Developing an educational tool to promote evidence-based treatment in health care

A pilot study

Ben-Richard Sletten Ebbesvik

Research proposal for master's thesis in Software Engineering at

Department of Computing, Mathematics and Physics,
Bergen University College

Department of Informatics,
University of Bergen

April 2019

Western Norway
University of
Applied Sciences

Contents

Chapter 1

Introduction

1.1 Motivation

1.2 Research questions

- Based on clinical guidelines, can we make a data structure which is easy to implement in the system, as well as adaptable?
- How to use such a model for generating and testing case based multiple choice questions and answer elements?
- How can we model the work-flow of a clinical encounter, a patient at a given point in the clinical encounter, and a student at the current point in his learning process. How to represent these?

1.3 Structure of the thesis

Chapter 2

Background

2.1 Clinical Practice Guidelines

For a clinician, Fervers, Carretier, and Bataillard (2010) claims that increased medical knowledge is associated with an exponential growth of scientific data and published material. It is impossible to keep up, as well as integrating all the new information into daily practice to give patients the best possible care. Masic, Miokovic, and Muhamedagic (2008) gives an example where a general practitioner should read 19 articles per day to keep up with the new medical information, while only having time to read one hour per week. The clinician will spend more time reading articles than treating patients. This problem is known as academic isolation.

Evidence Based Medicine (EBM) suggests that instead of routinely reading dozens of articles, the clinicians should target their reading to specific patient problems. Developing clinical questions and then searching for the answer (problem based approach), may be a more productive way to keep up with the new medical knowledge (Masic, Miokovic, and Muhamedagic 2008). The EBM definition further puts an emphasize on integrating the best evidence in decision making with the clinicians expertise and the patients values and expectations (Masic, Miokovic, and Muhamedagic 2008).

The concept of EBM is about transferring knowledge from clinical research into clinical practice, and Clinical Practice Guidelines (CPG) can play an instrumental role in this process (Fervers, Carretier, and Bataillard 2010).

The Institute of Medicine (IOM) has given the following definition of clinical practice guidelines: "CPGs are statements that include recommendations intended to optimize patient care. These statements are informed by a systematic review of evidence and an assessment of the benefits and costs of alternative care options" ((US) et al. 2011)

The definition given by IOM covers the goals in EBM, and also takes the cost into account. In fact, Clayton and Hripcsak (1995) have shown that in some situations good use of appropriate guidelines and protocols can reduce as much as 25% of the cost of healthcare.

Even though the CPGs have proven to improve the quality of health care while reducing practice variability and the cost of patient care (De Clercq, Kaiser, and Hasman 2008), it is well recognized that CPGs have had a limited effect on changing the clinicians practice methods. Cabana et al. (1999) lists the following reasons:

- Lack of awareness.
- Lack of familiarity.
- Lack of agreement with the content.
- Lack of self-efficacy.
- Lack of outcome expectancy.
- Inertia of previous practice.
- External Barriers; the guidelines are not easy to use, not convenient, cumbersome and confusing.

One example of external barrier is the Guidelines for the Diagnosis and Management of Asthma (National Heart Lung and Blood Institute and U.S. Department of Health & Human Services 2007), which consists of 440 pages. Such a large journal is not convenient to use at the point of care. According to Shortliffe (1998), CPGs in monographs and journal articles tend to sit on book shelves at the time their knowledge could prove the most valuable to the clinicians.

According to Woolf et al. (1999), clinicians sometimes have good reasons to disagree with some of the content of a guideline. Woolf et al. (1999) points out three reasons:

1. The scientific evidence of the recommendation can be lacking, misleading or misinterpreted.
2. The recommendations may be influenced by the authors. What the authors believe, may be inferior to other options, ineffective or harmful.

3. As the guideline may be written to control cost, serve societal needs or protect special interest, the recommendations may be suboptimal for the patient.

There exists grading systems which grade the quality of evidence and strength of recommendations. GRADE is such a grading system (Guyatt et al. 2008).

- Medical knowledge increases. Hard to keep track
- Guidelines is a summary of the available evidence of the medical conditions and provide management and recommendations
- A well-developed guideline reduces variations in care, improves diagnostic accuracy, promotes effective therapy and discourages ineffective therapies all which contribute to improved quality of care (citation)
- The CPGs are not used enough
- Dissemination and implementation
- Large volume of excisting guidelines. Difficult to use at the point of care
- Dissemination
- Different practice even in the same country

2.2 Serious games

2.3 Asthma

2.3.1 Challenges

2.4 Related work

2.5 Summary

Chapter 3

Method

3.1 Design study

3.2 Iterations

3.2.1 Testing mobile app technology, conceptual model

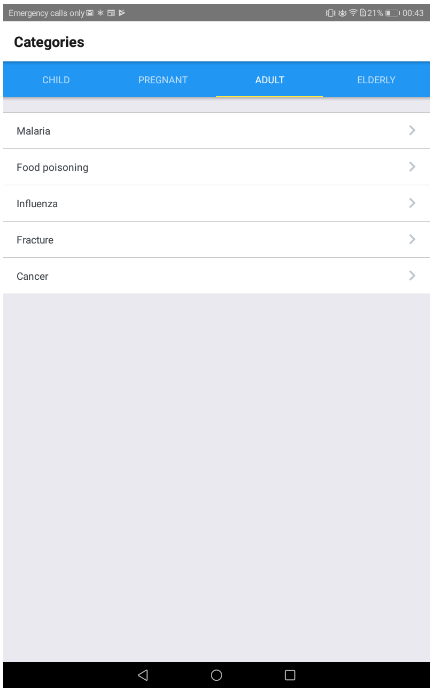
The goal of this iteration was to make a prototype using the proposed technology. The prototype will tell us if it is sensible to use the technology for the project, as well as having a conceptual model to use when discussing initial ideas around the aim of the project, functionality, as well as interaction design.

The technology to be tested was React Native. The reason for this to be the first choice, was to support multiple platforms and not having to rewrite the application for every platform. The same application can be used for Android and iPhone with little modification to the code. React Native is build upon the React front end web development framework, which means some code can be reused for web as well. But all the the views need to be modified, as they use specific React Native components for mobile units. React and React Native are JavaScript frameworks.

We have also used the React web development framework in previous projects. It s one of the more popular and mature front end web development frameworks in the market.

The prototype itself and the development process showed that the technology could be used for displaying and organize CPGs. Lessons learned about content flow, databases, app navigation, displaying information and dialogues. JavaScript in the view gives a lot of flexibility compared to just tags.

Figure 3.1: A very simple conceptual model to test one of the proposed technologies, as well as acting as a starting point for discussions



The interface was evaluated with the supervisors, using cognitive walk-throughs. It also worked as a conceptual model, using the prototype as a base for discussing ideas. Both for discussing the purpose of the project, what the user should be able to do, how to organize content and what functionality to add.

3.2.2 Studying similar products

The purpose of the iteration is to see what exists in the market. What have others done. What can we improve, where can we add value to both the medical community and to computer science.

The results from this iteration is presented in appendix 6.1, and was done together with a fellow master of science student in software engineering. As a conclusion, none of these application have a data model which can represent CPGs, nor a patient in a clinical encounter. The representations of CPGs are mostly text based, flow charts or flow charts which expand when you click on decision vertices. LIFE: Neonatal Resuscitation Training is a pretty advanced 3D game, but there is no data model representing the content in the quizzes and tasks.

3.2.3 Technology used to represent CPGs

In this iteration we evaluated technologies which could be used to model CPGs.

- **GLIF** and **PROforma** was compared during a literature study as a semester assignment in health informatics. We will here give a brief summary of GLIF.

  GLIF or the Guideline Interchange Format, was developed by the InterMed Collaboratory. The intention was to make a guideline representation language, which can be viewed with a various of software application as well as adapting them and making them valid for different local uses. The representation should be precise, ambiguous, readable by humans and interpretable by computers, adaptable to different clinical information standards and facilitating guideline sharing (Peleg et al. 2000).

  To make the guidelines readable for humans, interpretable by computers and adaptable by different institutions, GLIF defines the guidelines at three different abstraction levels: conceptual, computable and implementable levels (De Clercq, Kaiser, and Hasman 2008)

  The conceptual level is the highest abstraction level. It consists of flow-charts which can be viewed by humans, using guideline viewing programs. At this level the guidelines can not be used for computations in decision support (Peleg et al. 2000).

  At the computable level expressions, patient data elements, clinical actions and guideline flow are specified at this level. The guidelines can also be verified for logical consistency and completeness (Peleg et al. 2000).

  The implementable level contains the information to incorporate the guidelines into the particular institutions knowledge or information system such as EPR (Peleg et al. 2000).

  In figure 3.2 we present a model of the paediatric possible asthma guideline Republic of Kenya 2016 at the conceptual level. We will now explain the guideline steps the UML-model in figure 3.2 consists of.

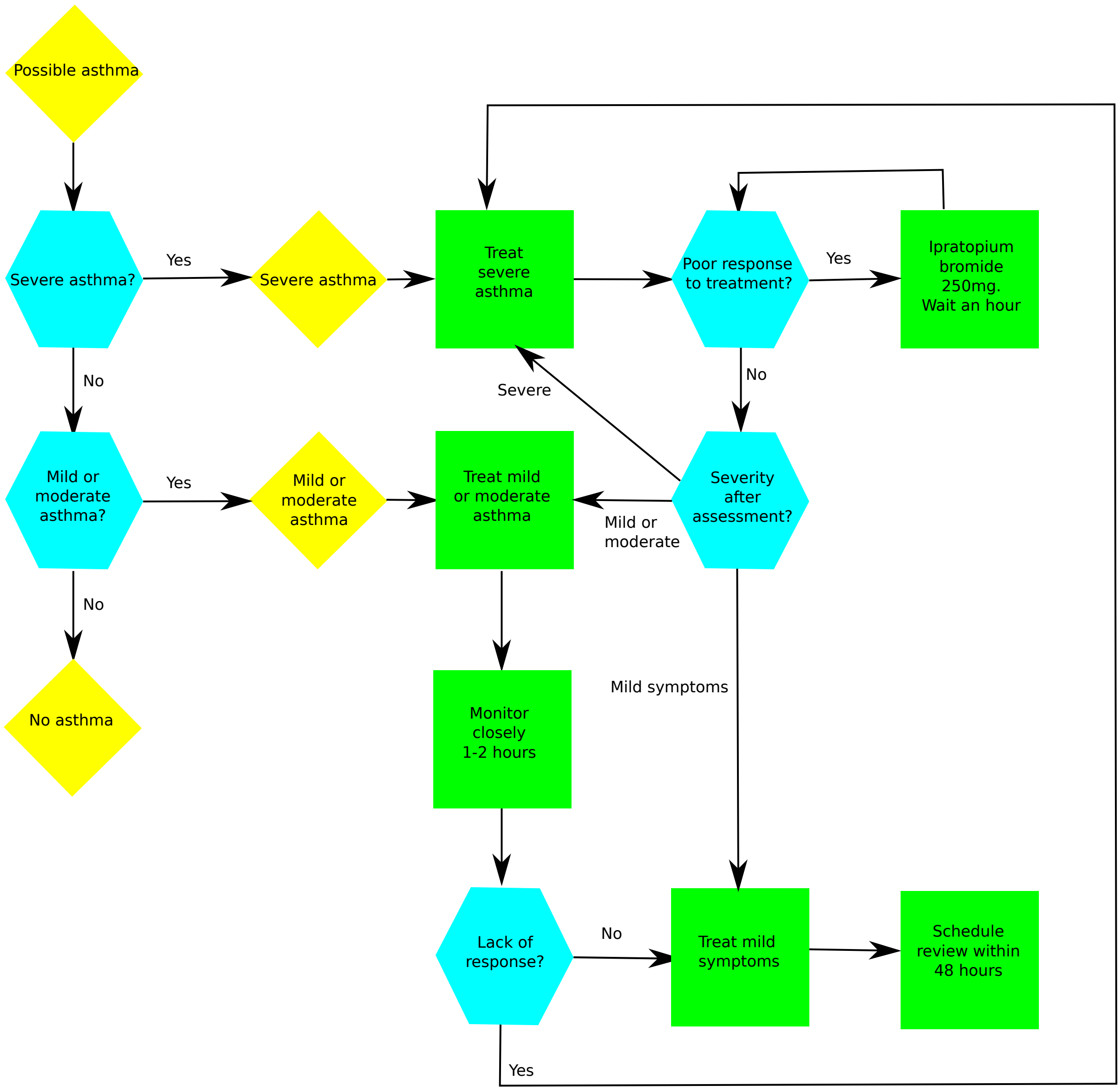
  - **Action step** a representation of a recommended tasks or action. There are three types of actions steps: Medically oriented (such as a recommendation for a treatment), programming oriented (such as retrieving information from EHR), control-oriented that invokes nested control structures (subguidelines or macros to support recursive specification) (De Clercq, Kaiser, and Hasman 2008). Action steps are green squares in figure 3.2.

- **Decision step** represents a decision point in the model. There are two types of decision steps: Case and Choice. In a Case step the decision will be made up of a number of logical expressions, a deterministic decision. On the other hand, the Choice step displays a various suggestions and the agent (clinician e.g.) needs to choose between them (De Clercq, Kaiser, and Hasman 2008). Decision steps are turquoise hexagons in figure 3.2.
- **Patient state step** which characterizes the specific state or condition a patient is in. It can be used as an entry point into the flow-chart, or as an summation to describe the clinical state a patient is in (Boxwala et al. 2004). Each patient state step includes attributes which describes the condition a patient is in (e.g. iron level in the blood has been reduced since last blood sample). When this condition occurs in practice, the guideline corresponding to the Patient state step is executed (De Clercq, Kaiser, and Hasman 2008). Decision steps are yellow diamonds in figure 3.2.
- **Synchronization step** working together with the **branch step** to provide multiple concurrency paths (Boxwala et al. 2004). Multiple guidelines that follows a branch step, eventually comes together in a synchronization step. A continuation attribute specifies whether all or some conditions need to be fulfilled before we can continue to the next step (De Clercq, Kaiser, and Hasman 2008). Synchronization and branch steps are not used in figure 3.2.

• **Asbru** is another approach for computer interpretable guidelines, and sort of a competitor to GLIF and PROforma. The project was developed at the Stanford University, Vienna University og Technology and Ben-Gurion University. It uses time-oriented skeletal plans. These skeletal plans are schemata at different detail levels. To manage skeletal plans, Asbru introduces some key functionality: the representation of high-level goals (intentions), temporal patterns, time annotations, as well as graphical user interfaces to view the skeletal plans (De Clercq, Kaiser, and Hasman 2008).

