Filtering Clinical Guideline Interactions with Pre-conditions: A Case study on Diabetes Guideline*

Veruska Zamborlini
Institute for Logic, Lang. and Computation
Universiteit van Amsterdam
v.zamborlini@uva.nl

Roelof van der Heijden
Graduate School of Informatics
Universiteit van Amsterdam

Annette ten Teije
Computer Science Department
Vrije Universiteit Amsterdam
annette.ten.teije@vu.nl

Abstract

Clinical Guidelines are meant to support healthcare providers to offer a better service via evidence-based recommendations that apply according to certain circumstances given a certain disease or condition. However, the high number of recommendations in a single guideline makes it humanly impossible to verify for all possible interactions. The goal of this work is twofold: (i) to analyse pros and cons of formalising a real guideline using the TMR model and then (ii) to infer interaction among (some of) the recommendations from the the Scottish Guideline on Diabetes. To this end we extend the TMR Model to formalize the pre-conditions that define in which circumstances a recommendations may apply and we implemented the reasoning in SWI-Prolog. The results show that (i) properly formalising the diabetes guideline is a cross-disciplinary task that requires both the formalisation know-how and the medical background; and (ii) indeed the diabetes guideline presents conflicting recommendations which can be automatically detected provided the suitable modeling and background knowledge. It is reasonable to conclude that these conclusions hold for other guidelines too.

Introduction

Clinical guidelines are a standard in the medical field. There is a separate clinical guideline available for almost every major disease. They serve as a framework for the treating doctor, to put his actions in perspective and point him in new directions. Of course, they are only guidelines and doctors can deviate from them when they deem it necessary when creating a treatment plan for a patient.

Guidelines are often long and complex documents, therefore tools have been developed to help with this developing process of clinical guidelines. One of these tools is the modeling language Asbru (Shahar, Miksch, and Johnson 1998), which allows practitioners to precisely formulate each step in a treatment plan and the conditions that need to be satisfied in order to perform the next step. One feature that Asbru does not support however, is conflict detection between recommendations in the guideline. When a patient suffers from multiple diseases, multiple guidelines might apply, whose recommendations might contradict or in other ways interact with each other. It might even be the case that within a single guideline, some recommendations interact with some other recommendations. These interactions might result in errors in the treatment of patients.

The TMR model (Transition-based Medical Recommendation), designed by Zamborlini et al. (Zamborlini et al. 2014) is another approach to describe clinical guideline recommendations. It is designed to detect conflicts between recommendations. It has been used in this regard for a constructed set of recommendations already. However, the TMR model is also limited, for example in its ability to handle preconditions of recommendations. This research tests the expressiveness of the TMR model for formalizing recommendations from a real life clinical guideline and detecting possible interactions among the guideline recommendations. The proposed case study uses the Scottish diabetes guideline SIGN116 (Network 2013). Furthermore, we extend the TMR model and implement some logic to allow the TMR model to handle preconditions.

The adopted methodology comprises of three steps further described in the next section. The first step consists of verifying if the information available in guidelines is sufficient to allow formal representation and detecting interactions in the TMR model. Is the help of a medical expert required? Are the recommendations clear and unambiguous? The next step is to look into the limitations of the TMR model. Which recommendations can be described using the TMR model and what choices need to be made in doing so? Are there recommendations that can not be described by the TMR model and what extensions would be required to be able to? The third step consists of improving one particular aspect of the TMR model, namely the representation of pre-conditions. How can we implement them and what new insights does this bring to our previous results? Does the handling of pre-conditions improve the detection of true interactions?

In the sequel we present the core concepts of the TMR model and how they are used to detect interactions, followed by the proposed extension to use pre-conditions for identifying what interactions can avoided. Then we present the experiment of formalising the Diabetes Guideline in TMR and detecting interactions. Finally we discuss some related work and present the conclusions of this work.

*This paper is based on the master thesis of van der Heijden, available in https://tinyurl.com/heijdenMScThesis

Methodology

This section describes the steps of the methodology, results and contributions as presented in (van der Heijden 2017).

Step 1: Guideline selection

To test the abilities of the TMR model to describe recommendations from real life guidelines, we first chose a real life guideline. We were looking for a guideline that based its statement on verifiable scientific research. It also needed to be written in English, so that the international scientific community can take part in the discussion. It would have our preference if the guideline was also recently updated. The chosen guideline was the Scottish diabetes guideline SIGN116 (Network 2013). Next, we went through the guideline and identified several recommendations that could possibly interact with each other. Only if we know that interactions are supposed to exist between the recommendations, can we verify any results the TMR model might give us.

