Identifying Inconsistencies in Multiple Clinical Practice Guidelines for a Patient with Co-morbidity

Marisela Mainegra Hing*, Martin Michalowski†, Szymon Wilk‡, Wojtek Michalowski*, and Ken Farion§
*University of Ottawa, Telfer School of Management, Email: wojtek@telfer.uottawa.ca, mhing@uottawa.ca
†Adventium Labs, Minneapolis MN 55401, USA,Email: martin.michalowski@adventiumlabs.org
‡Poznan University of Technology, Institute of Computing Science, Email: szymon.wilk@cs.put.poznan.pl
§Children Hospital of Eastern Ontario, Email: farion@cheo.on.ca

***Abstract*— This paper describes a methodological approach to identifying inconsistencies when reconciling multiple clinical practice guidelines. The need to address these inconsistencies arises when a patient with co-morbidity has to be managed according to different treatment regimens. Starting with a well-known flowchart representation we discuss how to create a formal guideline model that allows for easy manipulations of its components. For this model we present how to identify conflicting actions that are manifested by treatment-treatment and treatment-disease interactions, and how to reconcile these conflicting actions.**

I. Introduction

This paper describes methodological research on developing a guideline model that allows for the semi-automatic adaptation of a guideline to a patient-specific situation characterized by the need to treat multiple conditions present at the same time (so called co-morbidity). The proposed model can be easily implemented as part of a larger clinical decision support system and thus its use is suited for point of care support.

Clinical practice guidelines (CPGs) are important knowledge uptake tools in the practice of evidence-based medicine. They represent systematically developed statements to assist physician decision-making, aiming to improve quality of care and reduce practice variations. However, the use of CPGs at the point of care is limited and a variety of barriers to use have been identified (see [1]) and still exist in practice. Some of the barriers relate to the fact that the guidelines seldom allow for patient-specific advice and moreover are often presented in a way that increases the cognitive burden of decision-making. These two shortcomings can be addressed by customizing the guidelines to a patient encounter and by creating a computer interpretable model of the guideline that can be easily manipulated to meet the physicians' needs.

CPGs can be modeled differently (see [2] for a review), however the majority of modeling approaches rely on a flowchart format that includes patient state steps, decision steps and action steps [3]. These are the most common steps found in the computerized guideline models and they also conform to the proposed standard for medical flowcharts [4]. A patient-state step represents the specific condition of a patient; it includes symptoms, signs and diagnosis. A decision step describes the process of medical decision-making; it involves selection from a set of alternatives. An action step represents a specific clinical task that should or shouldn't be performed on a patient, i.e. a single treatment, intervention or test request.

In this work, we assume that a CPG is originally represented as a flowchart and we present algorithms that convert a CPG into a decision graph that is subsequently converted into a set of constraints corresponding to different paths linking symptoms and signs of the disease with diagnosis and treatment. We use the constraint programming (CP) methodology to analyze these constraints. CP is an effective paradigm for modeling and solving combinatorial problems [5], [6], [7], [8]. A CP model consists of a set of variables, to be assigned values from a set of choices (a variable's domain), while respecting a set of constraints that restrict the set of allowable combinations of variable/values pairs. Usually, a domain expert defines the variables and their domains along with the applicable constraints. In our case these definitions are automatically derived from the CPGs.

In the proposed approach we separate the conversion of a flowchart into a decision graph from the conversion of a decision graph into a CP model for better comprehensibility and readability of the process. In the end we consider consulting a CPG as the process of solving a CP model. In this model, variables are mostly associated with decision steps and instantiated with values on the basis of available patient information (i.e. known symptoms and signs). Unknown variable values are assigned using search such that the model is consistent and no constraints are violated. As such, consulting multiple CPGs at the same time (for the same patient) can be viewed as a process of reconciling multiple CP models for potential contradictions. Identifying these contradictions is the subject of the research described in this paper. Thus, our research falls into a broad category of work on reasoning in medicine (see [9] for an overview) and our attempt to identify and resolve contradictions has some similarities to conflict resolution methods in automated reasoning.