  In Asbru, skeletal plans are used to represent guidelines. Each plan's functionality is described by attributes such as: preferences, intentions,

Figure 3.2: The paediatric possible asthma guideline Republic of Kenya 2016 modelled using elements from GLIF

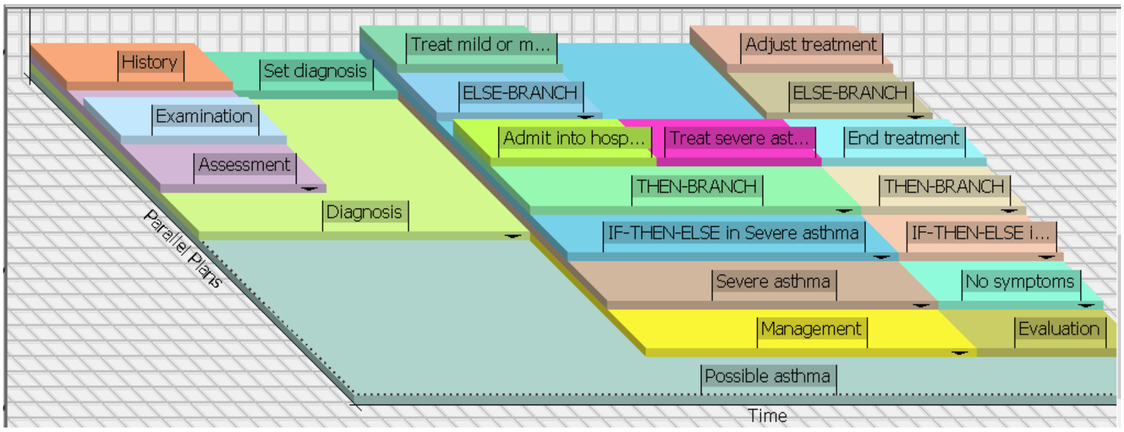


conditions, effects and plan body (De Clercq, Kaiser, and Hasman 2008).

In figure 3.3, we have modelled the paediatric possible asthma guideline Republic of Kenya 2016 using a graphical interface tool AsbruView (The Aasgard Project 2006). The sequential plans are organized from a start point to an end poing along a time x-axis. Plans can be executed in a parallel, so the y-axis is lanes. Subplans are stacked in z-axis, making the presentation of a guideline 3 dimensionally. In addition to execute plans in sequential or parallel order, we also have the option to execute the in an arbitrary order or without any fixed order (unordered). There are also sequential plans and control structures with if-then-else.

For the skeletal plans shown in figure 3.3, we can also set the attributes conditions, intentions, effects and preferences for each of them. These describes the functionality of each plan (De Clercq, Kaiser, and Hasman 2008).

Figure 3.3: An attempt at modelling the paediatric possible asthma guideline Republic of Kenya 2016, using Asbru editor tool AsbruView (The Aasgard Project 2006)



- See the description of **DPF** in other parts of this thesis.

Studying PROforma, GLIF and Asbru we learn that there are several ways to approach the task of making computer interpretable guidelines. We can learn about challenges and see how they solved them. Dissemination, where one needs to regularly update the guidelines and make sure that the different computer systems use the updated versions, is such a topic. Dissemination is relevant for this project, as you want to make sure that every student trains for the most current guideline. You don't want them to memorize old and outdated content.

Our conclusion is that Asbru, GLIF and PROforma are rather large systems, putting a lot of emphasise on working in a hospital setting, communicating with other computer systems such as decision support on the electronic health record. With large computer systems, it is much more difficult to customize.

They also work on a higher abstraction level than the symptoms the clinician needs to look for when doing an examination for asthma. When training clinicians, we are very interested in modeling the details a clinician has to do during a clinical encounter. Asbru, GLIF and PROforma are more concerned with customizing the workflow for one or a sequence of treatments for a medical conditions. We want to use more general models which can be reused to represent several guidelines. With DPF we can create custom domain specific modelling languages.

3.2.4 Designing alternatives

3.2.5 Entity and workflow models

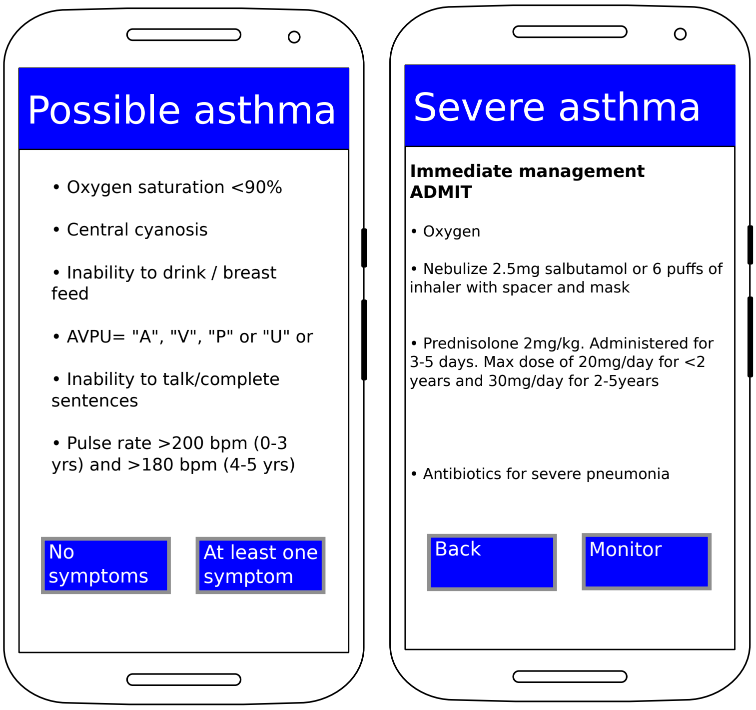
The goal of the iteration was to model patients at different stages of the clinical encounter. The entity model should be able to represent patients in scenarios in quizzes with answer keys.

We based our models on the paediatric possible asthma guideline Republic of Kenya 2016. The first step was to understand every symptom, medication, equipment used in the treatment and keywords such as "admit". Resources like Disease et al. (2011) and Johansen et al. (2018) were some of the resources used, but most importantly was an expert of domain.

The developer made a prototype of his understanding of the CPG, which simulated a clinical encounter with a patient. See figure 3.4. The GLIF chart presented in figure 3.2 was used as a base when making the simulation tool. The simulation would start with listing symptoms of asthma and the clinician would choose among the symptoms, redirecting to the treatment for severe, mild, moderate and no asthma. The simulation would continue with cycles of evaluating the given treatment, and the clinician needs to act accordingly to the evaluation for each cycle until the treatment can be ended. An expert of domain got the task of going through the entire simulations of use cases such as "a patient with severe asthma responds poorly to salbutamol treatment". The stimulation gave a thorough understanding of the guideline, emphasized details which would have been missed or details of the treatment which is left out of the guideline itself, such as differential diagnoses.

The guideline is also unclear at some points, where a discussion with a domain expert was needed. For example: wheeze + history of cough or difficulty breathing. When parentheses are not used, it is difficult to see if wheeze always needs to be present, or if it is enough that cough alone is. It is also difficult to understand the effect and model the situation where

Figure 3.4: Screenshots of the prototype used to simulate a clinical encounter with a domain expert



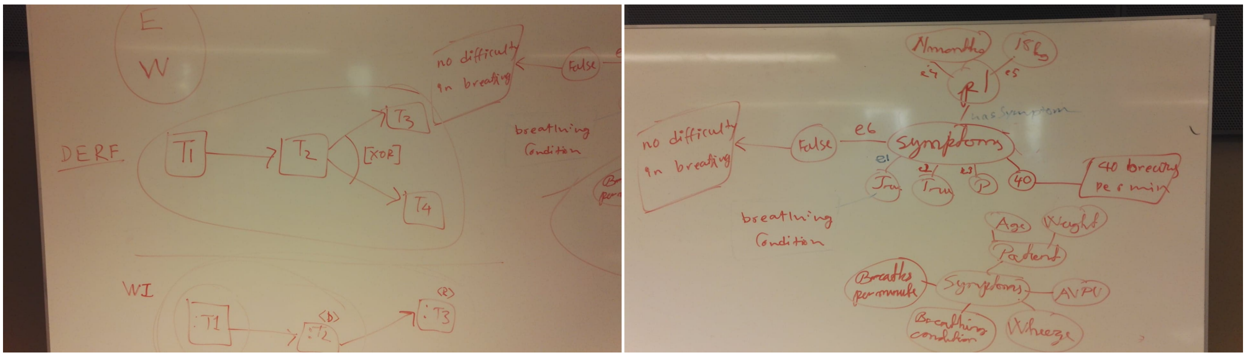
the presence of a symptom at a certain age gives "increased likelihood" for asthma. In computer science the terms need to be very clear.

A domain expert in Model Driven Engineer and DPF was consulted for discussing problems such as inheritance, how to create good sentences from vertex values in the graph, as well as metamodelling and the use of constraints in DPF. In figure 3.5 there are two images from the whiteboard during the meeting. On the left we see a workflow and an instance below. The XOR is a constraint, which tells that we can choose one of the two paths only. On te right we see the entity model, and a suggestion with vertices which make good textual presentation of graph values which can be used in texts.

For evaluating the entity model, a domain expert in medicine made quizzes with factual questions as well as scenario based. By replacing patient related variables in the questions and scenarios, with tags pointing to variables in the entity model, we could see whether the model fulfilled the requirements of displaying good and valid sentences. The sentences were equally good and valid when using different instances of the entity model.

The workflow model got evaluated by making quiz scenarios, and by confirming that these scenarios covered the entire guideline.

Figure 3.5: Pictures of the whiteboard during a meeting with domain expert in MDE and DPF. To the left there are some discussion around workflow models. To the right is entity models with vertices which holds a textual representation of the parent vertex.



3.2.6 User interface

3.2.7 Game Engine

3.2.8 Content flow and creating questions

3.2.9 Focus group

3.2.10 Workshop

3.3 Focus group

The 25th of February 2018, we arranged a focus group at a kitchen in Alrek student home. The participants were:

- 6th year medical doctor student Fredrik Hoel.
- 6th year medical doctor student August Hoel.
- Master degree student in computer science Mohnd Skr.
- Master degree student in computer science Ben-Richard Ebbesvik.

As the project was in a very early stage, the purpose of the meeting was just exploring topics. How the CPGs are used today. How the students work in their practice periods at the hospital. What format are the CPGs in now. What challenges limits the use of CPGs among health workers and medical students at the point of care.

By using a an unstructured interview form, we managed to collect broad and general information, as well as going into detail on interesting topics. As the focus group was small, the contestants could discuss between themselves, highlighting consensus and conflicts Preece, Sharp, and Rogers 2015. A very free and exploitative approach was very successful, as the master degree students in computer science had a very limited knowledge of the medical domain, compared to the medical students. The discussion was documented using an audio recorder.

Topics explored:

- The students are learning new medical routines by studying typical cases. Drilling the routines.

- Red and yellow flags, which are alarm systems they need to be aware of. A red flag is when the patient's condition is quite critical. What triggers the flags and how to act upon them is something the clinicians need to know by heart, as time is critical and the action needed might be advanced like surgery. No time to use the guidelines.

- Clinicians and medical students use a collection of short guidelines in a pocket book format for references. The guidelines are written mostly in text format and sometimes takes use of tables for presentation.

- Mobile devices can't be used at the point of care because of condemnation risks. There are also techniques of consultant with a patient. A mobile device will get in the way for important non-verbal communication.

- For departments they need access to specific cases where there isn't much written material. They need access to scientific articles.

- They brought up the case where doctors in developing countries have so many patients, and the time to each patient is very limited. To be able to look up information in the guidelines, they need to have a format which makes it very fast to extract that kind of information. Flowcharts is more suitable for developing countries where acting quickly is more often important than in developed countries where they can have a focus on being more thorough in green flag situations.

- In addition to guidelines, hospitals can also have their own protocols. The protocols are for situations where the treatment needs to be different than what is described in the guideline. An example is a patient with a blood clot. T he medical personnel in Finnmark will start removing the blood clot immediately, because of the long distance to the nearest hospital. They want to reduce the risk for complications. While in Bergen, they will wait with such a treatment. In developing countries you might have to put into consideration what kind of equipment and staff is available at that specific hospital or health care station. The likelihood of different diseases is different from each geographically position, the patients background or the season in the year. Social and economic status also matters, even in Norway.

- Discussion about presentation of learning material in an application. Medical cases are often too obvious, too simple or too complicated in existing applications. Flashcards. Show image of an ECG or a picture of a symptom.
- Notifying the student about guidelines would be useful. But only for the most common and dangerous conditions, relevant for the students medical field to avoid unnecessary notifications which will only be ignored.
- The diagnostic process. The doctor have several conditions in mind, but tries to eliminate the statistically most common and dangerous first. Trying to narrow down the alternatives until the doctor is quite certain about the medical condition of the patient.

3.4 Workshop

The 22nd of February 2019 we had a workshop. The purpose of the workshop was to

- Identify components in the treatment plan of asthma patients.
- Identify difficulty levels, and how the questions will be more detailed for every difficulty level.
- Make a map of the learning content. Where the content is categorized in components and difficulty levels. Identify paths the student can take through the learning content.f

The antecedences for the meeting was

- Professor in computer science Yngve Lamo. Background in model driven engineering and health informatics.

- Assistant professor in computer science Svein Ivar Lillehaug. Background as a researcher in health informatics.
- Postdoctoral fellow Fazle Rabbi. Background in model driven engineering.
- Medical doctor and PhD student in health informatics Job Nyangena.
- PhD research fellow in interaction design Rosaline Barendregt. Has written a master thesis in gamification.
- PhD candidate in computer science and health informatics Suresh Kumar Mukhiya.
- Master degree student in computer science Ben-Richard Ebbesvik.

Figure 3.6: Anticlockwise from front left: Yngve Lamo, Rosaline Barendregt, Suresh Kumar Mukhiya, Svein Ivar Lillehaug, Fazle Rabbi and Job Nyangena

The meeting started with Ben-Richard informing the status of the project by doing a cognitive walk-through and a demonstration of the application.

Yngve presented ideas for further development of the application. The important thing, was the concept of splitting up the questions in themes which relates to components in the treatment plan. Job helped identify these themes as assessment, diagnosis, management and treatment. Further we identified what type of questions we wanted to ask, and how they fits into different difficulty levels, based on the details of the questions. We ask factual questions for level 1. We use scenarios in level 2, where we apply facts and the detail level is categories. I.e. what class of medication should be administered to the patient. In level 3 we continue with scenarios, but here we ask for much more details, like the dosage of a medication or how often it should be administered.

When playing level 1 the student should get questions from all themes in level 1. When the student completes level 1, the student should no longer get questions from that theme. This is to avoid boring the user by repeating the questions the student already knows the answer of. He should only get questions from themes he struggles with. I.e. on the first run level 1, the user gets every question in assessment right, but have some mistakes with diagnosis and management. Then on the next run, he only gets questions from level 1 diagnosis and management. This continues until he has reached the passing condition on every theme in level 1.

