PDRS13] Geert Poels, Ken Decreus, Ben Roelens, and Monique Snoeck. Investigating goal-oriented requirements engineering for business processes. *Journal of Database Management (JDM)*, 24(2):35–71, 2013.
- [PPR02] Macario Polo, Mario Piattini, and Francisco Ruiz. Using a qualitative research method for building a software maintenance methodology. *Software: Practice and Experience*, 32(13):1239–1260, 2002.
- [PQZ08] D. Prandi, P. Quaglia, and N. Zannone. Formal analysis of bpmn via a translation into cows. In *Coordination Models and Languages*, pages 249–263. Springer, 2008.
- [RIRG10] Jan Recker, Marta Indulska, Michael Rosemann, and Peter Green. The ontological deficiencies of process modeling in practice. 2010.
- [RM06] Jan C Recker and Jan Mendling. On the translation between bpmn and bpel: Conceptual mismatch between process modeling languages. In *The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium*, pages 521–532. Namur University Press, 2006.
- [RP07] Colette Rolland and Naveen Prakash. On the adequate modeling of business process families. *BPMD07*, 2007.
- [SDJ07] Dag IK Sjoberg, Tore Dyba, and Magne Jorgensen. The future of empirical methods in software engineering research. In *2007 Future of Software Engineering*, pages 358–378. IEEE Computer Society, 2007.
- [SDJSD00] Eleanor M Suderman, Judith V Deatrach, Leslie S Johnson, and Doris M Sawatzky-Dickson. Action research sets the stage to improve discharge preparation. *Pediatric nursing*, 26(6):571, 2000.
- [SHK⁺] P Schloeffel, S Heard, D Kalra, D Lloyd, and T Beale. Architecture overview.
- [Sig] Signavio user guide. http://academic.signavio.com/userguide/en/modeling_and_notations/bpmn/creating_views.html. Accessed: 2016-09-3.
- [Sil09] B. Silver. *BPMN method and style*, volume 2. Cody-Cassidy Press, 2009.
- [SM01] Alec Sharp and Patrick McDermott. *Workflow Modeling: Tools for Process Improvement and Application Development*. 2001.

- [SP85] Abraham B Shani and William A Pasmore. Organization inquiry: Towards a new model of the action research process. *Contemporary Organization development: Current Thinking and Applications*, Scott, Foresman, Glenview, IL, pages 438–448, 1985.
- [ST09] Paulo Sergio Medeiros dos Santos and Guilherme Horta Travassos. Action research use in software engineering: An initial survey. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 414–417. IEEE Computer Society, 2009.
- [STGK] L Sonola, V Thiel, N Goodwin, and DL Kodner. South devon and torbay: proactive case management using the community virtual ward and the devon predictive model. the kings fund; 2013.
- [SW11] Carlos Solis and Xiaofeng Wang. A study of the characteristics of behaviour driven development. In *Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on*, pages 383–387. IEEE, 2011.
- [Tas] Openehr task planning information model. http://www.openehr.org/releases/RM/latest/docs/task_planning/task_planning.html. Accessed: 2017-04-04.
- [Tol] Tolven data flow. http://home.tolven.org/?page_id=72. Accessed: 2017-05-04.
- [UKM03] Sebastian Uchitel, Jeff Kramer, and Jeff Magee. Synthesis of behavioral models from scenarios. *IEEE Transactions on Software Engineering*, 29(2):99–115, 2003.
- [UL13] Azmat Ullah and Richard Lai. Requirements engineering and business/it alignment: Lessons learned. 2013.
- [VAE⁺04] Meera Viswanathan, Alice Ammerman, Eugenia Eng, Gerald Garlehner, Kathleen N Lohr, Derek Griffith, Scott Rhodes, Carmen Samuel-Hodge, Siobhan Maty, Linda Lux, et al. Community-based participatory research: Assessing the evidence: Summary. 2004.
- [VdA99] Wil MP Van der Aalst. Formalization and verification of event-driven process chains. *Information and Software technology*, 41(10):639–650, 1999.
- [VL01] A. Van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, pages 249–262. IEEE, 2001.
- [VL09] Axel Van Lamsweerde. *Requirements engineering: From system goals to UML models to software*, volume 10. Chichester, UK: John Wiley & Sons, 2009.

- [VLL98] A. Van Lamsweerde and E. Letier. Integrating obstacles in goal-driven requirements engineering. In *Software Engineering, 1998. Proceedings of the 1998 International Conference on*, pages 53–62. IEEE, 1998.
- [VLW98] Axel Van Lamsweerde and Laurent Willemet. Inferring declarative requirements specifications from operational scenarios. *IEEE Transactions on Software Engineering*, 24(12):1089–1114, 1998.
- [WG08] P. Wong and J. Gibbons. A process semantics for bpmn. *Formal Methods and Software Engineering*, pages 355–374, 2008.
- [WH12] Matt Wynne and Aslak Helleoy. *The cucumber book: behaviour-driven development for testers and developers*. Pragmatic Bookshelf, 2012.
- [Whi04] S.A. White. Introduction to bpmn. *IBM Cooperation*, pages 2008–029, 2004.
- [Whi05] Stephen White. Using bpmn to model a bpel process. *BPTrends*, 3(3):1–18, 2005.
- [YM96] E.S.K. Yu and J. Mylopoulos. Using goals, rules and methods to support reasoning in business process reengineering. *Intelligent Systems in Accounting, Finance and Management*, 5(1):1–13, 1996.
- [ZM05] Jörg Ziemann and Jan Mendling. Epc-based modelling of bpel processes: a pragmatic transformation approach. In *Proceedings of MITIP 2005, Italy*, 2005.