The long term goal of our research is to support physicians in making treatment decisions for a patient with co-morbidity. Such support does not imply replacing a physician with an automata but providing information about potential implications of the contradictory treatments and suggesting ways of resolving them. The following simple use case describes how we envision our approach being used at the point of care. Upon arrival to the Emergency Department (ED) of a

hospital, a patient is diagnosed with an asthma exacerbation. After a physical examination and a medical history review, the patient's treatment plan is derived from an asthma CPG and information about an underlying peptic ulcer is deduced from their medical records. Upon diagnosing the patient as having moderate asthma exacerbation, a physician prescribes a treatment of oral corticosteroids. Consultation with our approach reveals that such a treatment is not recommended for patients with a peptic ulcer condition because of the increased chance of bleeding, leading to a contradiction that needs to be resolved. Automatically consulting with a drug/drug and drug/disease interactions knowledge base reveals an alternative recommendation of treating this patient with inhaled corticosteroids. This alternative treatment option is presented to a physician for consideration while formulating a treatment plan. This use case positions our proposed approach within the patient management process and illustrates its adjunct and advisory role.

This paper is organized as follows. First we provide a brief overview of CPGs. Secondly we show how we transform a CPG represented as a flowchart into a decision graph and ultimately a CP model to be used when dealing with multiple guidelines. Next we describe how to identify the contradictions that may appear when reconciling multiple CPGs. Finally, we summarize our contributions and present areas for future research.

II. Clinical Practice Guidelines

Evidence-based medicine views a physician's clinical expertise as a key component in the decision-making process that starts with relevant patient data, and then considers research evidence and the patient's preferences before a final decision is made [10]. Research evidence is most often brought to a physician in the form of clinical practice guidelines (CPGs). There is a significant body of literature on representing a CPG, developing a CPG from text, and transforming a CPG into a computer readable and actionable format (see [3], [11] for an overview).

A computer readable format is especially important if a CPG needs to be incorporated into point of care decision support. Examples include among others, PRO*forma* [12] and Asbru [13] formats. A very interesting system called EON [14] was developed at Stanford University to help with creating guideline-based applications using Protege-2000 [15] as a modeling/editing tool for the core guideline model. Research on EON resulted in one of the more popular CPG formats called GLIF [11] that defines the guideline as a flowchart consisting of steps representing clinical actions and decisions. GLIF is strong in incorporating standards with regards to how decision steps are internally represented (their representation relies on Arden Syntax [16]) and how CPG integrates with data using the HL7-RIM model [17]. GLIF is also a first attempt to actively incorporate CPGs into the clinical information system of a healthcare institution. Such integrated guidelines can be executed using the GLEE component [18], but it appears that this execution engine is still under development. A more recent format SDA* [19] was proposed as part of the K4CARE project with the goal of creating a CPG model that can be used as an active patient management tool by multiple caregivers.

Our proposed work uses earlier research on representing and modeling guidelines and shows what it takes to reconcile multiple CPGs to support managing a patient with comorbidity. In our research we consider those CPGs that were developed to assist physicians in one-time diagnosis or disease management and that are used in a primary care organization on the in-patient population. Our proposed modeling process starts with each CPG being represented as a flowchart that includes patient state, decision, and action steps. A flowchart model is automatically transformed into a decision graph that is subsequently transformed into a set of the constraints that make up the CP model. The individual CP models are solved (we define a solution as a set of variable/value pairs satisfying the constraints) and the solutions are parsed for potential contradictions.

III. Reconciling Multiple Clinical Practice Guidelines

In this section we present the process by which we model a single CPG and describe how we solve the corresponding CP model. We continue by discussing *points of contention*, an issue that occurs when trying to reconcile multiple CPGs.

A. Modeling Practice Guidelines

The proposed modeling process is composed of three major stages:

1) Transforming a CPG represented as a flowchart into a decision graph,
2) Transforming a decision graph into a set of logical rules,
3) Transforming the set of rules into a CP model.

The first stage is aimed at simplifying an original CPG. In a flowchart links going out from a decision step are labeled as *true* or *false* depending on whether the condition in this step is satisfied or not. Such a flowchart can be expressed as a pair $\mathcal{F} = (\mathcal{S}, \mathcal{A})$ where $\mathcal{S}$ is the set of the steps and $\mathcal{A}$ is the set of arcs connecting the steps.