We further identified a dependency in the treatment plan. To be able to do a follow-up, the students first needs to know something about assessment, diagnosis and management of the patient. The follow-up is actually an evaluation of the treatment which have been given, based on the suggested diagnosis. The evaluation will tell how the patient responded to the treatment, and we need the student needs to take actions whether the patient responded or his condition became better or worse. When we have such a dependency, the student needs to complete assessment, diagnosis and management before follow-up gets unlocked. Since follow-up is only relevant in a situation where there has already been set a diagnosis and given a treatment, the follow-up is only part of level 2 and 3, where the questions are given as scenarios.

To complete a level, all passing conditions at that level in each theme have to be met. When the student qualifies for a new level, he only gets questions from the level he plays. The same concept of only getting questions from the themes at that level you haven't met the passing condition, continues for level 2 and 3.

We planned to have a visualization of the passing condition in the application. The passing condition will be shown in a chart in the summary section after each game. The passing condition will be marked as a line over every theme for the level the student plays. The students scores for each theme at that level be shown as bars. When a bar reaches the line, a passing condition is met.

Figure 3.7: Assessment and diagnosis are components in the treatment plan. In the learning map they are themes. Under each theme there are difficulty levels. Questions for each level are written on post-it notes.

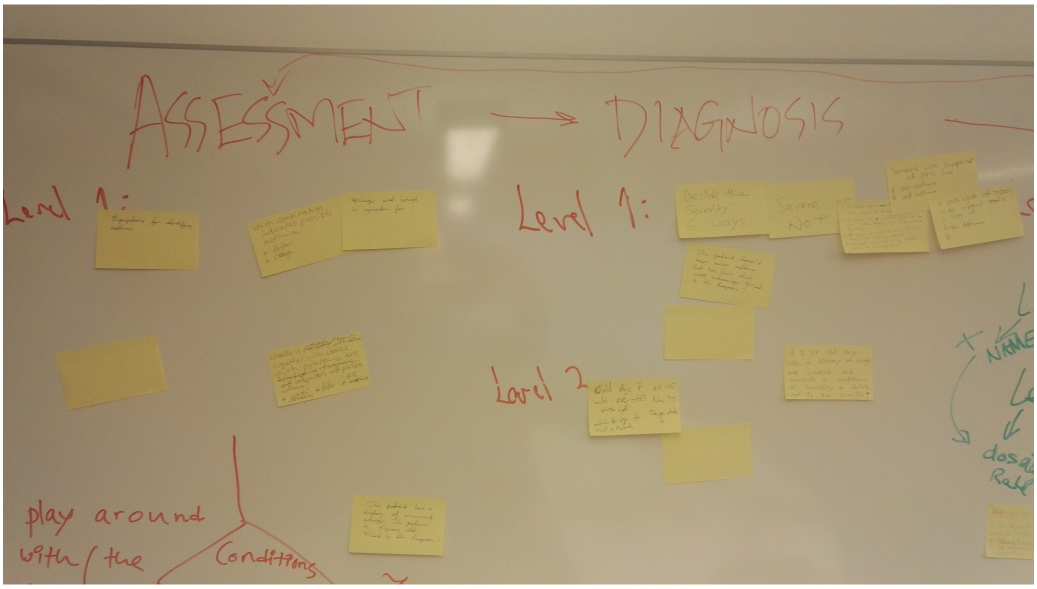


Figure 3.8: Management and follow-up are components in the treatment plan. In the learning map they are themes. Under each theme there are difficulty levels. Questions for each level are written on post-it notes.

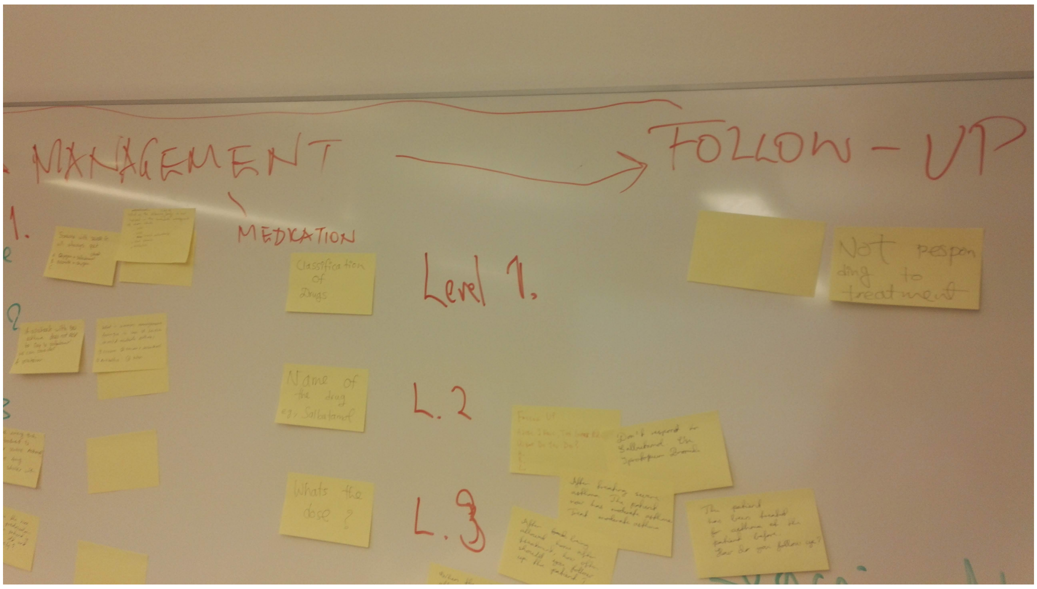
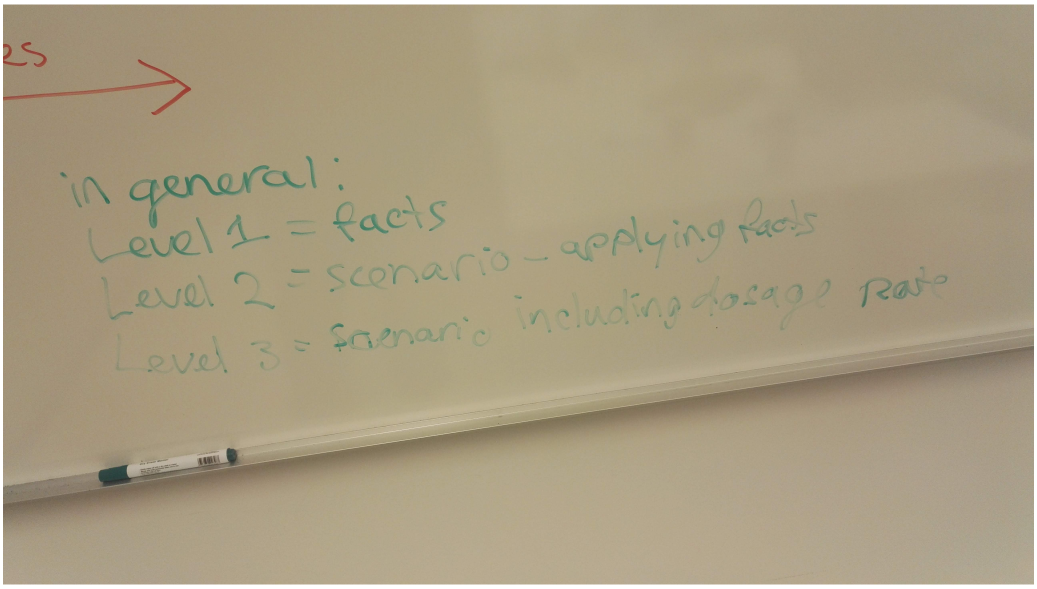


Figure 3.9: What type of questions the student will get at each level.



Job continued the meeting by talking about the guidelines.The paediatric guideline of asthma is called "possible asthma". That is because in an emergency situation asthma is the most dangerous airway condition and can be lethal. If the patient shows signs of asthma, he will be treated for asthma to reduce risks of an unwanted scenario.

We identified users of the application:

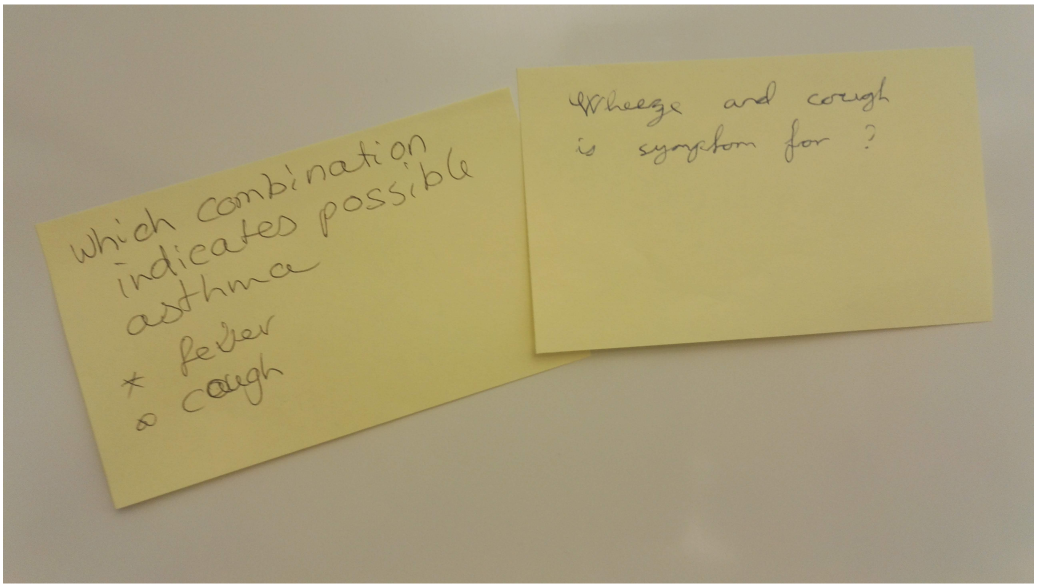
- Formal training, where last year students are reading for their exams.
- Anyone can learn, so it can be used to inform and educate the public.
- In countries such as Kenya, where there are a large deficit in doctors and nurses, sometimes nurses has to work as doctors. Or community workers need to take the role as doctor or nurse. The application will help educating nurses and community workers for such scenarios.

There was also talked about how the situation in medical training is for the student. When a patient comes to the emergency room with severe asthma, the medical doctors will have all their focus on that particular patient. The medical student will typically not take part in the assessment, diagnosis or the initial treatment of the patient. The medical student will typically only take part in the monitoring, evaluation and follow-up of the patient, when the situation is less critical. The application will give the medical student an alternative way to train in assessment, diagnosis and initial treatment of a made-up patient with severe asthma.

Job continued the workshop by going through the Kenyan paediatric guideline of possible asthma Republic of Kenya 2016. This is the guideline we will base our quiz on. Job answered questions from the group about details of the guideline. It is important to understand the general flow as well as the details to be able to make good questions for the quiz. The guidelines is poorly written in terms of wrong use of sentential operators. These mistakes needed to be clarified.

The rest of the workshop was for the participants brainstorming around the questions which will be used in the quiz. Each participant wrote questions on post-it notes, and placed them at a suiting level and theme on the blackboard. At the end, Job went through the questions and we had a small discussion around the suggested questions. We managed to produce question templates to be used in asthma quiz of the application.

Figure 3.10: Suggestions for questions were written on post-it notes and attached to a difficulty level under a theme on the black board.



Chapter 4

Developing a learning tool for health workers

4.1 Extracting knowledge from the clinical practice guidelines

4.2 Data models

4.2.1 DPF

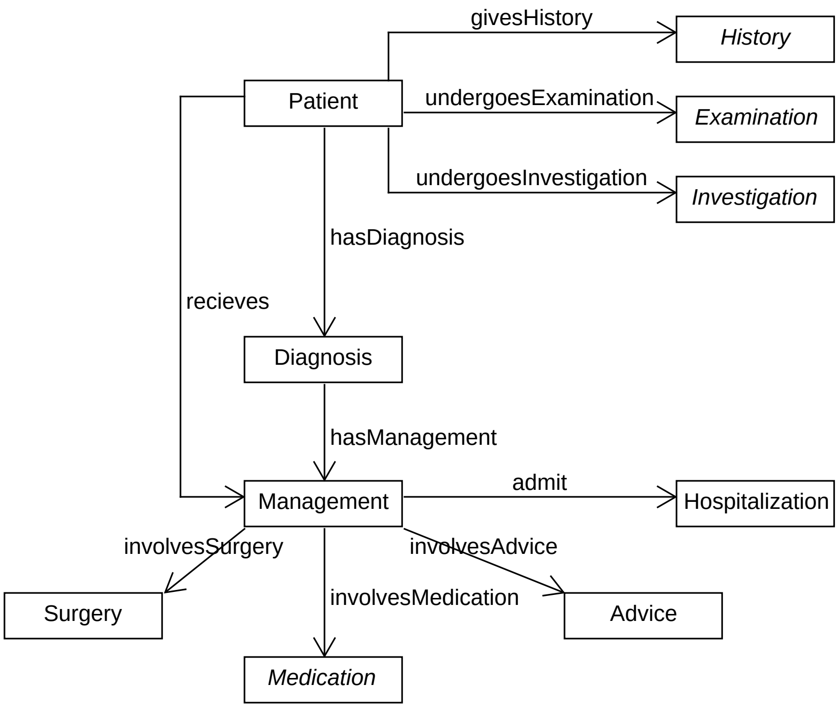
4.2.2 Entity model

We will now present the entity model by showing en excerpt, hiding away the details. This will let us easier focus on the concepts and the flow, as the model itself is quite large and complex. See figure 4.1

The entity graph shows a specific patient at a certain point or time in the clinical encounter. The patient comes to the emergency clinic. He has some symptoms which the clinician needs to uncover, by doing examinations and asking questions about the patients conditions. What the patient or caregiver tells is modelled as history, while quick examinations such as listening to the chest, looking at the skin, count the number of breaths per minute are modelled in the examination vertex. In some cases the clinician wants to test which require more time and resources, such as MRI scan, spirometry or blood tests, which are modelled as investigations.

Based on the the symptoms collected in history, examination and investigation, the clinician will set a diagnosis. The procedures for what to do with a patient with a given diagnosis is modelled under management. Hospitalization is to change the patients status to outpatient, or inpatient if he is admitted into the hospital. He might need some medication or be given advise for how he should deal with his condition the in every day life. Here the model can be expanded with routines found in other guidelines, we have identified surgery as an example.