After that, we examined the recommendations closely, to determine if there is any missing or ambiguous data in the selected set. It is important to note these problems already, for it might influence the choices we will make during the modeling. Furthermore, clinical guidelines are written with a target audience of medical experts in mind. We need to determine if we are able to understand the guideline before starting to model the recommendations.

Results Selection of recommendations, their analysis and expected interactions.

Contribution Discussion about the ambiguities and missing information found in the guideline.

Step 2: Modeling and interaction detection

The next step was to model the recommendations in TMR. Any problems we encountered or decisions we had to make are discussed in this section. With our models of the recommendation in hand, we executed the algorithm to detect the interactions. We compared these results with our initial expectations and analysed any differences.

Results Recommendations modeled in TMR, calculated interactions.

Contribution Discussion about (i) the benefits and difficulties faced while modeling, (ii) analysis of the computed interactions compared against the expected ones, and (iii) how could the TMR model be improved.

Step 3: Extension of TMR model

Subsequently, we extend the TMR model to allow for the handling of preconditions. To do this we have identified a common structure of preconditions and designed an implementation for it. Using these observations, we performed the actual encoding of the preconditions. We ran the detection algorithm with the extended TMR models and discuss the results compared to our manual expectations as well as the previous results to determine whether we have been successful in extending the TMR model.

Results The TMR model extended with preconditions, new calculated interactions.

Contribution Discussion about the proposed solution, advantages & limitations.

TMR Extension

TMR Model - Background

In this section we present the core concepts of the TMR model in order to provide the necessary information to understand our modeling decisions (Zamborlini et al. 2014; Zamborlini et al. 2017a). Besides actions and recommendations, the TMR-model accounts for transitions and the reasons for performing an action, the so called causation beliefs. An UML class diagram of the TMR model can be found in Figure 1 and its components.

Each component is described as follows and illustrated using the following recommendation taken from the Diabetes Guideline: "*People with type 1 diabetes* should be encouraged to participate in *physical activity or structured exercise* to improve *cardiovascular risk factors*". Its representation according to the TMR model is illustrated in Figure 2.

Clinical Guideline Each recommendation is part of a clinical guideline object. In our case study, since all our recommendations come from the same guideline, we only have to define a single guideline object for diabetes and can reference it in each recommendation.

Recommendation Each recommendation references an action and a causation belief that justifies the recommendation, besides specifies its strength and which clinical guideline it is a part of. In Figure 2 the recommendation is labeled as "Type 1 should exercise regularly" and recommends that an action *should* be performed for a reason.

Action Types An action object can represent the administration of drugs, but can also be some other medical action. In our example the action describes exercising, so we label it as "Perform exercise."

Transition Types A transition has two objects associated with it: the transformable situation and the expected situation. These describe the beginning and ending state of a transition respectively. If we perform exercise, we reduce our risk of cardiovascular diseases. So in our example the transformable situation is "increased cardiovascular risk factors" and the expected situation is "reduced cardiovascular risk factors."

Situation types Both transformable and expected situations are instances of situation types. They are used to describe states of the patient. It allows for example to describe when two actions promote the same transition or as well the opposite effect of each other. In a situation we always talk about a specific property that is some value, for example "cardiovascular risk factor is reduced."

Causation Belief A causation belief object is used to describe what transition is caused by a specific action. It answers the question: why do we perform this action? For example, we perform exercise to reduce the cardiovascular risk factors. Each causation belief has a probability and a strength, which is defined by the quality of the evidence of the recommendation. [1]

[1]The probability information as well as the strength are currently not used by the implementation of the TMR model to com-

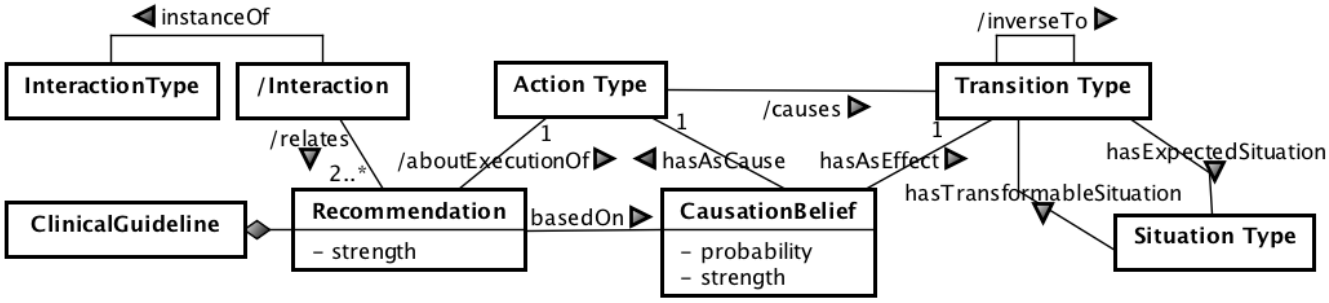


Figure 1: UML class diagram of the TMR model.