Definition 1: Step s in $\mathcal{S}$ is a tuple $s = (id, type, content)$ where id is a unique identifier of the step s, $type$ labels s as a patient state, decision or action step, and $content$ describes the step in terms relevant to the CPG.

Definition 2: Arc a in $\mathcal{A}$ is a tuple $a = (s_1\_id, s_2\_id, value)$ where s_1_id and s_2_id are identifiers for the start and end steps of the arc. The *value* field is $null$ if s_1_id does not identify a decision step; it is $true$ if s_1_id identifies a decision step that when satisfied is followed by the step identified by s_2_id and it is $false$ if s_1_id identifies a decision step that when not satisfied is followed by the step identified by s_2_id.

This representation of a flowchart is very similar to a decision graph except for the intermediate steps that are not decision steps and correspondingly for the arcs having value $null$. To obtain the decision graph we focus on both decision steps and the steps on the leaves as these are essential for

manipulating the models. Hence, a resulting decision graph includes nodes corresponding to decision steps and the steps on the leaves. A leaf node is a patient state or action step from the original flowchart. The decision graph also includes the arcs corresponding to links between these nodes. To simplify the conversion process we assume that the flowchart satisfies the following assumptions:

- intermediate (non-leaf) patient state and action steps have a unique entry arc,
- decision steps have multiple entry arcs only if they originate from other decision steps,
- the flowchart has no cycles.

In practice many well-structured flowcharts that follow standards for medical flowcharts meet or can easily be revised into a version that complies with these assumptions.

For each step $s \in \mathcal{S}$ s.t. $s.type \neq$"decision"
  if $\exists s' \in \mathcal{S} \wedge \exists a \in \mathcal{A}$ s.t. $a := (s, s', null)$
    $s'.content := s.content \cup s'.content$
    if $\exists s'' \in \mathcal{S} \wedge \exists a' \in \mathcal{A}$ s.t. $a' := (s'', s, value)$
      Insert a new arc $a'' := (s'', s', value)$ in $\mathcal{A}$
      Delete a' from $\mathcal{A}$
    Delete a from $\mathcal{A}$, Delete s from $\mathcal{S}$

Algorithm 1: From flowchart to decision graph

To obtain a decision graph from a flowchart we use Algorithm 1. The algorithm deletes intermediate non-decision steps and their related arcs. The remaining steps in $\mathcal{S}$ and the arcs in $\mathcal{A}$ are converted into the set of nodes and arcs in the decision graph. Figure 1 provides an example of the algorithm in action. Figure 2 shows a sample CPG as a decision graph. For a more concise representation, textual descriptions of the steps have been shortened.

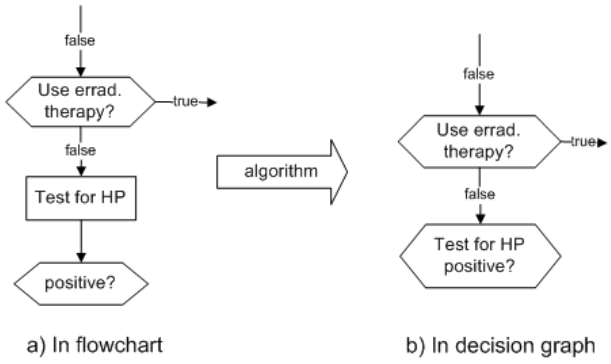

Fig. 1. Example of the algorithm to convert a medical flowchart into a decision graph. The action step "Test for HP" is eliminated and its content is added to the next step "Positive?". A new arc connects the two steps around the deleted one.

The second stage of the modeling process starts with enumerating all paths in the decision graph and then obtaining a set of decision rules. Each path is a sequence of decision nodes connected by the arcs and terminated in a leaf node.