Figure 4.1: An excerpt of the entity graph. Entity graph represents a patient at a certain point in the clinical encounter

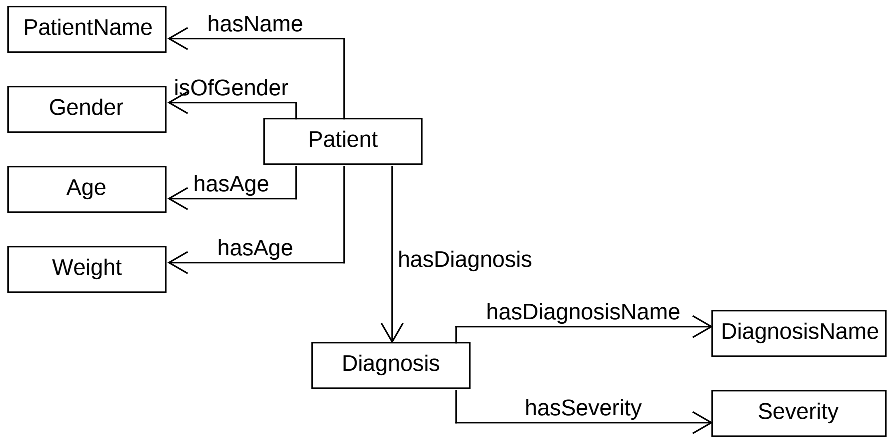


In figure 4.2 we have expanded the Patient and Diagnosis vertices to reveal more details. PatientName and Gender, identifies the patient with a name and gender. These attributes are important when presenting a patient and his condition in a narrative or scenario. By using a name, it is easier for the reader to see that this is the same patient in different stages of the clinical encounter.

A diagnosis has a name. In the paediatric possible asthma guideline Republic of Kenya 2016, the diagnosis has a severity. A lot of medical conditions doesn't have a severity, or they are classified in another way. Here the model needs to be expanded to support other CPGs.

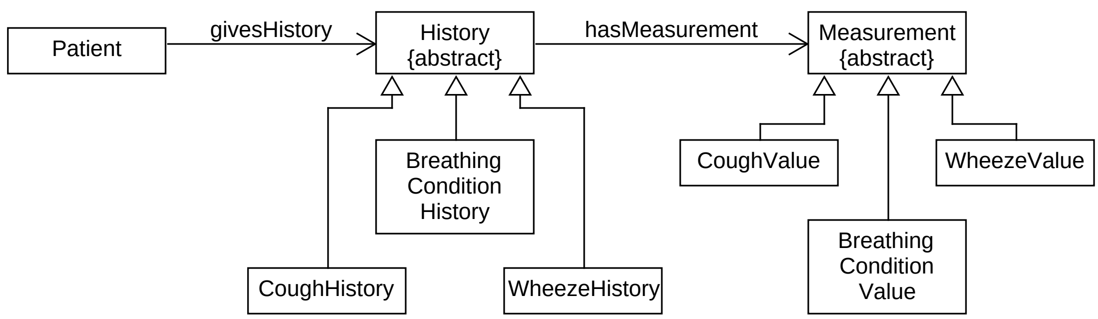
In figure 4.3 we have shown our implementation of the History vertex

Figure 4.2: Showing the details of the Patient and Diagnosis vertices of the entity graph



from figure 4.1. History is what the patient or the caregiver tells about the patient's condition. In the paediatric possible asthma guideline Republic of Kenya 2016 we have identified three symptoms which the clinician can ask the patient or the caregiver about. Here we introduce inheritance, where the specific symptoms inherits the examination vertex. Each symptom the patient or caregiver tell about, will have a measurement. In this specific case, all the history symptoms are boolean. Either they have the symptom or they don't. The symptom values inherits from a Measurement vertex.

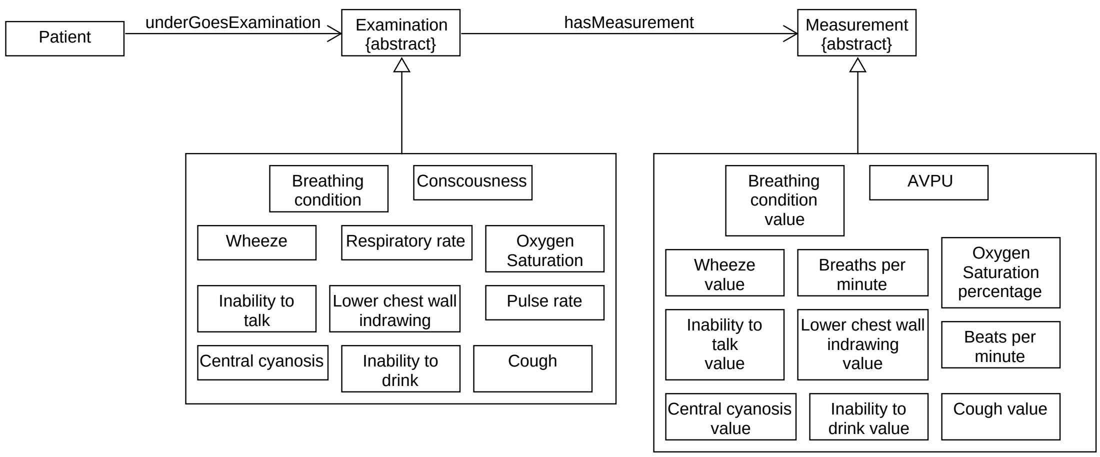
Figure 4.3: Showing the implementation of History in the entity graph. What the patient or caregiver tell about the patient's condition



For Examination we follow the same principles as History. We implement it by letting each symptom inherit an Examination vertex. Each symptom has value which inherits from a Measurement vertex. In figure 4.4 we have shown the inheritances with one arrow and a box. This is of practical reasons when drawing, as there are so many symptoms and it will be confusing to draw an arrow for each of them. Here the values which are stored are a bit more mixed than for History. Consciousness is measured using an AVPU scale, where A is the patient is Alert, V is Verbal, P is responding to pain and U unconscious. These are enumerates, where we store either A, V, P or U. Pulse Rate, Respiratory Rate, Oxygen Saturation are all numerical values. The other symptoms are registered as boolean values. Should I mention clan morphism? I am not sure I have implemented it as it is defined in the papers There is a possibility that a patient tells about a symptom and when the clinician examines the symptom, he finds something else. I haven't asked Job about this, and I don't think it matters for our game.

The implementation of Investigation would be just as we did with History and Examination. For asthma they use a lab test called spirometry, but it is not included in the paediatric possible asthma guideline Republic of Kenya 2016, so we don't include it in our model

Figure 4.4: Showing the implementation of Examination in the entity graph. What symptoms the clinician can observe the patient has



In the paediatric possible asthma guideline Republic of Kenya 2016, there are used six medications to treat the patient. I will talk about each medicine in the model, as they are being administered differentially. Medication is modelled in figure 4.5.

- **Oxygen** is a medication. The CPG doesn't think it is necessary to elaborate on how giving it to the patient.

- For **antibiotics**, the CPG refers to a separate guideline just for administering antibiotics and is therefore out of this scope. For this reason, we just say administer antibiotics here, and don't go into the details.

- **Prednisolone** is a steroid. The clinician will administer a low dosage and increase the dosage over time until getting the desired effect. The clinician will then decrease the dosage over time to stop the treatment. For this reason we have a dosage vertex in the model, which represents the current dosage of prednisolone administered to the patient. The dosage will change as the clinician adjust it. Prednisolone has a duration vertex, as the medication should be administered for 3-5 days. The dosage of prednisolone is calculated in regards of the patient's weight, with a maximum dose per day depending on the age of the patient.

- **Corticosteroid** is another steroid, which is given to in scenarios of recurrence asthma symptoms. The CPG specifies that corticosteroid should be inhaled. This is represented in the model by the Route vertex, where other medication can be inhaled, injected, taken orally (as with pills) as some examples. The method the which will be used to inhale corticosteroid is represented by the Form-vertex. The Form is MDI with spacer, preferably with spacer with face mask. Form and Route are represented by strings as free text. However, we could make predefined vertices which inherits Corticosteroid-Route to give the content author to choose from. The Route of a medication is quite limited compared to Form for all medications.

- **Salbutamol** is inhaled to open the airways of an asthma patient. It is modelled much like corticosteroid, except it has a Rate vertex. The Rate vertex tells at which rate the patient should be taking salbutamol. Asthma patient is given salbutamol at a rate of 2.5mg per hour. The Duration is up to one hour or three doses if needed.

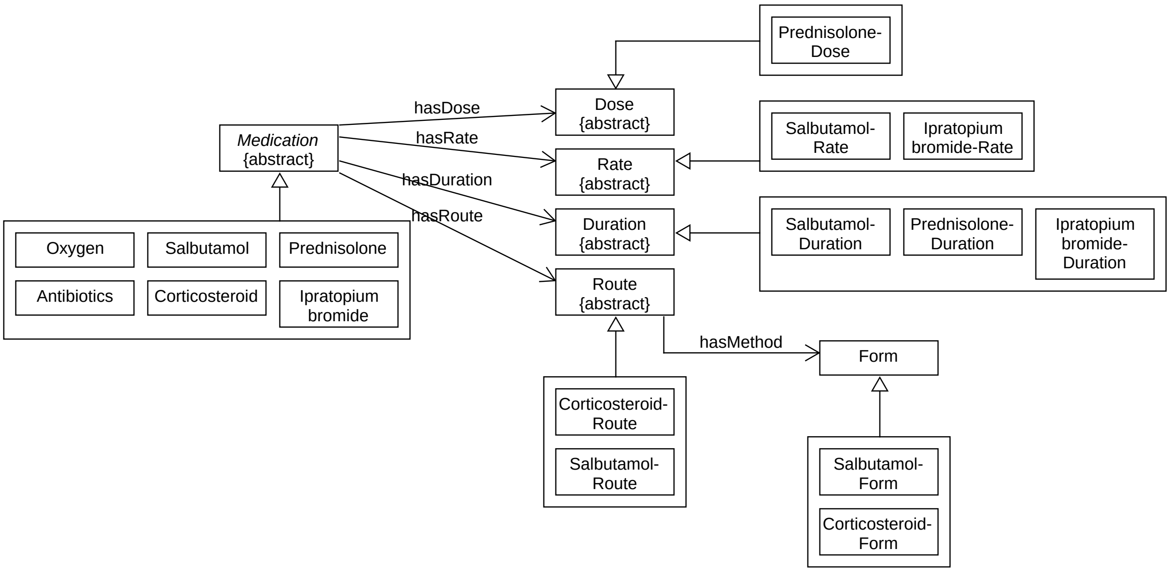
- **Ipratropium bromide** is modelled much like salbutamol with a Rate and a Duration. The CPG doesn't specify a route or a method.

Talk to Job about this model tomorrow. Oxygen, prednisolone and ipratopium bromide may all have a route and form. Talk to Job also about evaluation.

4.2.3 Workflow model

We model the clinical encounter with a workflow model. The clinician starts with the assessment, where he examine the patient and listen to what the

Figure 4.5: Showing the implementation of Medication in the entity graph. How to administer a medication to patient



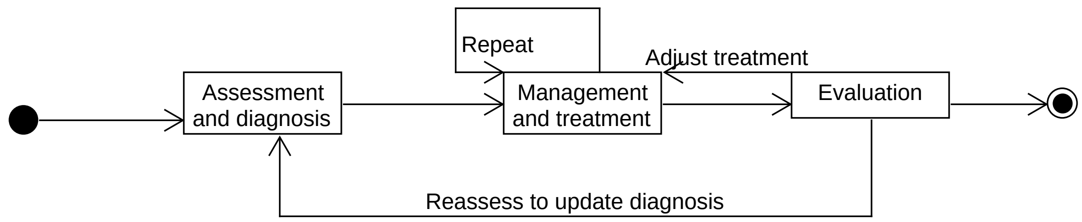
patient and the caregiver has to say about the the patient's condition. The clinician starts to get an idea of what condition the patient may suffer from. The clinician continues with the diagnostic part, where he asks more targeted questions to the patient and caregivers about the condition, do more of the examination and perhaps order lab tests as part of the investigation. This process can strengthen the clinician's assumptions about the condition, and he may be able to set a specific diagnosis.

The next step is the management and treatment. This can be changing the patient's status from outpatient to inpatient, do surgery, medication, physiotherapy, cognitive behavioural therapy or other forms of treatment. The treatment may be done in iterations or repeated.

The final step is to evaluate. The treatment may have to be adjusted to get the right effect. The diagnosis has changed, for example severe asthma is mild asthma after treatment. Or we have initial set the wrong diagnosis, for example we have treated a patient for possible asthma, but in fact an object was stuck in the airways of the patient.

The idea of the workflow model is to describe the process of a clinical encounter. When making scenarios for the game, we know in which order the scenarios should come. The entity model is also connected to the workflow model. When doing a an assessment and diagnosis, you are looking at the examination, history and vertices of the entity model, where the diagnosis

Figure 4.6: The workflow models is a model of the clinical encounter



vertex answer to what the specific diagnosis is. For doing management and treatment, you look at the vertices under management in the entity model. An evaluation will be done by looking at the examination and investigation vertices to see if the patient has become better. The treatment needs to be adjusted accordingly to the evaluation. If the evaluation says we can't do more for the patient and there's no need for a follow-up, we exit the workflow model.

Should perhaps reference to Rabbis articles here? Coordination of multiple metamodels, with application to healthcare systems. A flexible metamodelling approach for healthcare systems.

4.2.4 Metamodeling

I struggle when it comes to talking about MDE, DPF, metamodeling and model transformation, as I lack very basic knowledge about the subjects. Rutle, Rossini, Rabbi are all hard to read

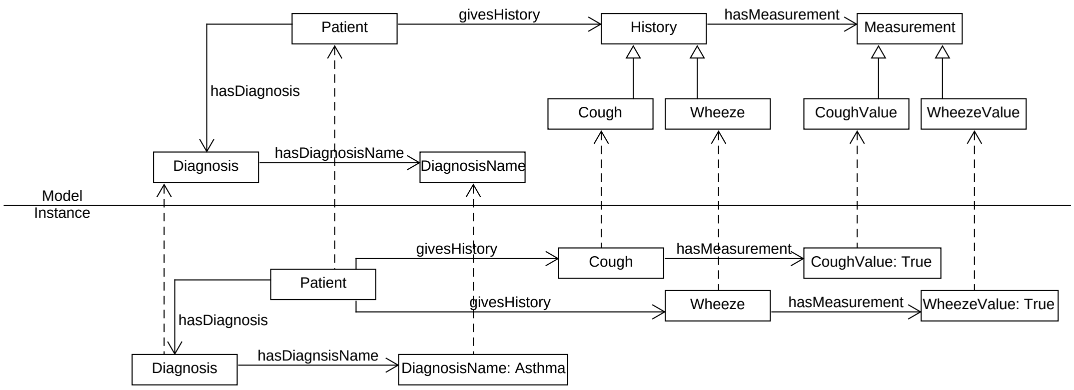
In figure 4.7, we make an instance of the entity model. An instance of the entity model describes an actual patient at one point in the clinical encounter.