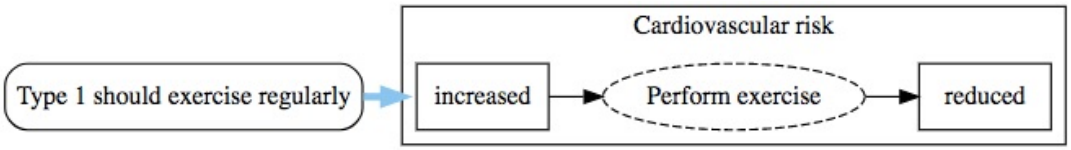


Figure 2: Example of instance of the TMR model.

Interaction Types An interaction relates two recommendations. Each interaction is of a specific type, e.g. contradiction or repetition.

One of the goals of the TMR model is to allow for detection of interactions among recommendations, as described in the next section.

Detection of several interaction types

In the current implementation of the TMR model, four types of interactions that can be detected. These are: *contradiction, repetition, alternative actions and repairable transitions*. In particular, some interactions may be defined in terms of counteracting effects, which requires us to define also inverse transitions. Other types of interactions, such as interactions in time or dosage are not yet supported. Hereby we provide their informal descriptions based on the formal definitions provided in (Zamborlini et al. 2017a).

Definition 1 (Inverse) *Two transitions T_1 and T_1 are inverse if the expected end state of T_1 is the initial state of T_2 and the initial state of T_1 is the expected end state of T_2.*

Definition 2 (Contradiction) *Two recommendations R_1, R_2 are considered contradicting if one of the following is true:*

- *Recommendation R_1 recommends action A_1, whereas R_2 recommends not performing A_1.*
- *Recommendation R_1 recommends action A_1 to achieve transition T, whereas R_2 recommends performing action A_2 in order to prevent T from occurring.*
- *The recommendations R_1, R_2 recommend actions A_1, A_2 that promote transitions T_1, T_2 that are inverse to each other.*

pute interactions. Therefore, we have used a probability value of "always" for all our causation beliefs.

Definition 3 (Repeating) *Recommendations $R_1, R_2, ..., R_n$ are considered repeating when they all recommend the action A.*

Definition 4 (Alternative) *Recommendations $R_1, R_2, ..., R_n$ are considered alternatives to each other when their causation beliefs C_1, C_2 reference the same transition T, while the actions A_1, A_2 are different.*

Definition 5 (Repairable) *Two recommendations are considered repairable when one recommends and the other does not recommend actions whose transitions are inverse.*

A recommendation is considered repairable when it is meant to prevent a transition while another recommendation recommends an action that promotes an inverse transition, i.e. it indicates that the undesired effect could be repaired.

All those rules are implemented in SWI-Prolog and the interactions are automatically calculated. The implementation of these rules is discussed in details in (Zamborlini et al. 2017b), where they are generically applied to (parts of) Hypertension, Osteoarthritis and Diabetes II guideline also using existing medical knowledge available as Linked Open Data.

Pre-Condition Extension and Implementation

One of our contribution of this paper is the extension of TMR model with pre-conditions, i.e. the conditions in which a recommendation would apply (van der Heijden 2017). Our extension is described in UML as pictured in Figure 3. It comprises two modifications: (i) introduction of a relation *hasFilterSituation* between the Recommendation and a Situation Type; and (ii) specialization of the class *Situation Type* into two sub-classes: *Atomic and Complex Situation Types*. The latter is a combination of other situation types using one of the logic operators: *and*, *or* and *negation (neg)*.