In order to obtain the set of rules from a decision graph we use Algorithm 2. In the algorithm a path starts in the root

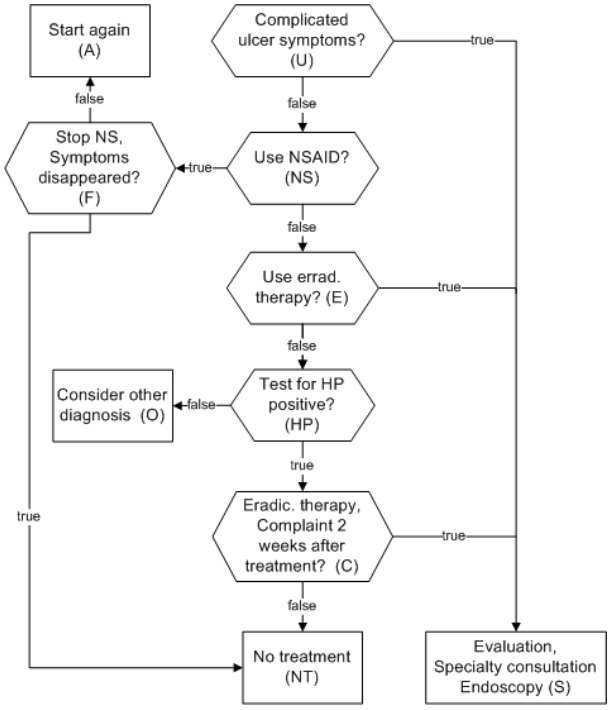

Fig. 2. Decision graph for the simplified (AIHA) CPG for peptic ulcer. The letters in parentheses are identifiers of the nodes.

node and continues with a sequence of arcs in such a way that the end node of an arc is the initial node of the next arc in the sequence. We say that the last node of the path is the ending node of the last arc in the path. If the last node of the path is a leaf we call the path a complete path. The operation $\cup$ between two sets creates a new union set where the elements are in order according to their incorporation to the set. This algorithm extracts all complete paths from a decision graph and from this set generates a set of corresponding logical rules. The premise of a logical rule is a conjunction of conditions from the decision nodes and the consequent is a diagnosis or treatment from the leaf node. An outgoing arc in the path labeled as *false* results in a negated condition for the initial node, and consequences of rules corresponding to "don't do" actions are also negated. In the algorithm the operation $\oplus$ is defined as $\oplus : \mathcal{P} \times \mathcal{A} \rightarrow \mathcal{P}$ where $\mathcal{P}$ is the path space and a new path is created by adding a new arc to the path. For the operation $\oplus$ to be valid the last node in the path being expanded must be the starting node of the arc added to this path. Figure 3 shows the set of logical rules obtained from the decision graph in Figure 2.

In the third stage, the set of logical rules is mapped into a CP model. A CP model is given by a tuple $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$, defined as follows:

- $\mathcal{V} = \{V_1, V_2, ..., V_n\}$ is a set of variables.
- $\mathcal{D} = \{D_1, D_2, ..., D_n\}$ are their respective value domains.
- $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ is a set of constraints that restrict

Q is the set of current complete paths, $Q = \{\}$
P is the set of incomplete paths being explored, $P = \{\}$
v is the current path being explored, $v = (R)$, R is the root node
While $v \neq ()$
  Take last node x in v.
  Let e_1 and e_2 be the false and true arcs from x
  If none of the arcs e_1 and e_2 end in a leaf
    $v := v \oplus e_1$ and $P := P \cup \{v \oplus e_2\}$
  elseif both arcs e_1 and e_2 end in a leaf
    $Q := Q \cup \{v \oplus e_1, v \oplus e_2\}$
    $v ::= ()$
    if $P \neq \{\}$
      $v :=$last path in P, $P := P/\{v\}$
  else {only one of the arcs e_1 and e_2 ends in a leaf}
    let f be the arc that ends in a leaf and g the other arc
    $Q := Q \cup v \oplus f\}, v := v \oplus g$
end {while}
For each path in Q build a logical rule

Algorithm 2: From decision graph to set of rules

1) $U \rightarrow S$
2) $\tilde{}U \wedge NS \wedge F \rightarrow NT$
3) $\tilde{}U \wedge NS \wedge \tilde{}F \rightarrow A$
4) $\tilde{}U \wedge \tilde{}NS \wedge E \rightarrow S$
5) $\tilde{}U \wedge \tilde{}NS \wedge \tilde{}E \wedge \tilde{}HP \rightarrow D$
6) $\tilde{}U \wedge \tilde{}NS \wedge \tilde{}E \wedge HP \wedge C \rightarrow S$
7) $\tilde{}U \wedge \tilde{}NS \wedge \tilde{}E \wedge HP \wedge \tilde{}C \rightarrow NT$

Fig. 3. Set of rules for the peptic ulcer CPG

the possible combinations of values assigned to each variable.