For an instance to be valid, the vertices and edges have to correspond to a part of the model. We demonstrate this by adding dotted arrows in figure 4.7.

In figure 4.7 a patient tells the clinician that he struggles with a wheeze and a cough. Cough and Wheeze inherit from History in the model. Difficulty breathing is part of 4.3, but is not represented here as the patient hasn't brought up this issue or been asked about it. We see how two inheritances og History translated in the instance from the model. The Measurement vertex holds the measurements of the History vertices. A patient with a wheeze and a cough is diagnosed with asthma, which is shown in the instance.

In figure 4.8 we show a entity instance working together with the work-

Figure 4.7: A model and an instance of the entity model. For a valid instance, every vertex and edge in the instance has a corresponding vertex and edge in the model.



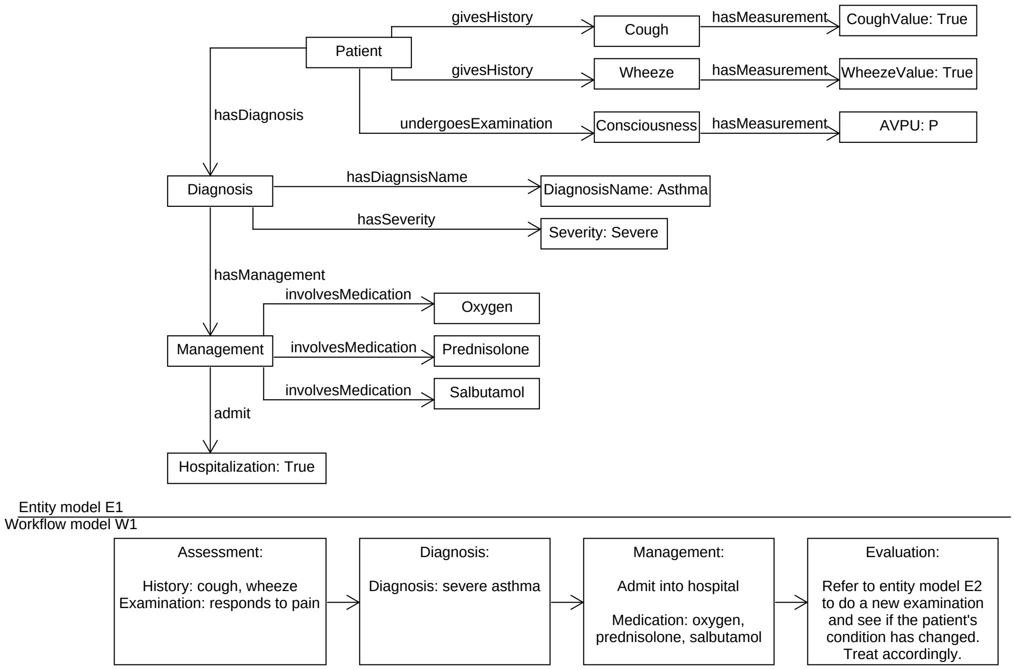
flow model. For the assessment, we look at the History and Examination vertices. For Diagnosis, the DiagnosisName and Severity. Keep in mind that under Diagnosis, the clinician may do further examinations and questions to the patient to confirm his assumption, or which may cause him to think about other diagnosis. Management, the asthma is severe so we change the patient's status to inpatient by updating the Hospitalization vertex. We also look at the Medication vertex under Management. We only care about the medications for now in this example, and not how the medications should be administered. The Evaluation holds a reference to a new entity instance, which holds the updated information about the patient's symptoms. The clinician needs to act accordingly and adjust the treatment.

4.2.5 Game model

4.3 Game engine

As each question in a quiz are related to a certain component in the treatment plan or theme in the learning map, the student will be measured how well he performs on each of these themes. For the asthma guideline (Republic of Kenya 2016), we have identified four themes. Assessment where the student will be tested in the initial examination. Diagnosis, where the student will determine a diagnosis as well as the severity. Management, where the student will determine which actions should be done to treat and best give the best care to the patient. The last discipline is the follow-up, where the student

Figure 4.8: An instance of the workflow model at the bottom, working together with an instance of the entity model at the top



will be tested in evaluating the treatment, give advise to and educate patient and caregivers, provide the right medication and regular follow-up.

By splitting up the score in themes, the student can easily see which areas he is strong and where he needs more training.

We can also adapt the questions in each discipline to the student's level. If the student has proven to be very good in providing the right amount of medicine to asthma patient, we can provide more difficult questions to challenge the student some more. If he struggles at setting the right diagnose, we can provide more basic questions to strengthen the students basic knowledge.

The disciplines should be automatically picked from the entity (and worflow?) model.

The tree structure of discipline scores. Diagnosis have examination, investigation, setting the severity. Management have advises, medication, admit, surgery and so on.

The student will also be provided with a total score, which will be the average score of each of the disciplines. The student can compare the total score of e.g. the asthma quiz and the jaundice quiz, and see which medical condition he needs to train more on.

4.3.1 Dynamic Content Management

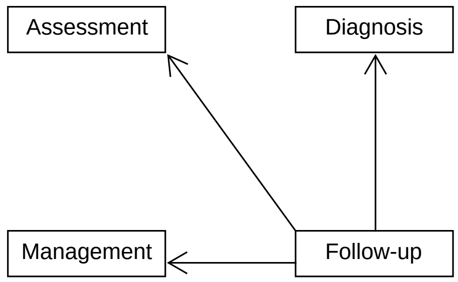
The game engine is based on some of the concepts presented in the articles of Eide, Kristensen, and Lamo (2008), Kristensen (2011) and Kristensen, Bech, and Dyngeland (2013). The motivation for using DCM, is to support the principles of adaptive learning as well as flexibility in the learning process. Adaptive learning means that the student can solve problems which are suited to his knowledge level. Flexibility in the sense that the student can go through the learning content in many different ways.

Eide, Kristensen, and Lamo (2008) presents a dynamic content management system (DCM) made for e-learning. In DCM the focus is on removing the tight coupling between the learning material and the teaching course. By analysing the learning material and course, they can define conceptual atomic units of knowledge which they put into a knowledge repository. From this repository they may draw knowledge elements and organize into the hierarchy of a course. To model a course they use concept maps, which are directed graphs, where the vertices are concept labels and edges indicates the relationships between vertices. DCM operates with three concept maps: knowledge map, learning map and student map. The knowledge map is used to model the entire content of the knowledge repository and the hierarchy structure of a course. A learning map is used to model a specific course and is a representation of the learning process. The content units (vertices) in the knowledge map gets expanded, and becomes evaluation and resource vertices in the learning map. Content units from the knowledge map can be omitted if they are not needed in the specific course. Detailed prerequisites can be specified for the content units. The student map represents the progress of a specific student taking a specific course. The edges shows which resources he has used, the evaluations of the student and in which order.

When we apply this to our project, we will first decouple the content from the flow-chart of the paediatric possible asthma guideline (Republic of Kenya 2016). To decouple the flow-chart, we use the workflow model as a helping tool. In figure 4.9 we have identified knowledge elements and hierarchically structured the knowledge repository into a knowledge map. The edges shows the dependencies in the learning process. To learn how to evaluate a treatment and act accordingly to the evaluation, the student needs to know something about how to set a diagnosis and how to initially treat it. Hence there is an edge between follow-up and the other knowledge units.

In table 4.1 we have chosen a knowledge element from the knowledge map, and defined a content unit. The content unit contains a theme "Assessment". Resources is the learning material, which are relevant sections of the guideline. Evaluations are the tests to see if the student has reached the

Figure 4.9: Knowledge map



learning goals. At level 1 we try to learn facts about the guideline. In level 2 and 3 we give the student scenarios to work with.

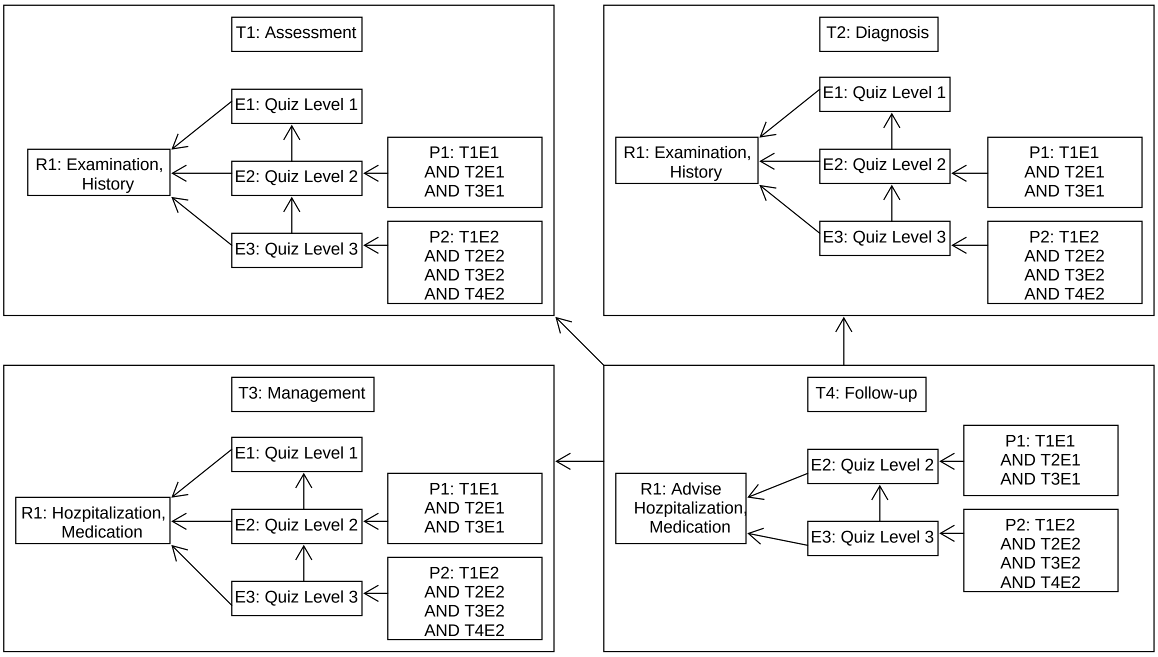
Table 4.1: The content unit Assessment

T1: Assessment		
Resources	Evaluations	Aspects
R1: History and examination sections in the possible asthma guideline (Republic of Kenya 2016)	E1: Quiz Level 1	Facts
	E2: Quiz Level 2	Scenario
	E3: Quiz Level 3	Scenario

In figure 4.10 we have identified four content units T1: Assessment, T2: Diagnosis, T3: Management and T4: Follow-up from the knowledge map. The importance of these content units in the asthma guideline (Republic of Kenya 2016) was the reason these four got selected. The hierarchically structure of the knowledge map, makes the child nodes of the content units become learning material for their parent nodes.

Inside the content units in the learning map, we see the relationships between resources and evaluation. There is also dependencies between the content units. To be able to do a follow-up, a student needs to learn the assessment, diagnosis and management first, because follow-up is an evaluation and reaction to how the patient responded to the previous steps. We have also specified the prerequisites for each evaluation. The prerequisites are written as logical expressions, as seven edges and operators per content unit would be confusing to read. What the prerequisites says is that all level 1 evaluations need to be completed before any level 2 evaluation can be taken. All level 2 evaluations need to be completed before any level 3 evaluation can be taken.

Figure 4.10: Learning map



A student map would show the progressions for one specific student, and the path he has taken and the scores for each evaluation. Table 4.2 shows a student's student map for the Assessment part of the course. He got the score 34 on the first evaluation. 34 matches the passing condition for assessment level 1, so he got a passing grade for that test. However, he scored 43 points on evaluation 2 and didn't meet the passing condition for that test. He has no attempts for evaluation 3, as he doesn't meet the prerequisites for that test. The tests which have been completed are stored in the database on the students phone. In that sense, the database shows the student's current position in the learning map.

4.3.2 Multiple-try feedback

The quiz uses a concept which is called multiple-try feedback (MTC). That means for every question the student gets more than one attempt to get the answer right. A feedback will be given immediately after each answer is submitted. The feedback consists of a message which tells whether the answer is correct or wrong. If the answer is correct, the user will receive

Table 4.2: Student Map T1:Assessment

T1: Assessment		
Resources	Evaluations	Passed
R1: History and examination guideline	E1: 35 E2: 43 E3:	True False

"correct" and an explanation of the answer. If the answer is wrong, there will be no hints or explanations than just "incorrect".

Concept	Abbreviation	Feedback after each question	Multiple attempts at each question	Hints on wrong answer
No or delayed Feedback	NF or DF	No	No	No
Knowledge of Correct Response	KCR	Yes	No	No
Multiple-Try feedback with knowledge of Correct response	MTC	Yes	Yes	No
Multiple-Try feedback with Hints	MTH	Yes	Yes	Yes

The point of doing MTC, is to make the student think over what was wrong with his first answer. Did the student misinterpreter the question? Was there a detail he missed? Does the student lack the knowledge or was he just sloppy in his first attempt?

Clariana and Koul (2006) did a study where they divided 82 students into five groups. DF-, KCR, MTC and two control groups. The first control group got a text and a question at the end. The second control group got a text, but there were no question given. After 5 days,post-test was held to see what the students had learned and remembered. The post-test questions were either identical to the questions in the learning material, transposed where the order of the stem of the question and the correct-response gets reversed, paraphrased where post-test questions had the identical content as the learning material, but the phrasing was different and used different words, and a combination of transpose and paraphrasing. The results showed that DF and KCR groups performed better on identical, transposed and paraphrased-transposed questions. MTC performed better on paraphrased questions. The conclusion was that DF and KCR was much better methods for remembering the learning material word for word, but MTC was better when you have to think and reason about what you have learned.

Attali (2015) further did a did a study on NF, KCR, MTC and MTH using open ended and multiple choice questions on mathematical problems. They showed that solving an open ended question rather than multiple choice was a more efficient way to learn. The learning outcome was the same for the students using NF and KCR. However the learning transfer was greater when using multiple-try (MTC), and even more so when getting a hint on incorrect answer (MTH). They explained the results effortfull and mindful problem solving. In a multiple-try feedback, the user will have to reflect on their errors, re-evaluate the problem and understand the initial error. An open ended question will also require more effort of the student, as they have to generate a an answer rather than selecting from alternatives. On the combination of multiple-try and multiple-choice, it was suggested that some users might be less likely to review their incorrect answer and mindlessly clicking on another alternative.

According to Morrison et al. (1995), students which perform badly on answer until correct questions, will often become frustrated, loose interest for reviewing the material and probably depress learning.

As thinking and reasoning about a diagnosis, treatment plan, evaluation and follow-up of a treatment is part of a medical procedure, we believe that multiple-try feedback is the right approach. Because of the nature of a mobile app, where gestures are more convenient than typing sentences, multiple-choice seems to be the right choice even, though open ended questions has proven better results in. There's also a technical problem with evaluating free typed sentences.