Since the TMR model is implemented in RDF format, it requires to represent the preconditions in RDF as well. This

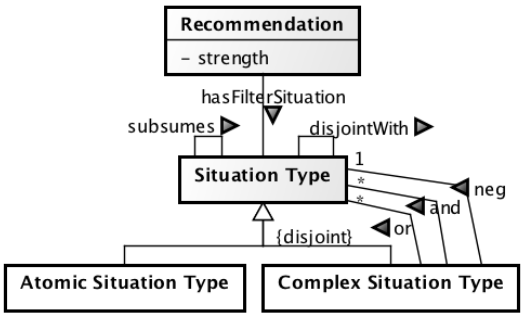


Figure 3: UML class diagram of the extension to the TMR model.

is not straightforward since a statement in RDF is a triple that consists of a subject, predicate (verb) and object. Each triple can be viewed as a directed arrow from the subject to the object, that is annotated with the verb. A formula in predicate logic consists of boolean operators, predicates and literals. Each operator takes two (or one, in case of negation) objects, applies some operation on them and produces a result. We need a way to represent the one using the other.

Our representation uses intermediate objects to represent the logic formulas. Let's say we want to represent the following formula in RDF.

$$(A \vee B \vee C) \wedge \neg B \wedge (\neg A \vee C) \tag{1}$$

Here we are using the n-ary extensions of the binary boolean operators. Using prescript notation this can be written as

$$\wedge(\vee(A,B,C),\neg B,\vee(\neg A,C)). \tag{2}$$

We can write this formula as a directed graph where each node is either a literal or an operator. If an edge points from an operator to a node, then it means that this node is an argument for this operator. The graph corresponding to the formula of Equation 2 can be found in Figure 4a.

Figure 4b pictures the graph representation of the RDF file that encodes this same formula. Here, we use boxed nodes to denote intermediate objects. Each arrow is labeled with its operation.

In RDF, let's first define the literals. The only requirement we have for literals is that we want to be able to refer to them. So we only need to define a subject and type. We also give them a label. Listing 1 shows how we can do that.

```
:A      a               tmr:SituationType;
        rdfs:label      "Situation A" .

:B      a               tmr:SituationType;
        rdfs:label      "Situation B" .

:C      a               tmr:SituationType;
        rdfs:label      "Situation C" .
```

Listing 1: Predicates defined in RDF.

Now we use intermediate objects as subjects and the operators as verbs. The object of each statement is either a predicate, or an intermediate object. For each argument of each operator, a new RDF triple needs to be defined.

```
:orABC  a           tmr:SituationType;
        rdf:or      :A, :B, :C.

:negB   a           tmr:SituationType;
        rdf:neg     :B.

:negA   a           tmr:SituationType;
        rdf:neg     :A.

:orNegAC  a           tmr:SituationType;
          rdf:or      :negA, :C.

:final  a           tmr:SituationType;
        rdf:and     :orABC, :negB, :orNegAC.
```

Listing 2: Intermediate RDF resources to define logic formulas.

The final formula object `:final` is referenced by the recommendation. Using this representation we are able to uniquely describe any logic formula in RDF.

There are a couple things to note here.

- There is nothing preventing a user from defining relations that use different verbs to a single intermediate object, e.g. defining both an `and` relation and an `or` relation for a single intermediate object. However, the resulting structure can not be translated back into predicate logic. This should therefore never be done.
- Because in RDF a subject can be related to any number of objects, we can define multiple relations for a single intermediate object. This allows us to use n-ary boolean operators instead of binary boolean operators.
- We only support the boolean operations conjunction, disjunction and negation. All other operations have to be expressed using these three.

Finally, the reasoning task is implemented in SWI-Prolog[2]. It first extracts Prolog rules from the above described RDF representation. Then it verifies their satisfiability by using the SAT-solver CLP(B) (Triska 2016). In other words, it determines if the conditions imposed by a set of interacting recommendations are satisfiable. If it is not, it means the interaction will never occur since no patient can satisfy all the conditions at the same time.

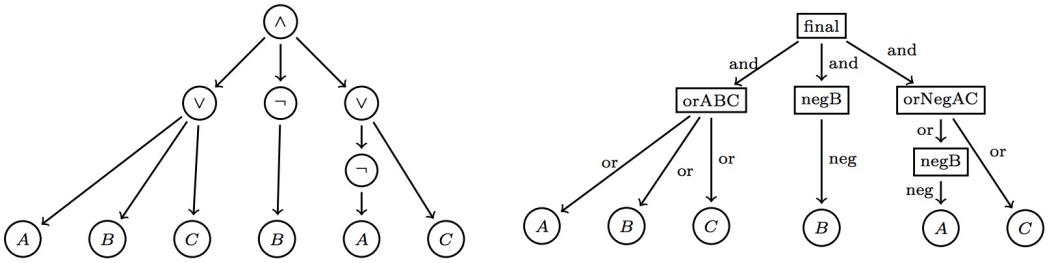
Experiment and Results

Selecting and analysing recommendations

From SIGN116 (Network 2013) 21 recommendations were selected for which interactions were expected to be found. The following list presents a subset of the recommendations selected in (van der Heijden 2017):[3]

