AIME 2017 16th Conference on Artificial Intelligence in Medicine
Vienna, Austria June 21-24, 2017

USING CONSTRAINT LOGIC PROGRAMMING FOR THE VERIFICATION OF CUSTOMIZED DECISION MODELS FOR CLINICAL GUIDELINES

Szymon Wilk[1,2], Adi Fux[3], Martin Michalowski[2,4], Mor Peleg[3], Pnina Soffer[3]

[1] *Poznan University of Technology, Poznan, Poland*
[2] *University of Ottawa, Ottawa, Canada*
[3] *University of Haifa, Haifa, Israel*
[4] *MET Research Group, Ottawa, Canada*

Motivation

- When managing a patient according to clinical practice guidelines (CPGs) physicians consider additional contextual aspects (e.g., social situation, comorbidities) [Fux et al., 2012]
- Computer-interpretable guidelines (CIGs) should include arguments corresponding to these aspects → an evidence-based (primary) layer and a contextual (secondary) layer
- A two-layered CIG (decision model) should results in a more standardized management and better compliance to therapy

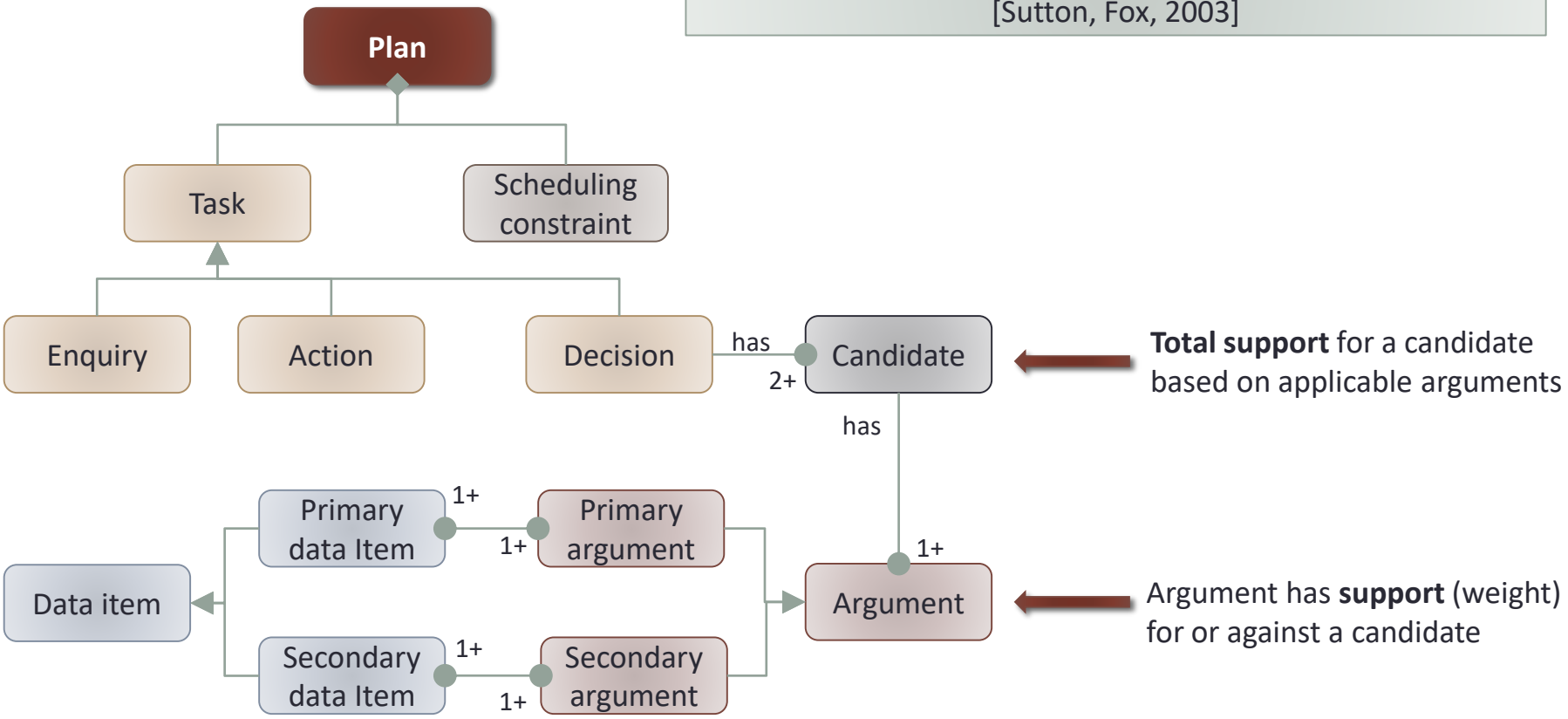
Motivation

- A two-layered decision needs to have the following properties
  - **Completeness** → for each valid combination of patient data items at least one option for each decision should be indicated
  - **Secondarity** → the secondary layer should *modulate* only these decision options that are indicated by the primary layer
- Manual verification of these properties is difficult and requires involvement of a knowledge engineer and domain expert

How to automatically verify and control revisions of two-layered decision models represented in the PROforma CIG formalism?

Two-layered Decision Model

Extension of the PROforma CIG formalism
[Sutton, Fox, 2003]



Two-layered Decision Model

- **Primary layer** → a plan where all arguments associated with candidates are primary arguments
- **Secondary layer** → a set of secondary arguments (and their secondary data items) associated with the candidates from the primary layer (no new candidates!)

- **Completeness** → for each decision there is at least one candidate with total support in both layers ≥ *threshold*
- **Secondarity** → there is no candidate with total support in the primary layer < 0 and total support in both layers ≥ *threshold*

Constraint Logic Programming (CLP) and MiniZinc

- CLP is a technique to solve a constraint satisfaction problem using logic programming
- A CLP model is composed of variables, constraints and a goal (with an optional goal function)
- A solution is such assignment of variable values that satisfies (optimizes) the goal and does not violate any constraints

---

- MiniZinc is a modeling language (and a set of reference tools) that becomes a standard for CLP