Some of the questions in the app are too simple for a hint to be meaningful.Example: "the symptoms for asthma is" and the answer can be "cough and wheeze". Where hinting "cough", would be giving away the answer, especially in a multiple-choice format. However, the data model supports hints as links to external learning material. E.g. the student could look for the answer in the guideline itself.

We solved the "answer until correct"-problem described by Morrison et al. (1995), by having a "read more" button displayed upon incorrect answer. The "read more"-button will display the correct answer, an explanation and continue to the next question. Avoiding the user becoming frustrated and discouraged by having to brute-force the answer keys to progress.

4.3.3 Reward system

By having multiple-try feedback, another problem rises, and that is the reward system. If there is no penalty for incorrect answers, a student which needs ten attempts per questions, will get the same score as a student which answers all the questions correctly on the first attempt.

Attali (2015) solved the problem by giving 1 point for answering correctly on the first attempt. 2/3 points for the second attempt, 1/3 for the third and 0 points if the third attempt was incorrect. A limitation with this method is that it makes no sense for the student to make more than three attempts. Morrison et al. (1995) had another strategy where they adjust the scores by dividing the total score by the total number of attempts during the quiz. A consequence is that attempt number two will have a huge penalty which is halving the students total score. While attempt number twenty will give a very small penalty from attempt nineteen. A method to dampen this effect could be dividing the total score by the sum of reviews and number of questions.

The solution we used was having a fixed value for every answer alternative. The quiz author chooses the penalty for each distraction and reward for each answer key. The idea is that the distractions can have some sort of degree of wrong or right, and this can be reflected in the scoring. On the question "what are the symptoms of asthma?", "difficulty breathing" is a more correct answer than "fever", as "difficulty breathing" is a symptom of asthma in combination with wheeze. Fever is not an asthma symptom at all. In future work, the penalties can be automated as you can see from the entity model whether the symptom belongs to the asthma guideline or not. A distraction from respiratory disorders may give a larger penalty than a distraction from the asthma guideline, but smaller penalty than symptoms not belonging to respiratory diseases.

Both Attali (2015) and Morrison et al. (1995) avoids the scenario where the user gets a total minus score. This may be a strength of these methods, as a negative total score seems like a very harsh feedback and might demotivate the student. In our solution we use negative numbers as penalties on distractions, such that a negative total score may happen. We try to limit the likelihood of a negative score by providing a very reward for a correct answer and a very small penalty for a distraction. Typically the reward is 10 points and the penalty -1 og -2 points. The intention is to encourage the student to review the incorrect answer and try again. As the format is multiple-choice and the penalty-reward ratio, there is a little risk involved trying multiple times. But giving up by clicking "learn more", the student will not get an additional penalty, but will miss out on the reward. By clicking answer alternatives mindlessly and consequently clicking "learn more" will probably not end up in a negative score, but is more likely to end up in a negative score than mindlessly click answer alternatives until correct.

A solution to having a not very strict game, encouraging to playing and learning, one can also have a very strict examination version. The idea is that after examination, the results will be sent to the lecturer (or a governing body of some kind) to evaluate what the overall knowledge of the students, as well as details of what the students are really good in and where do they struggle. The lecturer can then target the weak of points of the students in one of the next lectures.

4.3.4 Constructing scenarios and answer keys

A quiz consists of several questions, where each question has answer keys and distractions. At level 1 these questions will be factual, and in level 2 and 3 we will work with scenarios. We write the questions in the format of a template, where we use tags to refer to variables in the entity model. The tag is a path in the entity graph. The game engine will traverse through the graph and return the value of the vertex specified by the path. The entity graph represents a patient at a certain point in the assessment or treatment. By replacing the entity graph, we can reuse the same template with different patients, generating many different questions with the same template.

The same goes for the answer key. The answer key can be a text, one or several tags referring to variables in the entity model, or a combination of both. The application always uses the same entity graph for the scenario and the answer key. When the application traverses entity graph, the answer key and the question given always matches as they are the same patient at the same given time.

One of the problems we encountered with this method, was how to present the variables returned by the graph in a text. As an example, we can look at some of the symptoms of asthma and their type.

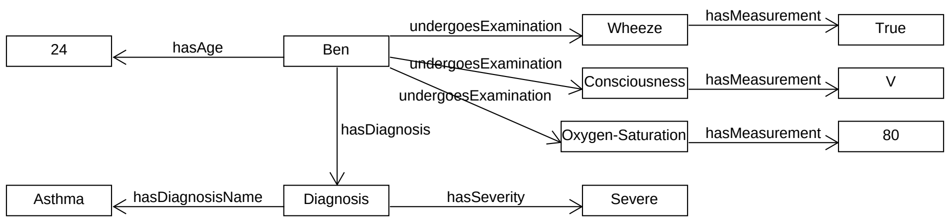
- Wheeze is a whistling sound when you breath. In the model it is represented as a boolean. True or false. Either you have it or you don't.

- Age is relevant several places in the guideline. In the model it is stored as an integer.

- AVPU is a scale system, which clinician use to measure a patient's level of conciousness. A is alert. V, the patient is verbal which means he can somehow respond to questions. P, the patient responds to pain. The patient will react if you pinch him. U is unresponsive or unconscious. He doesn't respond to either voice or pain. The AVPU is represented in the model as an enumerate.

- Severity classifies the asthma severity to be either severe, model or mild. The severity is represented as a string in the model.

Figure 4.11: The patient object used with the template




```
Question:
 A <%Patient.hasAge.Age%> months old patient arrives at the emergency department.
 The patient has a <%Patient.underGoesExamination.Wheeze%>,
 has conciousness level
 <%Patient.undergoesExamination.Consciousness%>
 and has an oxygen saturation of
 <%Patient.undergoesExamination.Oxygen-Saturation%>%.
 What is the asthma severity?

Answer key:
<%Patient.hasDiagnosis.Diagnosis.hasSeverity.Severity%>
```

This translates to

```
Question:
A 24 months old patient arrives at the emergency department.
The patient has a true,
has conciousness level V
and has an oxygen saturation of 80%.
What is the asthma severity?

Answer key:
Severe
```

We see that the template author needs to be aware of how the variables will be printed. Here he knows that the model will just return an integer for the oxygen saturation. He writes a descriptive text of the value first, and then adds a percentage after the variable. The severity gets nicely printed as answer key.

The problem is the boolean for wheeze, which prints a "true". It really should have printed "wheeze". We solved the presentation of consciousness in the same way as we did with oxygen saturation. However, it could be nicer to write "responds to pain", rather than just "V". When the child 12 months or older, it is often easier to read if we can present the age in years.

Another problem arrives when we replace the entity graph with another, where some of the examinations haven't been done. In traditional model driven engineering, we use something called the closed world assumption (Sadowska and Huzar 2019). If a node doesn't exist, we say the value is false. But how can we say that patient doesn't have wheeze when we haven't examined? In open world assumption a none-existing vertex is "unknown" (Patel-Schneider and Horrocks 2006) (Bergman 2018), and this strategy seems more correct for our scenario. If the vertex doesn't exist, we simply return an empty string. This further motivates ut to remove the variable specific text from the template, as we don't want text representation for a variable we don't present. Example when consciousness and oxygen saturation haven't been examined:

```
Question:
A 24 months old patient arrives at the emergency department.
The patient has a true, has conciousness level
and has an oxygen saturation of %.
What is the asthma severity?
```

How we solved the problem was to add a textual presentation vertex to each of the vertices in the graph referred to by the template. If there exist a presentation for the vertex, return the presentation. If there doesn't exist a presentation, simply return the variable.

```
A <%Patient.hasAge.Age%> old patient arrives at the emergency department.
The patient <%Patient.underGoesExamination.Wheeze%>
<%Patient.undergoesExamination.Consciousness%>
<%Patient.undergoesExamination.Oxygen-Saturation%>.
What is the asthma severity?
```

Which translates to

Figure 4.12: Making graph variables fit a story format

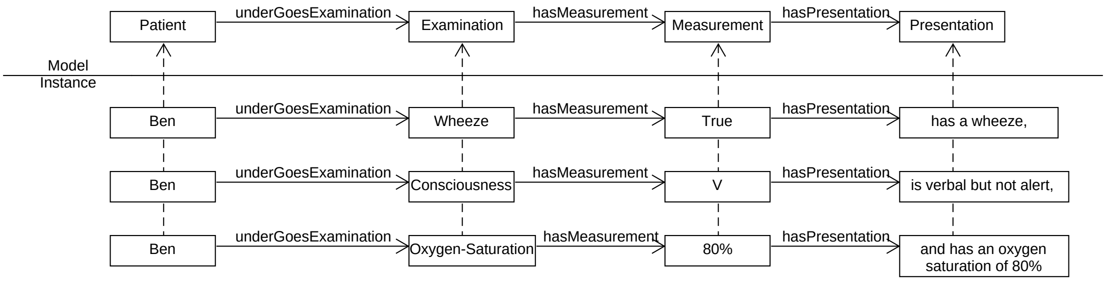
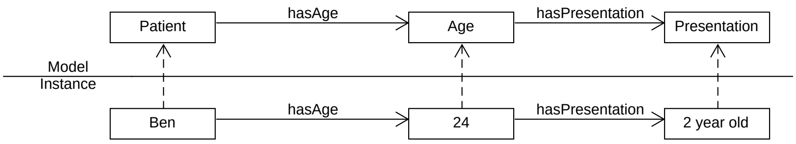


Figure 4.13: Making graph variables fit a story format




```
A 2 year old patient arrives at the emergency department.
The patient has a wheeze, is verbal but not alert
and has an oxygen saturation of 80%.
What is the asthma severity?
```

This also works with the open world assumption, as a patient which haven't undergone consciousness and oxygen saturation examinations, would result in the following text:

```
A 2 year old patient arrives at the emergency department.
The patient has a wheeze.
What is the asthma severity?
```

For future work, the commas and "and" should not be in the presentation vertex. This becomes a limitation where the variable can only be used in a list and has to be in a specific place in the list. The solution would be to have a list tag in the template, and have all the paths inside that tag. Then the game engine can see how many of the list items are in the graph, and can set the commas and "and" at the appropriate places.

4.3.5 UML model of how all the classes are connected in Game Engine

The conceptual model of the game engine is shown in figure 4.14. Category is a quiz game for a certain CPG, such as the paediatric possible asthma guideline Republic of Kenya 2016. Each of these quiz categories are divided into several disciplines. Disciplines are themes in knowledge units, which we identified using DCM (dynamic content management). Each of these knowledge units contain evaluations at different difficulty levels. These evaluations are collections of questions, which contain a collection of multiple choice questions and answer elements.

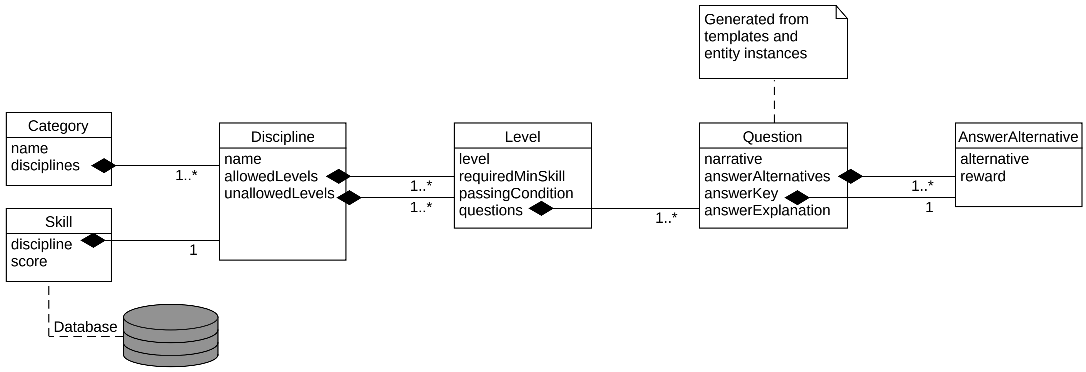
The quiz data are read from JSON-files, which are produced by a content author. The quiz author defines a quiz for a specific CPG. He identifies themes for content units, which contains evaluations with varying difficulty levels. The questions are written in a template format, which we have already covered. The templates contain tags, which refer to vertices in the entity graph, representing a given patient. The answer key is also such a tag, referring to one or more vertices in the entity graph. When the quiz is initialized, the tags will be replaced with values from the entity graph. The content author will write answer alternatives for each question, where he specifies a reward (or penalty) for each of the answer alternatives. An answer alternative which matches the answer key value in the entity graph will represent the correct answer. The content author needs to know what is in the entity graph to be able to provide meaningful rewards.

The student skills is the scores from the last played game, and is fetched from the database on the student's phone. The allowed- and unallowed levels are calculated from the student skills and the requiredMinSkill for each level. Unallowed levels are levels that the student can't play and are locked for the student, as the student's score is lower than the required minimum skill.

A new instance of skill will be created when the student starts a new game. This instance will hold the scores for the current game, and will be stored to the phone's database once the game has been completed. This score determines which levels the student can play in the next game of this category.

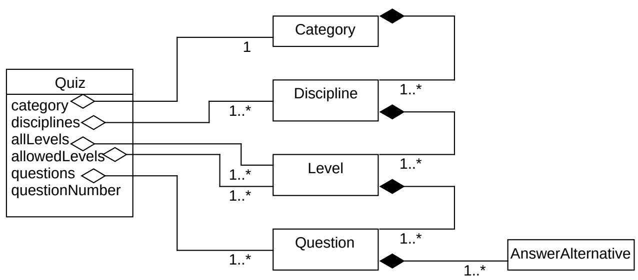
One problem with the conceptual model is that it is somewhat complex. Once we have determined which levels we should pick questions from and we have generated the questions, we really don't need the structure. Especially when playing scenarios, it is nice to just go through an ordered array, instead of dealing with "now I've played the third question of level 2 diagnosis, the next question in this scenario is the third question in level 2 management, unless I've already completed level 2 management in the previous run of the

Figure 4.14: Conceptual model for the game elements. There are two instances of skill. One which represents the student's previous game stored in database. The other one represents the ongoing game in memory.



game. Then the next question is follow-up instead". We rather deal with the problem at the initialization phase and just go through the array when playing the game. A solutions is to implement the façade pattern (Gamma et al. 1994), such that other parts of the system can use the game engine, without having to deal with the underlying complexity.

Figure 4.15: Façade pattern hides the complexity of the game engine and makes it easier for other parts of the system to use it.



Should I show how I populate the models with data from JSON and database? I'm not very happy about the code on that part.