[2]The main code can be found in `guidelines-tmr.github.io/tmr-precond-code` and a notebook showing the results can be founf in `guidelines-tmr.github.io/tmr-precond-notebook`

[3]The code associated to each recommendation is kept as in the original work, except for a small correction for B1, B2, D1 and D2.



(a) Graph representation of the formula in Equation 5.

(b) Graph representation of Listing 3.

Figure 4: Different representations of the same formula.

B1 People with type 1 diabetes should be encouraged to participate in physical activity or structured exercise to improve cardiovascular risk factors.

B2 People with type 2 diabetes should be encouraged to participate in physical activity or structured exercise to improve glycaemic control and cardiovascular risk factors.

B3 Patients with existing complications of diabetes should seek medical review before embarking on exercise programmes.

C1 Metformin should be considered as the first line oral treatment option for overweight patients with type 2 diabetes.

C2 Sulphonylureas should be considered as first line oral agents in patients who are not overweight, who are intolerant of, or have contraindications to, metformin.

C3 Pioglitazone can be added to metformin and sulphonylurea therapy, or substituted for either in cases of intolerance.

D1 Examination of the retina prior to conception and during each trimester is advised in women with type 1 and type 2 diabetes. More frequent assessment may be required in those with poor glycaemic control, hypertension or pre-existing retinopathy.

D2 Patients' retinas should be screened at least annually.

F1 Patients with type 1 diabetes should be screened from age 12 years.

F2 Patients with type 2 diabetes should be screened from diagnosis.

G1 Lipid-lowering drug therapy with simvastatin 40 mg should be considered for primary prevention in patients with type 1 diabetes aged $>$ 40 years.

G2 Patients under 40 years with type 1 or type 2 diabetes and other important risk factors, e.g. microalbuminuria, should be considered for primary prevention lipid-lowering drug therapy with simvastatin 40 mg.

We have found issues that make the modeling of recommendations difficult, among them: (i) *ambiguity* is common due to imprecise nature of natural language descriptions, for example the use of 'and' and 'or' often do not correspond to how it would be formalised using the corresponding logical operators; (ii) *implicit knowledge* about causation or preconditions is also common since either it is considered common medical knowledge, or it is omitted by relying on the context of the the document section in which the recommendation appears, or it is provided in an accompanying text which describes the evidence for the recommendation and or often not very clear, specially for non-domain-experts.

For example, consider recommendation C3. It ambiguously suggests to add pioglitazone to metformin AND sulphonylurea therapy. However, it is not clear whether this means adding to both metformin and sulphonylurea, or to either metformin or sulphonylurea therapy. In this case, if we look into C1 and C2 then we can conclude that the latter is correct. Moreover, neither of them clearly mention the effect expected to be achieved by any of the therapies. It actually is meant to lower the average blood glucose level.

Formalizing the Diabetes guideline

The expressiveness of the TMR model can also pose some difficulties in the modeling process: it may require information that is missing in the guideline (see previous section) or it may not offer means to formalise all knowledge within a recommendation, such as time and quantity, since it is a work in progress.

Surpassed those difficulties/limitations, the recommendations are formalized[4] according to the TMR model.

Listing 3 describes the formalisation of the recommendation B1 according to the RDF implementation of the TMR model. Note that lines 23 to 26 describe the TMR extension of pre-conditions as discussed earlier.

Calculating Interactions

The system detects interactions for the all the formalised recommendations, which included all the expected ones but another two that were not foreseen. For the list of recommendations of previous section, Table 1 presents a subset of the detected interactions, without taking into account the pre-conditions.