The set of rules defines a CP model in the following way. The variables in the CP model correspond to the predicates in the rules. The domain is {true, false} for all the variables and the constraints are defined by the truth value of the rules. From the set of rules in Figure 3 we can create a CP model as shown in Figure 4. In general we refer to the CP model of a CPG as a *CP-CPG model*.

$$\mathcal{V} = (U, NS, F, E, HP, C, S, NT, A, D)$$
$$\mathcal{D} = \{D_1, .., D_{10}\}, \text{ where } D_i = \{true,\ false\}\ \forall i = \overline{1..10}$$
$$\mathcal{C} = \left\{ \begin{array}{c} U \rightarrow S \equiv true, \tilde{}U \wedge NS \wedge F \rightarrow NT \equiv true \\ \tilde{}U \wedge NS \wedge \tilde{}F \rightarrow A \equiv true, \\ \tilde{}U \wedge \tilde{}NS \wedge E \rightarrow S \equiv true \\ \tilde{}U \wedge \tilde{}NS \wedge \tilde{}E \wedge \tilde{}HP \rightarrow D \equiv true \\ \tilde{}U \wedge \tilde{}NS \wedge \tilde{}E \wedge HP \wedge C \rightarrow S \equiv true \\ \tilde{}U \wedge \tilde{}NS \wedge \tilde{}E \wedge HP \wedge \tilde{}C \rightarrow NT \equiv true \end{array} \right\}$$

Fig. 4. CP-CPG model for the peptic-ulcer CPG

A valid solution for a CP model is an assignment of a value to each variable such that no constraints are violated. As such, the solution task can be approached in three different ways: (1) model checking (to determine whether the problem has a solution), (2) search to find a single solution, or (3) search to find all solutions. For the purpose of our work, we solve the CP-CPG model by looking for a single solution. This enables us to identify consistent models (those with a solution) and inconsistent ones (where no solution exists). We start instantiating the variables for which we know values and perform search to assign values to the remaining variables, see an example of a solution in Figure 5.

$$U = false,\ NS = true,\ F = false,\ E = false,$$
$$HP = false,\ C = false,\ S = false,$$
$$NT = false,\ A = true,\ D = false$$

Fig. 5. A single solution for the peptic ulcer CP-CPG model. This solution only satisfies path 3 because it contains $U = false, NS = true, F = false$ and $A = true$. This solution implies that the patient discontinued the use of NSAID and the symptoms did not disappear thus the CPG should be followed again from the beginning.

In the case where the CP-CPG model is inconsistent we must relax its constraints or replace values for some variables. Such situations arise for example when patient information is incorrectly entered and/or when additional tests need to be carried out to alleviate inconsistencies in a patient's description. Additionally we know a solution must include a valid path because by definition constraints include valid paths and a solution cannot violate any of these constraints.

B. Identifying Points of Contention in Multiple Guidelines

CPGs were originally intended to help with managing a patient who has a single medical condition. While the population is aging and medical care involves increasingly complex cases, there is clearly a need to adapt guidelines for management of patients with co-morbidity. There are several challenges associated with such an adoption and here we focus on those that involve contradictory treatments/interventions when guidelines for different diseases are followed for a single patient.

The contradictions can occur at two levels. One is related to drug interactions when combining different medications that may have harmful effects on a patient. The other is related to drug-disease interactions [20] where a certain medication can exacerbate symptoms of a condition, or a disease can mitigate the absorption rate of a medication. Thus, when reconciling CPGs for a patient with co-morbidity, it is necessary to identify contradictory treatments if they exist. We call such contradictions *points of contention* (*POC*) and they can be explicit or implicit.