4.3.6 Types of questions

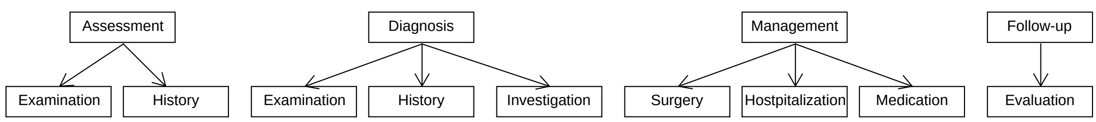
We can further identify which kind of questions we are going to ask, by further examining the entity model and the workflow model combined. When we are doing an assessment in the workflow model, we are looking at the examination and history vertices of the entity graph. History is what the patient or the patient's dependent can tell around the patient's condition. Such that he has been coughing a lot the last days. Examination and history will provide the clinician with an idea of a diagnosis.

Diagnosis in the workflow model is connected to the history, examination and investigation. The clinician will continue asking questions, do examinations and probably order some tests to strengthen his assumption of the diagnosis. The clinician can also set a more specific diagnosis, such as in the asthma guideline (Republic of Kenya 2016), where we categorize in severe, mild and moderate asthma.

Management is how we manage the patient with the given diagnosis. Hospitalization is if we are going to set patient status to inpatient or outpatient. Medication is given to treat the patient or relieve the symptoms. Surgery is not part of the asthma guideline (Republic of Kenya 2016), but it is important to be aware that management procedures needs to be added when making quizzes for other conditions than asthma.

Follow-up will contain questions related how to evaluate the treatment and how to act upon the evaluation. The symptoms of the patient may have worsen, getting better or are unchanged after the treatment, and the clinician needs to act accordingly. The treatment may have been given at the hospital or it may have been something the patient have had to do at home. The patient may also get some instructions from the clinician when there is something he should do on his own.

Figure 4.16: By looking at the workflow- and entity models we can identify the type of questions for each content unit in the knowledge map. The content units are the parent nodes, while the leaves are the type of questions



As the student learns, the questions need to adapt to the student's new level of knowledge. How we do this is by defining levels, where the questions becomes more detailed at higher levels. At level 1 the questions are all about stating facts. In level 2, we create scenarios such that we follow one patient through all of the steps assessment, diagnosis, management and follow-up. The same for level 3, but the detail level will be higher. Typically the student will only be asked for categories of medication in level 2, but in level 3 the students needs to be specific about the names of the medication as well as measurements for both dosages and symptoms. See table **??**.

Level	Assessment	Diagnosis	Management	Follow-up
1	Factual	Factual	Factual	-
2	Scenario	Scenario	Scenario	Scenario
3	Detailed scenario	Detailed scenario	Detailed scenario	Detailed scenario

4.3.7 Unlocking harder levels at a certain category

One of the strengths with Dynamic Content Management is the focus on adaptive learning and flexible learning (Eide, Kristensen, and Lamo 2008). By adaptive learning, we mean that the student can solve problems which are suited for his knowledge level. While flexible learning means that the student can go through the learning material in the way he prefers, as long as the knowledge dependencies are met.

For the adaptability we have already covered how we progress from factual statements to scenarios and the scenarios with a higher ability to make the questions more difficult. For the flexibility we have divided the content into knowledge units: assessment, diagnosis, management and follow-up. It doesn't matter which of assessment, diagnosis and management the student finishes first.

The evaluations in each content unit has passing conditions. As an example; to complete assessment level 1, you need a score of 30 in assessment. These passing conditions are provided by the quiz author. We also recall that to unlock questions, such as management level 2, there are certain prerequisites that need to be fulfilled. In this example, all evaluations in level 1 need to be completed to unlock any level 2 questions. The flexibility is that it doesn't matter the order of level 1 evaluations the student finishes first, as long as all of them are finished.

To avoid that the student gets bored, he will not have to redo an evaluation once the passing condition is met. That means that if the student meets the passing condition of level 1 assessment and diagnosis, but not management, the student will only get questions from management level 1 the next time he takes the quiz. This makes a challenge for the scenarios. All scenario questions must be formed in a way, such that the student doesn't have to remember information from one question to another. All the necessary information should be listed in every scenario question. In that way the student won't miss important information if he completes diagnosis in a run before management and follow-up.

Each evaluation also has a minimum required skill value. If the student gets a lower score than the required minimum skill, the evaluation gets locked and the student needs to complete the evaluation at lower level. An example is that the student plays management level 2. He completes the evaluation with a lower score than the required minimum skill. The student will have to redo management level 1 evaluation to learn the basic skills necessary to play level 2. When this situation happens, the student no longer meets the prerequisite for the other level 2 evaluations. Level 1 management needs to be completed before any level 2 evaluations can be taken.

4.3.8 Visualization of game statistics

The student needs some feedback, where he is in the learning map, how close he is to pass an evaluation, how close he is to progress to the next level and in the case where he gets relegated to an easier difficulty level.

The scores for the evaluations can be shown as bars in a graph, and the passing conditions as a line. When the bars reaches or passes the line, the student knows that the he has met the passing condition for the evaluation at that level. This can also work as a motivation for the student, as he sees that he gets closer and closer to the passing condition as he learns he more and performs better.

To visualizing where the student is in the learning map, has been solved by showing the learning map in almost a table format. Colour combinations shows which evaluations have been unlocked, which are locked and which are the current ones the student plays. Red and green indicates whether a student gets relegated from a level or whether he progresses to the next level.

4.3.9 Automatically generating new questions

One of the main problems with the application is generating entity model instances used for generating questions. To make one scenario with assessment, diagnosis, management and follow-up, we need at least to instances of the entity graph. One to show the status of the patient before the treatment, and one after the treatment. As the treatment needs to be evaluated and the clinician needs to act accordingly. One single entity model can be several hundreds of lines written in JSON, as there are quite many vertices and edges to describe as well as the complexity of the model. This is tedious work, both because of the complexity, the amount of code lines, but the author also needs to put a great emphasize on writing the code correctly. The latter can be solved with an authoring tool.

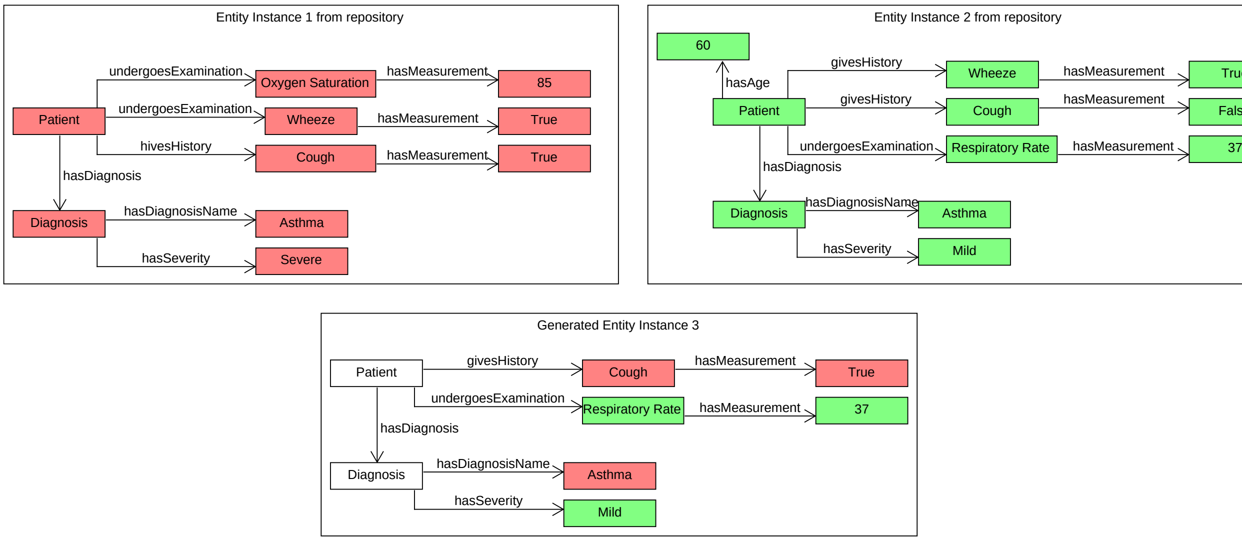
As so much effort has to put in typing these instances of entity graphs, a better way would be to make functionality to automatically generate new instances. We already have made template with tags to vertices in the entity graph. By generating a lot of new entity instances, one template can be reused to make just as many questions. By looking at the paediatric possible asthma guideline Republic of Kenya 2016, we see that there are described twelve symptoms we can combine in every way we like. Some of the symptoms also have more possibilities than just true/false. They have numerical ranges like pulse rate, breaths per minute and oxygen saturation. So in fact, we can generate a quite large amount of graph instances and equally as many questions per template.

The method we suggest for automatically generating such graph instances, is by building up a repository of graph instances. These graph instances are written in the tedious way by a content author. An algorithm picks a set of graph instances from the repository, and randomly chooses vertices from each instance graph. These vertices will be combined to form a new instance graph. This is illustrated in figure 4.17, where the bottom instance is generated from two upper ones. The colour of the vertices indicate which graph instance they come from.

For those who are familiar with the paediatric possible asthma guideline Republic of Kenya 2016, will know that wheeze has to be present in an asthma patient. Wheeze is not present in the generated graph, but still claims that the patient has mild asthma. This indicate the problem with automatically generated graphs without adding any domain knowledge about which vertices to pick and combine. For this reason we need a domain expert to validate which of the generated graphs we can use, and those who are invalid patient representations. Even though this method relies on a domain expert, it will save us from a ton of work. Instead of carefully write hundreds of JSON formatted lines for each graph instance, we can simply approve or disapprove the graph instances which are valid or invalid.

The same method can be used for generating distractions for quiz elements. Randomly choose distractions from other entity graph instances, which corresponds to the answer key tag given by the template. A small penalty can be set on distractions from the asthma guideline. Medium penalties for distractions from respiratory diseases. A bit larger penalties for distractions from completly unrelated guidelines. Then we have automated the quiz scores as well!

Figure 4.17: The red and green graph instances at the bottom are picked from a repository. We randomly pick vertices from each entity instance to make the new entity instance at the bottom. Red vertices indicate that they come from the red entity graph instance. The green vertices that they come from the green entity graph instance.



4.4 The mobile application

- React
- React-Native
- React-Native-Navigation (Wix)
- Redux
- React-Redux
- Redux-Thunk
- Highcharts
- Jest

4.4.1 React-Native and Redux

4.4.2 User interface and flow of the user interaction

4.5 Architecture of the whole system

4.5.1 Visualization

4.6 Evaluation

Chapter 5

Discussion

5.1 Research questions

5.2 Limitations of the model

- Can't ask questions like "what are the symptoms for severe asthma?"

- Difficult to ask what NOT to do. If the vertex doesn't exist, only an empty string gets returned. Can only be used were we actually have written "don't admit to the hospital" as an example with hospitalization.

- The inheritance makes it difficult to generalize some questions. We can't make a template which asks about the Rate a medicine should be taken with. We need to specifically ask for that medicine. To be able to ask for a general medicine, one solution can be to introduce a new tag which compares the substring of the type of the vertex. Another solution is to use the meta model and not the instance model. We don't use inheritance on diagnosis because of this.

- To avoid the problem described in the previous point, we don't use inheritance on Diagnosis. A limitation here is that a patient can only have one diagnosis.

5.3 Observations

5.4 Challenges

5.5 Reflection

Chapter 6

Conclusions

6.1 Further research and development

Bibliography

Attali, Yigal (2015). "Effects of multiple-try feedback and question type during mathematics problem solving on performance in similar problems". In: *Computers & Education* 86, pp. 260–267. ISSN: 0360-1315. DOI: 10.1016/J.COMPEDU.2015.08.011. URL: https://www.sciencedirect.com/science/article/pii/S0360131515300312.

Bergman, Michael K. (2018). "Keeping the Design Open". In: *A Knowledge Representation Practionary*. Cham: Springer International Publishing, pp. 183–205. DOI: 10.1007/978-3-319-98092-8_9. URL: http://link.springer.com/10.1007/978-3-319-98092-8{_}9.

Boxwala, Aziz A et al. (2004). "GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines". In: *Journal of Biomedical Informatics* 37.3, pp. 147–161. ISSN: 1532-0464. DOI: 10.1016/J.JBI.2004.04.002. URL: https://www.sciencedirect.com/science/article/pii/S1532046404000334.

Cabana, Michael D. et al. (1999). "Why Don't Physicians Follow Clinical Practice Guidelines?" In: *JAMA* 282.15, p. 1458. ISSN: 0098-7484. DOI: 10.1001/jama.282.15.1458. URL: http://jama.jamanetwork.com/article.aspx?doi=10.1001/jama.282.15.1458.

Clariana, Roy B. and Ravinder Koul (2006). "The effects of different forms of feedback on fuzzy and verbatim memory of science principles". In: *British Journal of Educational Psychology* 76.2, pp. 259–270. ISSN: 00070998. DOI: 10.1348/000709905X39134. URL: http://doi.wiley.com/10.1348/000709905X39134.

Clayton, P D and G Hripcsak (1995). "Decision support in healthcare." In: *International journal of bio-medical computing* 39.1, pp. 59–66. ISSN: 0020-7101. DOI: 10.1016/0020-7101(94)01080-K. URL: http://www.ncbi.nlm.nih.gov/pubmed/7601543.

De Clercq, Paul, Katharina Kaiser, and Arie Hasman (2008). "Computer-Interpretable Guideline formalisms." In: *Studies in health technology and informatics* 139, pp. 22–43. ISSN: 0926-9630. URL: https://www.researchgate.net/publication/23271685{_}From{_}guidelines{_}to{_}careflows{_}Modelling{_}and{_}supporting{_}complex{_}clinical{_}processes.

Disease, Division of Leprosy Tuberculosis et al. (2011). *Guidelines for Asthma Management in Kenya*. Division of Leprosy, Tuberculosis and Lung Disease. URL: http://apps.who.int/medicinedocs/documents/s21973en/s21973en.pdf.

Eide, Sigvat, Terje Kristensen, and Yngve Lamo (2008). "A model for dynamic content based e-learning systems". In: *Proceedings of the 2008 Euro American Conference on Telematics and Information Systems - EATIS '08*. New York, New York, USA: ACM Press, pp. 1–8. ISBN: 9781595939883. DOI: 10.1145/1621087.1621089. URL: http://portal.acm.org/citation.cfm?doid=1621087.1621089.

Fervers, B., J. Carretier, and A. Bataillard (2010). "Clinical practice guidelines". In: *Journal of Visceral Surgery* 147.6, e341–e349. ISSN: 1878-7886. DOI: 10.1016/J.JVISCSURG.2010.10.010. URL: https://www.sciencedirect.com/science/article/pii/S1878788610001542?via{\%}3Dihub.