The interactions marked with * are the ones expected to be avoided by taking the preconditions into account. For example, because B1 and B2 recommend the same action type,

[4] https://github.com/l0ft3r/SIGN116Model

```
# DM Guideline
:CIG-DB     a    tmr:Guideline;
  rdfs:label   "CIG for Diabetes Mellitus"@en .

# Recommendation B1
:RecDB-ExercB1   a   tmr:ClinicalRecommendation;
  rdfs:label
        "Type 1 should exercise regularly"@en ;
  tmr:partOf      :CIG-DB ;
  tmr:basedOn     :CBExercise1 ;
  tmr:strength    "should"^^xsd:string ;
  tmr:aboutExecutionOf    :ActExercise ;
  tmr:hasFilterSituation :PredDM1   .

# Causation Belief
:CBExercise1 # named graph with a causation
{ :CBExercise1   a     tmr:CausationBelief ;
    tmr:probability    "always"^^xsd:string ;
    tmr:strength     "L2"^^xsd:string .
  :ActExercise
    tmr:causes      :TrExerciseType1 . }

# Situation Pre-condition
:PredDM1   a   tmr:SituationType,
           rdfs:label      "DM 1"@en .

# Transition
:TrExerciseType1   a   tmr:TransitionType;
  tmr:promotedBy     :ActPerformExercise ;
  tmr:hasTransformableSituation
                  :SitIncrCardiovRisks ;
  tmr:hasExpectedSituation
                  :SitRedCardiovRisks .

# Situation Types pre and pos state
:SitIncrCardiovRisks   a   tmr:SituationType ;
  rdfs:label "Cardiov. risk is increased"@en .
:SitRedCardiovRisks    a   tmr:SituationType ;
  rdfs:label "Cardiov. risk is reduced"@en .

# Action type
:ActExercise    a    tmr:CareActionType;
  rdfs:label      "Perform exercise"@en .
```

Listing 3: Excerpt of the TMR-based representation of the recommendation B1

namely to perform exercises, the system will detect an interaction of type repetition. However, looking into their preconditions, we can expect this interaction not to occur since the first is applicable for patients with diabetes type 1 and the second for diabetes type 2. The same happens for other two pairs of recommendations, namely F1-F2 and G1-G2. By chance they are all involved in interactions of type repetition, but it could have been the case for any other type.

Calculating Avoidable Interactions

In order to re-evaluate the applicability of interactions based on the pre-conditions some background knowledge is also required. For the moment, we have manually added the required background knowledge, but we hope such information can be either extracted from existing medical knowledge bases or ontologies (e.g. a patient can either have diabetes type 1 or type 2) or can be calculated when they are of a numeric nature (a patient cannot be below and above a certain age). For the case study presented in this paper, the following knowledge is added:

- `not(and(DM1, DM2))`
- `not(and(Age40+, Age40-))`

This means the patient cannot be suffering from diabetes type 1 and type 2 at the same time, and also cannot be below and above 40 years old.

By running the satisfiability checker, we found three unsatisfiable cases, presented in Table 2 together with another two satisfiable results out of several. This shows that our implementation can automatically find the groups of interactions that can be avoided. In other words we are able to improve the precision of our interaction detection method by extending the TMR model with pre-conditions.

Recommendations	Detected interaction
* B1–B2	repetition
B1–B3	contradiction
B2–B3	contradiction
C1–C2–C3	alternative
D1a–D1b–D2	repetition
* F1–F2	repetition
* G1–G2	repetition

Table 1: Recommendations with detected interactions.

Rec	Precondition	Sat?
B2	`DM1`	False
B1	`DM2`	
C1	`and(DM2, Overweight)`	True
C2	`or(not(Overweight), IntolerantForMetformin, ContraindForMetformin)`	
D1a	`or(Pregnant, WantsToBecomePregnant)`	True
D2	`or(DM1, DM2)`	
F1	`and(DM1, Age12+)`	False
F2	`DM2`	
G2	`and(or(DM1, DM2), OtherRiskFactors, Age40-)`	False
G1	`and(DM1, Age40+)`	

Table 2: A subset of our findings. For each recommendation pair that make up an interaction, it details their preconditions and the satisfiability of the conjunction of preconditions.

Related Work

Riano and Ortega (Riao and Ortega 2017) have performed a literature survey on computer systems that deal with multimorbidity. They classify the recent research in this area according to several systems, one of them being a classification system developed Abidi et al. (Abidi 2010) and extended by Jafarpour (Jafarpour and Abidi 2013), where the catego-

rization is based upon the combination point of the different guidelines. Five distinct categories are defined: guidelines can be combined and then computerized, the computerized guidelines can be combined, combination can occur of the individual treatment plans, or during the process of computerization. Finally the knowledge from guidelines can be combined based on the stored records of patients that match the multi-morbid criteria. Riano and Ortega identify the work of Zamborlini et al. as belonging to the category where guidelines are computerized and then combined. The extension we describe in this work does not change this categorization.