Explicit POC occurs when given a patient's state one CPG includes a statement that explicitly identifies a class of treatments that are inadmissible according to a treatment plan defined by another CPG. Implicit POC occurs when a contradiction cannot be identified from the guideline and its establishment requires additional knowledge that goes beyond that encapsulated in a CPG. In our work, we assume that this additional knowledge is stored in a knowledge base. An

example of implicit POC can be drug interactions that are established, for example, from the Epocrates database[1]. Most of the drug-disease interactions will belong to the implicit category.

Formally, explicit and implicit POC are defined as follows:

Definition 3: An explicit POC is a common variable T in multiple CP-CPG models that causes infeasibility for the combined CP-CPG model. If the CP-CPG models are solved independently T would take value $T = v$ in one model and $T = \tilde{}v$ in the other model.

Definition 4: An implicit POC is a variable $T = v$ in the solution of a CP-CPG model that is not explicitly present in another CP-CPG model but in the associated knowledge base $\mathcal{K}$ of the other model it is valued as $T = \tilde{}v$.

To identify POCs we generate the CP-CPG models and solve them as described in Section III-A. If the models do not share common variables, we can deduce there is no explicit POC and we need to look for potential implicit POCs. The more challenging case is when commonalities exist between the models and the solutions are contradictory (i.e. the same variable has different values across the solutions of the models) indicating the existence of an explicit POC. For example, imagine a patient with arthritis who develops a peptic ulcer. If the patient is using NSAID for treatment of arthritis an explicit POC appears because in the peptic ulcer CP-CPG model there is a solution with $NS = false$ (do not use NSAID) while in the arthritis CP-CPG model a solution contains $NS = true$.

Identification of implicit POC requires that a knowledge base containing information about drug interactions and drug-disease interactions is available. Instantiated variables from the solutions of CP-CPG models are located in each knowledge base and potential implicit POCs are established. For example, the CPGs for peptic ulcer and asthma do not share common variables, see Figures 2 and 6, therefore there is no explicit POC and one needs to search for implicit ones in the knowledge base.

A knowledge base $\mathcal{K}$ for each CPG may be represented as a lookup table with keys corresponding to instantiated variables. Figure 7 partially presents $\mathcal{K}$ for the peptic ulcer CPG and it contains information about potentially harmful treatments. Information in this table is transformed into constraints that are added to the respective CP-CPG model in order to define further restrictions on the domains of the variables corresponding to harmful treatments. Therefore a solution to the augmented CP-CPG asthma model that contains $OC = true$ indicates an implicit POC with respect to the peptic ulcer CPG.

The first step to eliminating both types of POC could be to check whether the POC is due to a variable's value that was assigned during search when solving the CP-CPG model or due to a variable with a value predefined according to available patient information. In the first case we could look for a different solution of the CP-CPG model that does not produce POC, if one exists. An alternate solution allows us to suggest a different course of treatment to the physician, one that does not include any known point of contention.

[1]www.epocrates.com

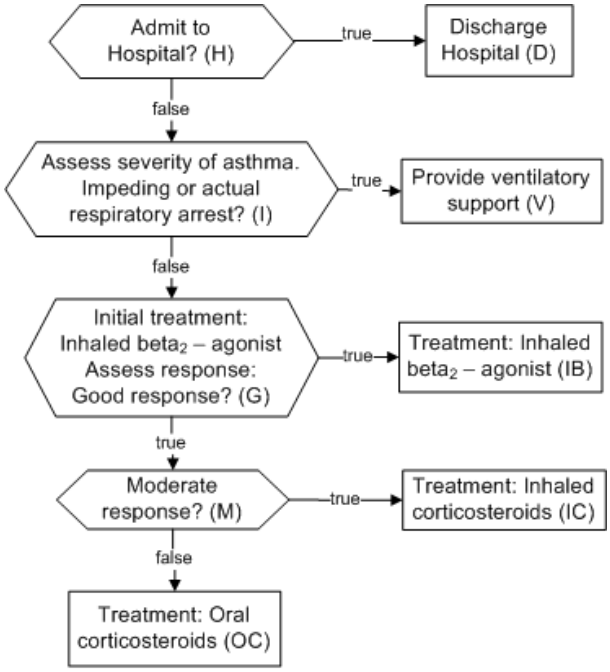

Fig. 6. An example of a decision graph for asthma disease. The letters in parentheses are identifiers of the nodes.