Gamma, Erich et al. (1994). "Design patterns : elements of reusable object-oriented software". In: Pearson Education. Chap. Structural, p. 185. ISBN: 0321700694. URL: https://books.google.no/books/about/Design{_}Patterns.html?id=6oHuKQe3TjQC{\&}source=kp{_}cover{\&}redir{_}esc=y.

Guyatt, Gordon H et al. (2008). "GRADE: an emerging consensus on rating quality of evidence and strength of recommendations." In: *BMJ (Clinical research ed.)* 336.7650, pp. 924–6. ISSN: 1756-1833. DOI: 10.1136/bmj.39489.470347.AD. URL: http://www.ncbi.nlm.nih.gov/pubmed/18436948http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2335261.

Johansen, Ingrid H et al. (2018). *Astma - Obstruktiv lungesykdom - Nedre luftveier og lunger - Legevakthåndboken*. URL: https://www.lvh.no/symptomer{_}og{_}sykdommer/nedre{_}luftveier{_}og{_}lunger/obstruktiv{_}lungesykdom/astma (visited on 05/08/2019).

Kristensen, T. (2011). "The Dynamic Content Management system". In: *2011 International Conference on Information Technology Based Higher Education and Training*. IEEE, pp. 1–8. ISBN: 978-1-4577-1673-7. DOI: 10.1109/ITHET.2011.6018677. URL: http://ieeexplore.ieee.org/document/6018677/.

Kristensen, T., Ø. Bech, and M. Dyngeland (2013). "Towards a Multi-Agent E-learning Platform". In: *Journal of Computer Engineering and Informatics* 1.2, pp. 64–81. ISSN: 23070072. DOI: 10.5963/JCEI0102004. URL: http://www.academicpub.org/DownLoadPaper.aspx?paperid=5471.

Masic, Izet, Milan Miokovic, and Belma Muhamedagic (2008). "Evidence based medicine - new approaches and challenges." In: *Acta informatica medica : AIM : journal of the Society for Medical Informatics of Bosnia & Herzegovina : casopis Drustva za medicinsku informatiku BiH* 16.4, pp. 219–25. ISSN: 0353-8109. DOI: 10.5455/aim.2008.16.219-225. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3789163/.

Morrison, Gary R. et al. (1995). "The Effects of Feedback and Incentives on Achievement in Computer-Based Instruction". In: *Contemporary Educational Psychology* 20.1, pp. 32–50. ISSN: 0361-476X. DOI: 10.1006/CEPS.1995.1002. URL: https://www.sciencedirect.com/science/article/pii/S0361476X85710028.

National Heart Lung and Blood Institute and U.S. Department of Health & Human Services (2007). "Guidelines for the Diagnosis and Management of Asthma (EPR-3) — National Heart, Lung, and Blood Institute (NHLBI)". In: URL: https://www.nhlbi.nih.gov/health-topics/guidelines-for-diagnosis-management-of-asthma.

Patel-Schneider, Peter F. and Ian Horrocks (2006). "A comparison of two modelling paradigms in the Semantic Web". In: *Proceedings of the 15th international conference on World Wide Web - WWW '06*. New York, New York, USA: ACM Press, p. 3. ISBN: 1595933239. DOI: 10.1145/1135777.1135784. URL: http://portal.acm.org/citation.cfm?doid=1135777.1135784.

Peleg, M et al. (2000). "GLIF3: the evolution of a guideline representation format." In: *Proceedings. AMIA Symposium*, pp. 645–9. ISSN: 1531-605X. URL: http://www.ncbi.nlm.nih.gov/pubmed/11079963http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2243832.

Preece, Jennifer, Helen Sharp, and Yvonne Rogers (2015). *Interaction Design: beyond human-computer interaction*. 4th. Wiley, pp. 196, 224.

Republic of Kenya, Ministry of Health (2016). *Basic Paediatric Protocols for ages up to 5 years*. 4th. Republic of Kenya, Ministry of Health, p. 32.

Sadowska, Małgorzata and Zbigniew Huzar (2019). "Representation of UML Class Diagrams in OWL 2 on the Background of Domain Ontologies". In: *e-Informatica Software Engineering Journal* 13, p. 89. URL: http://www.e-informatyka.pl/attach/e-Informatica{_}-{_}Volume{_}13/eInformatica2019Art03.pdf.

Shortliffe, Edward H. (1998). "The Evolution of Health-Care Records in the Era of the Internet". In: URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.2653.

The Aasgard Project (2006). *AsbruView*. URL: http://www.asgaard.tuwien.ac.at/asbruview/ (visited on 05/09/2019).

(US), Institute of Medicine et al. (2011). *Clinical Practice Guidelines We Can Trust*. National Academies Press (US), abstract. ISBN: 9780309164221. DOI: 10.17226/13058. URL: http://www.ncbi.nlm.nih.gov/pubmed/24983061.

Woolf, S H et al. (1999). "Clinical guidelines: potential benefits, limitations, and harms of clinical guidelines." In: *BMJ (Clinical research ed.)* 318.7182, pp. 527–30. ISSN: 0959-8138. URL: http://www.ncbi.nlm.nih.gov/pubmed/10024268http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1114973.

Comparison of CPG applications

This appendix shows two comparative tables of applications and websites that display clinical guidelines; one for international applications and one for Norwegian applications. The selection is based on a few articles concerning apps for clinical practice and a search for clinical guideline applications on Google Play. Helsebiblioteket provides access for Norwegian health workers and students to some of the applications and websites below [1], but users do not explicitly need accounts for this.

Table A1: Comparison of international guideline applications

Name	**Type**	**Platform(s)**	**Account needed**	**Account type(s)**	**Content**	**Interaction**	**Remarks**
Guidelines [2]	Mobile app	Android	Yes	Paid	Guidelines as text documents and flowcharts; notes	Flowcharts require separate PDF reader	Free in UK
AgileMD [3]	Mobile app	Android, iOS	Yes	Paid	Guidelines as lists and flowcharts; notes		Customizable for each organization
UpToDate [4]	Mobile app, website	Android, iOS	Yes	Paid	Guidelines as text documents; calculators	Can view outline of documents	
Medscape [5]	Mobile app	Android, iOS	Yes	Free	Guidelines as text documents; drug interaction checker	Can view outline of documents	Focus on medicine; allows resizing of text in app
PEPID [6]	Mobile app, computer app	Android, iOS, Windows Phone, Windows 10	Yes	Paid (free trial)	Protocols as text documents; drug interaction checker; notes		Focus on drugs and pills
Epocrates [7]	Mobile app	Android, iOS	Yes	Free, paid	Guidelines as text documents; drug interaction checker	Each guideline organizes its contents in an accordion	Unlocked features might offer more interactivity
Essential Medical Guidance (EMGuidance) [8]	Mobile app	Android, iOS	Yes	Free	Guidelines as text documents (some with flowcharts)	Limited through forms	Currently focused on South Africa

Table A1 (cont.)

Name	Type	Platform(s)	Account needed	Account type(s)	Content	Interaction	Remarks
MSF Medical Guidelines [9]	Mobile app	Android, iOS, Windows Phone	No	Free	Store medical guidelines as books; list of recent guidelines		Content can be accessed with low quality for weak connections
NICE Guidance [10]	Mobile app, website	Android, iOS	No	Free	Guidelines as text documents; bookmarks	Each guideline starts with an (blocking) outline containing links to its sections	
RCH Clinical Practice Guidelines [11]	Mobile app	Android, iOS	No	Free	Guidelines as text documents		
ACP Clinical Guidelines [12]	Mobile app, website	Android, iOS	No	Free	Guidelines as text documents (some with flowcharts)		Each guideline contain a link that brings you to the corresponding webpage
ALS Handbook [13]	Mobile app	Android, iOS, Windows Phone	No	Free	Guidelines as text documents		Focus on so-called checklists; dark and visually different style compared to others
Guideline Central [14]	Mobile app	Android, iOS	No	Free, paid	Guidelines as text documents		Some features require an account
BC Guidelines [15] [16]	Mobile app, website	Android, iOS	No	Free	Guidelines as text documents (some with flow charts)	Figures can be scrolled horizontally	Outline of guidelines can be accessed through menu button

Table A1 (cont.)

Name	Type	Platform(s)	Account needed	Account type(s)	Content	Interaction	Remarks
ACC Guideline Clinical App [17]	Mobile app	Android, iOS	No	Free	Guidelines as interactive and static flowcharts; calculators; bookmarks; notes	Clicking on answer buttons reveal further instructions	Uses breadcrumb elements that, instead of being on a line, are stacked on top of each other
ESC Pocket Guidelines [18]	Mobile app	Android, iOS	Yes	Free	Guidelines as interactive and static flowcharts	Clicking on answer buttons reveal further instructions	Uses breadcrumb elements that, instead of being on a line, are stacked on top of each other
ESMO Interactive Guidelines [19]	Mobile app	Android, iOS	No	Free	Guidelines as interactive and static flowcharts	Clicking on answer buttons reveal further instructions	Uses breadcrumb elements that, instead of being on a line, are stacked on top of each other
Walsall Join COPD Interactive Guidelines [20]	Mobile app	Android	No	Free	Guidelines as interactive flowcharts	Clicking on answer buttons reveal further instructions	Elements are very separated on tablets
Skyscape [21]	Mobile app	Android, iOS	Yes	Free, paid	Guidelines as text documents (some with flowcharts); instructions for nurses and a quiz with scoring system		Some content requires purchases
Doctor Diagnose Symptoms Check [22]	Mobile app	Android	No	Free	Interactive diagnosis systems for people to diagnose themselves	Clicking on answer buttons reveal further instructions	
BMJ Best Practice [23]	Mobile app, website	Android, iOS	Yes	Paid (free trial)	Guidelines as text documents	Some elements can be expanded	Uses breadcrumb elements; images are separated from the documents
Daily Anatomy [24]	Mobile app	Android, iOS	No	Free	Flashcard quizzes to learn anatomy	Clicking on answer alternatives within a time limit	Displays user statistics and uses learning strategies
LIFE: Neonatal Resuscitatotion Training (ETAT+NR) [25]	Mobile app	Android	No	Free	Visualization of clinical tasks with quizzes	First person 3D game. Quizzes and the use of game elements	Multiple try with hints. Doesn't let the user jump over questions

Table A2: Comparison of Norwegian guideline applications

Name	Type	Platform(s)	Account needed	Account type(s)	Content	Interaction	Remarks
Norsk Elektronisk Legehåndbok (NEL) [26]	Mobile app, website	Android, iOS	Yes	Paid	Guidelines as text documents	Some parts are organized in an accordion	Website has something about clinical procedures
LegeAppen [27]	Mobile app	Android	No	Free	Guidelines as text documents; links to other Norwegian health apps and websites	Uses Material elements such as cards and floating action buttons	Open source
Legevakthånd-boken [28]	Mobile app, website	Android, iOS	No	Free	Guidelines as text documents	Menus with tabs and tables	Available in Norway only
Sykepleiehånd-boka (SHB) [29]	Mobile app	Android, iOS	Yes	Paid	Guidelines as pictures, videos and text documents	Uses two tabs on each page to provide contextual information	
eHåndboken [30]	Mobile app, website	Android, iOS	No	Free	Guidelines as text documents; bookmarks	Can filter offline and bookmarked documents on the same page	Few flow charts
Tiltaksboka [31]	Mobile app, website	Android, iOS	No	Free	Guidelines as text documents (some with flowcharts)	Uses expandable menu lists	
SykepleiePluss [32]	Mobile app, website	Android, iOS	Yes	Paid	Lecture videos; quizzes; calculators	Feedback to quiz questions are given immediately after answering	Tailored towards medical students; no guidelines

Table A2 (cont.)

Name	Type	Platform(s)	Account needed	Account type(s)	Content	Interaction	Remarks
Akuttveileder i pediatri [33]	Mobile app, website	Android, iOS	No	Free	Guidelines as text documents		
Felleskatalogen [34]	Mobile app, website	Android, iOS, Windows Phone	No	Free	Mostly documents and info about medical products and pharmacies	Horizontally scrollable icon list on the main page	Not easy to see that the icon list is scrollable
Veileder i akuttpsykiatri [35]	Mobile app, website	Android, iOS, Windows Phone	No	Free	Guidelines as text documents	Guidelines are organized into an accordion	Feedback bar always visible; reduces area for content
Trygfonden Hjertestart [36]	Mobile app	Android, iOS	No	Free	Guidelines as text documents and pictures; map of heart starters		Focused on Denmark
Redd Liv [37]	Mobile app	Android, iOS	No	Free	Instructions for CPR	Clicking on answer buttons reveal further instructions	Huge buttons
Helsedirektoratet [38]	Website		No	Free	Guidelines as text documents	Uses accordion elements	Guideline header is a bit buggy

References

[1] http://www.helsebiblioteket.no/om-oss

[2] https://play.google.com/store/apps/details?id=uk.co.isai.android.mgp.guidelines

[3] https://www.agilemd.com/download-organizational-app

[4] https://www.uptodate.com

[5] https://www.medscape.com

[6] http://www.pepid.com

[7] http://www.epocrates.com

[8] https://emguidance.com

[9] https://medicalguidelines.msf.org

[10] https://www.nice.org.uk/guidance

[11] https://www.rch.org.au/clinicalguide

[12] https://www.acponline.org/clinical-information/guidelines/mobile-clinical-guidelines

[13] https://apps.appmachine.com/alshandbook

[14] https://www.guidelinecentral.com

[15] http://bcguidelinesapp.ca

[16] http://www.bcguidelines.ca

[17] http://www.acc.org/tools-and-practice-support/mobile-resources/features/guideline-clinical-app

[18] https://www.escardio.org/Guidelines/Clinical-Practice-Guidelines/Guidelines-derivative-products/ESC-Mobile-Pocket-Guidelines

[19] http://www.esmo.org/Guidelines/Pocket-Guidelines-Mobile-App

[20] https://play.google.com/store/apps/details?id=com.walsall.copd

[21] https://www.skyscape.com/sml/

[22] https://play.google.com/store/apps/details?id=com.appcolliders.doctordiagnose

[23] http://bestpractice.bmj.com/info/app/

[24] http://www.dailyanatomy.com/

[25] https://oxlifeproject.org/

[26] https://legehandboka.no

[27] https://olejon.github.io/mdapp/

[28] http://www.lvh.no

[29] https://www.shb.no/info

[30] http://ehandboken.ous-hf.no

[31] http://tiltaksboka.no

[32] https://sykepleiepluss.no

[33] http://www.helsebiblioteket.no/retningslinjer/akuttveileder-i-pediatri/forside

[34] https://www.felleskatalogen.no/medisin/smarttelefoner-og-nettbrett

[35] https://sykehuset-ostfold.no/helsefaglig/fagprosedyrer/veileder-i-akuttpsykiatri-vap

[36] https://hjertestarter.dk

[37] https://www.lhl.no/lar-a-redde-liv-med-gratis-app/

[38] https://helsedirektoratet.no