Riano and Ortega also mention the strengths and weaknesses of the used techniques. They identify "reusable knowledge, conceptual simplicity, decremental costs" as the strengths of transition fitting, the technique used by the TMR model. Decremental costs in this context means that the required effort of adding more guidelines to the system's knowledge base diminishes with each additional guideline, since the concepts shared between guidelines can be reused. The weaknesses are identified as follows: "not completely automatic, premature, only suitable for short-term treatments." The TMR model is indeed underdevelopment and its implementation is a proof of concept rather than a fully functional system ready to be used by the doctors. It is also not completely automatic in the sense that it does not provide a user with a treatment plan for a specific patient. Indeed, this was never part of the design goals of the TMR model: it focuses on *detection* of interactions rather than on conflict *resolution*. The decision is expected to be taken by the medical doctor taking into account the guidelines and the patient. In the future we might consider also the patient data to identify relevant guideline interactions for a particular case.

The study from Peleg et al. (Peleg et al. 2003) gives a comparison between five CIG modeling languages: Asbru, Eon, GLIF, GUIDE, PRODIGY and Proforma. It asked experienced modelers of each language to model two guidelines and compared the resulting models syntactically and semantically. Their aim was to identify common components between the different languages to try and establish some standards, as well as providing a starting point for discussions on comparing CG modeling languages. They have concluded that there are indeed major components that are very similar between the languages. The dimensions that Peleg et al. (Peleg et al. 2003) took into account were: 1) organization of guideline plan components, 2) specification of goals/intentions, 3) model of guideline actions, 4) decision models, 5) expression/criterion language used to specify decision criteria, 6) data interpretations/abstractions, 7) representation of a medical concept model and its use, and 8) patient information model. However they note that these are implemented in different ways because of the different goals of each language. They find that it is important to allow each research group behind a language to pursue their own goals rather than try to constrain them by imposing a standard.

GLARE (Terenziani, Molino, and Torchio 2001) is another structured language for describing clinical guidelines. It consists of two modules: a guideline acquisition module, and a guideline execution module. The acquisition module provides a user friendly interface to load a clinical guideline. While the doctor is entering the guideline, it already detects many forms of semantic or syntactic inconsistencies : name and range checking, to ensure standard nomenclature is being used; logical consistency, to ensure that each set of alternatives is preceded by a decision and each decision is preceded by a data query and to prevent circular dependencies; and temporal consistency, to ensure the entered guideline is still executable. This process ensures only high quality guidelines are entered into the system. However, these consistency checking apply to a single guideline. When multiple guidelines are relevant for a single patient, as is the case for multi-morbid patients, their simultaneous execution could lead to conflicts. Furthermore, as preconditions in GLARE are stored as plain text, it is not possible to perform an automated analysis of excluding preconditions for the calculation of interactions between recommendations.

In (Anselma, Piovesan, and Terenziani 2017), Anselma et al. describe their extensions to the GLARE system as the first to focus on the temporal interactions between actions guideline actions with the intent to detect conflicts that *actually* occur, rather than conflicts that *might* occur when their effects happen to be active simultaneously. They construct a constraint satisfaction problem based on three sources of information: a user-provided log that describes when certain actions have been executed, temporal constraints extracted from the guidelines as they are expressed in GLARE, and information present in their knowledge base. Once solved using a standard STP constraint propagation network framework, they interpret the results and provide the user with "YES," "NO" or "MAYBE" to indicate whether an interaction occurs. Although their work is a nice contribution to conflict detection between clinical guidelines, a user is still required to select the recommendations from the guidelines for the system to analyze. This means the user needs to be aware of possible interactions before using the system. In its turn, the TMR method present these possible interactions to a user, requiring only a selection of clinical guidelines to analyze, rather than the actual recommendations. Both studies use a form of constraint satisfaction programming to support the detection of conflicts. In this paper we use SAT, which is a special kind of constraint satisfaction programming, but different than the one used in (Anselma, Piovesan, and Terenziani 2017), which includes time constraints. This suggests the possibility of combining both works in a similar fashion, i.e. using the initial calculation of interactions to reduce the search space for the satisfiability problem.

Conclusion and Future Work

In our study on guideline interactions in diabetes we observed that there are many ambiguous recommendations, missing information on the motivation of recommendations and implicit information in the preconditions of the recommendations. This indicates that it is very difficult to accurately model these recommendations in a formal language, regardless of which language is chosen. In many cases a medical expert is required to resolve these issues on a case by case basis. Another possible solution, but less realistic in

the current guideline development process, would lie in the migration from a "paper-based" guideline paradigm (recommendation and evidence are mere texts to be interpreted) to a well supported computerized paradigm that would account for those problems at the authoring phase, i.e. making sure the meaning and purpose of recommendations are clear from the beginning.