Instantiated variable	Type of Interaction
$AS = false$	adverse
$OC = false$	adverse
$P = false$	adverse

Fig. 7. Knowledge base $\mathcal{K}$ for the peptic ulcer CPG (AS - aspirin, OC - oral corticosteroids, P - potassium supplements)

If no alternative solution exists or the POC is caused by a variable's predefined value, the problem becomes more complex. A relatively simple approach to this problem is to present the physician with a partial solution along with the identified POC. The partial solution would include assignments of values to variables not included in the POC and the POC itself would provide information that can be used by the physician in making additional treatment decisions. A more involved and difficult approach involves modifying the structure of the original guideline or performing actions that alter the patient's state. However, discussion of these modifications is beyond the scope of this paper and we touch on this issue briefly in Section V.

IV. Discussion

In this paper we presented an approach that enables the discovery of POCs as a first step in reconciling multiple CPGs. Our proposed methodology starts with a guideline represented as a flowchart and then automatically converts it into a decision graph. From the graph a set of logical rules are developed and transformed into a CP-CPG model.

A model for each guideline is solved and potential POCs are identified. Our use of CP to model CPGs allows for easy identification of solutions, even for instances with incomplete patient data. Once a solution is found for each CP-CPG model, we described methods for identifying both implicit and explicit POCs. In cases where an alternate solution free of POCs is available, we return this alternate solution to the physician. Otherwise, each POC is flagged and appropriate information is presented to the physician. We believe that flagging the POCs is an important step towards supporting the use of the CPGs in managing complex medical cases where multiple treatments need to be reconciled and alternatives need to be evaluated. The automatic identification of the POCs, when the CPG is originally represented as a flowchart, constitutes the main contribution of the research described in this paper.

V. Future Research

We are exploring several ideas for dealing with POC, especially when finding a valid solution requires modifications to the original CPGs. One of them involves the use of alternative treatments that would replace some of the original action steps from a flowchart and eliminate the root cause of the POC. However, such a modification of CPGs requires a comprehensive knowledge base that goes beyond simple interactions and needs to be extracted from textbooks and online evidence repositories. This knowledge base should be represented in a form that is easy to parse once CP-CPG models' solutions are identified. We are also working on tagging a CP-CPG model with indices to relevant meta-constraints that capture complex interactions between CP-CPG model variables and other variables not included in a model. Once implicit POC is established through consulting the expanded CP-CPG model, each variable/value pair associated with a POC will be passed to a set of meta-constraints for a potential solution that identifies alternative treatment(s). If such an alternative exists, it will be used to replace a node in a decision graph, and ultimately to create a new action step in a flowchart. If substitution is not possible (there is no solution to the meta-constraints), then one should consider which disease is more acute or least "costly" in terms of the POC and prioritize the associated CPG accordingly. Such automatic adaptation of the CPGs for patient-specific condition would allow to customize CPG when treating a patient with co-morbidity.

Acknowledgements: Research described in this paper was funded by grants from the Natural Sciences and Engineering Research Council of Canada. The third author acknowledges support of the Polish Ministry of Science and Higher Education (grant N N519 314435). The authors would like to thank Professor Brian Rowe, M.D. from the University of Alberta Hospital for help with interpreting clinical practice guidelines.

References

[1] M. D. Cabana, C. S. Rand, N. R. Powe, A. W. Wu, M. H. Wilson, P.-A. C. Abboud, and H. R. Rubin, "Why don't physicians follow clinical practice guidelines? a framework for improvement," *The Journal of the American Medical Association*, vol. 282, no. 15, pp. 1458–1465, 1999.

[2] N. Mulyar, W. van der Aalst, and M. Peleg, "A pattern-based analysis of clinical computer-interpretable guideline modeling languages," *Journal of American Medical Informatics Association*, vol. 14, no. 6, pp. 781–787, 2007.

[3] P. A. de Clercq, J. A. Blom, H. Korsten, and A. Hasman, "Approaches for creating computer-interpretable guidelines that facilitate decision support," *Artificial Intelligence in Medicine*, vol. 31, pp. 1–27, 2004.

[4] Society for Medical Decision Making and Committee on Standardization of Clinical Algorithms, "Proposal for clinical algorithm standards," *Medical Decision Making*, vol. 12, pp. 149–154, 1992.

[5] M. L. Littman, G. A. Keim, and N. Shazeer, "A probabilistic approach to solving crossword puzzles," *Artificial Intelligence*, vol. 134, no. 1-2, pp. 23–55, 2002.

[6] M. Michalowski and C. A. Knoblock, "A constraint satisfaction approach to geospatial reasoning," in *Proceedings of AAAI-05*, 2005, pp. 423–429.

[7] B. A. Nadel, "Representation selection for constraint satisfaction: A case study using n-queens," *IEEE Expert: Intelligent Systems and Their Applications*, vol. 5, no. 3, pp. 16–23, 1990.

[8] H. Simonis, "Sudoku as a constraint problem," in *Proceedings of CP*, 2005, pp. 13–27.

[9] E. Horvitz, "Automated reasoning for biology and medicine," in *Advances in Computer Methods for Systematic Biology: Artificial Intelligence, Databases, and Computer Vision*. Johns Hopkins University Press, From opening talk, Conference on AI in Systematic Biology,September 1990, Napa Valley, California, 1993.

[10] R. Haynes, P. Devereaux, and G. Guyatt, "Clinical expertise in the era of evidence-based medicine and patient choice," *Evidence Based Medicine*, vol. 7, pp. 36–38, 2002.

[11] M. Peleg, A. Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P. Mork, L. Ohno-Machado, E. Shortliffe, and R. Greenes, "Glif3: The evolution of a guideline representation format," in *Proc AMIA Symp*, 2000, pp. 645–649.

[12] R. A. Fox J, Johns N, "Disseminating medical knowledge: The PRO*forma* approach," *Artificial Intelligence in Medicine*, vol. 14, pp. 157–181, 1998.

[13] J. P. Shahar Y, Miksch S, "The Asgaard project: A task specific framework for the application and critiquing of time oriented clinical guidelines," *Artificial Intelligence in Medicine*, vol. 14, pp. 29–51, 1998.

[14] M. M. S.W. Tu, "Modeling data and knowledge in the EON guideline architecture," in *Proceedings of the Medinfo Conference*, 2001, pp. 280–284.

[15] W. Grosso, H. Eriksson, R. Fergerson, J. Gennari, S. Tu, and M. Musen, "Knowledge modeling at the millennium (the design and evolution of Protégé-2000)," in *Proceedings of the 12th International Workshop on Knowledge Acquisition, Modeling and Management*, 1999.

[16] J. Starren, G. Hripcsak, D. Jordan, B. Allen, C. Weissman, and P. Clayton, "Encoding a post-operative coronary artery bypass surgery care plan in the Arden syntax," *Computational Biology in Medicine*, vol. 24, pp. 411–417, 1994.

[17] G. Schadow, D. Russler, C. Mead, and C. McDonald, "Integrating medical information and knowledge in the HL7-RIM," in *Proceedings of the AMIA Symposium*, 2000, pp. 764–768.

[18] D. Wang and E. H. Shortliffe, "GLEE–a model-driven execution system for computer-based implementation of clinical practice guidelines," in *Proceedings of the AMIA Symposium*, 2002, pp. 855–859.

[19] D. Riaño, "The SDA* model: a set theory approach," in *Proceedings of the 20th IEEE International Symposium on Computer-based Medical Systems*, 2007, pp. 563–568.

[20] C. Lindblad, J. Hanlon, C. Ross, R. Sloane, C. Pieper, E. Hajjar, C. Ruby, and K. Schmader, "Clinically important drug-disease interactions and their prevalence in older adults," *Clinical Therapeutics*, vol. 28, no. 8, pp. 1133–1143, 2006.