Overall we were able to model the selected diabetes recommendations in TMR. The concepts in the TMR model match closely with the information present in the recommendations, although there is a need for less strict causal relations, like maybe, possibly and might be that has been addressed conceptually in (Zamborlini et al. 2017a) but not yet implemented.

The interaction detection algorithm computes all possible interactions. With our extension to the TMR model of preconditions, we were able to reduce the number of identified possible interactions. More importantly, the initial calculation of interactions are a mean to reduce the search space for the satisfiability problem between preconditions. In our case-study from 52 interactions to 49 interactions. Further reduction of interaction would be possible by entering some characteristics of the patient, like has type 1 diabetes, then many recommendations and, by extension, many interactions can be ruled out. In in (Zamborlini et al. 2017a) we also propose other way to filter out or rank interactions based on some characteristics of the recommendation, action or transitions involved, e.g. the deontic strength of the recommendations involved.

Finally, an important characteristic of clinical guidelines and of systems that implement them is to provide advice that addresses the general public, which means it supports mainly the majorities, for example, patients that have either diabetes 1 or 2. Although this is an understandable limitation of the "paper-based" clinical guideline system, we believe that computer-based systems should allow for supporting also minorities, such as patients that have both diabetes 1 and 2. Therefore, as future work the system should be flexible enough to account for both situations.

References

Abidi, S. 2010. *A Knowledge Management Framework to Develop, Model, Align and Operationalize Clinical Pathways to Provide Decision Support for Comorbid Diseases*.

Anselma, L.; Piovesan, L.; and Terenziani, P. 2017. Temporal detection and analysis of guideline interactions. *Artificial Intelligence in Medicine*.

Jafarpour, B., and Abidi, S. S. R. 2013. *Merging Disease-Specific Clinical Guidelines to Handle Comorbidities in a Clinical Decision Support Setting*. Berlin, Heidelberg: Springer Berlin Heidelberg. 28–32.

Network, S. I. G. 2013. *Management of Diabetes*. Gyle Square, 1 South Gyle Crescent Edinburgh EH12 9EB: Healthcare Improvement Scotland.

Peleg, M.; Tu, S.; Bury, J.; Ciccarese, P.; Fox, J.; Greenes, R. A.; Hall, R.; Johnson, P. D.; Jones, N.; Kumar, A.; et al. 2003. Comparing computer-interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association* 10(1):52–68.

Riao, D., and Ortega, W. 2017. Computer technologies to integrate medical treatments to manage multimorbidity. *Journal of Biomedical Informatics* 75:1 – 13.

Shahar, Y.; Miksch, S.; and Johnson, P. 1998. The asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial intelligence in medicine* 14(1):29–51.

Terenziani, P.; Molino, G.; and Torchio, M. 2001. A modular approach for representing and executing clinical guidelines. *Artificial Intelligence in Medicine* 23(3):249 – 276.

Triska, M. 2016. The boolean constraint solver of swi-prolog: System description. In *FLOPS*, volume 9613 of *LNCS*, 45–61.

van der Heijden, R. 2017. Modeling a Diabetes Guideline using the TMR model for interaction detection. Technical report, Universiteit van Amsterdam. Master Thesis.

Zamborlini, V.; Hoekstra, R.; da Silveira, M.; Pruski, C.; ten Teije, A.; and van Harmelen, F. 2014. *A conceptual model for detecting interactions among medical recommendations in clinical guidelines: A case-study on multimorbidity*, volume 8876 of *Lecture Notes in Computer Science*. Springer. 591–606.

Zamborlini, V.; da Silveira, M.; Pruski, C.; ten Teije, A.; Geleijn, E.; van der Leeden, M.; Stuiver, M.; and van Harmelen, F. 2017a. Analyzing interactions on combining multiple clinical guidelines. *Artificial Intelligence in Medicine*.

Zamborlini, V.; Hoekstra, R.; da Silveira, M.; Pruski, C.; ten Teije, A.; and van Harmelen, F. 2017b. Generalizing the Detection of Clinical Guideline Interactions Enhanced with LOD. In *Biomedical Engineering Systems and Technologies*. Springer International Publishing. 360–386.