Verification and Revision of Two-Layered Models

Indirect approach to verification

- Translation of a two-layered model into a CLP model that identifies “problematic” cases
- Problematic cases violate at least one of the two desired properties (secondarity and completeness)
- A two-layered model is positively verified if there are no solutions to a corresponding CLP model

---

- Revisions of a CLP model resulting in an empty set of solutions

Verification and Revision Algorithm

```
procedure verify_and_revise(in initial_proforma_model, out final_profoma_model)
begin
1    translate initial_proforma_model into minizinc_model
2    solutions := solve minizinc_model
3    while solutions exist do begin
4         revise minzinc_model given solutions
5         solutions := solve minizinc_model
6    end
7    translate minzinc_model into final_proforma_model
end
```

Components of the CLP/MiniZinc model:

- Variables corresponding to (1) primary and secondary data items, (2) intermediary decision steps and (3) support of specific candidates
- Constraints for (1) computing support for candidates, (2) enforcing the violation of both properties and (3) excluding clinically invalid cases (optional, based on domain knowledge)
- “Simple” goal with no goal function

Verification and Revision Algorithm

```
procedure verify_and_revise(in initial_proforma_model, out final_profoma_model)
begin
1    translate initial_proforma_model into minizinc_model
2    solutions := solve minizinc_model
3    while solutions exist do begin
4         revise minzinc_model given solutions
5         solutions := solve minizinc_model
6    end
7    translate minzinc_model into final_proforma_model
end
```

- Solving the initial MiniZinc model to check whether problematic cases exist and revisions are necessary

Verification and Revision Algorithm

```
procedure verify_and_revise(in initial_proforma_model, out final_profoma_model)
begin
1    translate initial_proforma_model into minizinc_model
2    solutions := solve minizinc_model
3    while solutions exist do begin
4         revise minzinc_model given solutions
5         solutions := solve minizinc_model
6    end
7    translate minzinc_model into final_proforma_model
end
```

- Revisions of constraints in the MiniZinc models corresponding to : (1) modifying support and conditions in arguments, (2) removing existing arguments, and (3) adding new arguments
- Revisions applied to both layers (the primary layer adjusted according to clinical experience)
- Revisions introduced until all problematic cases have been excluded/addressed

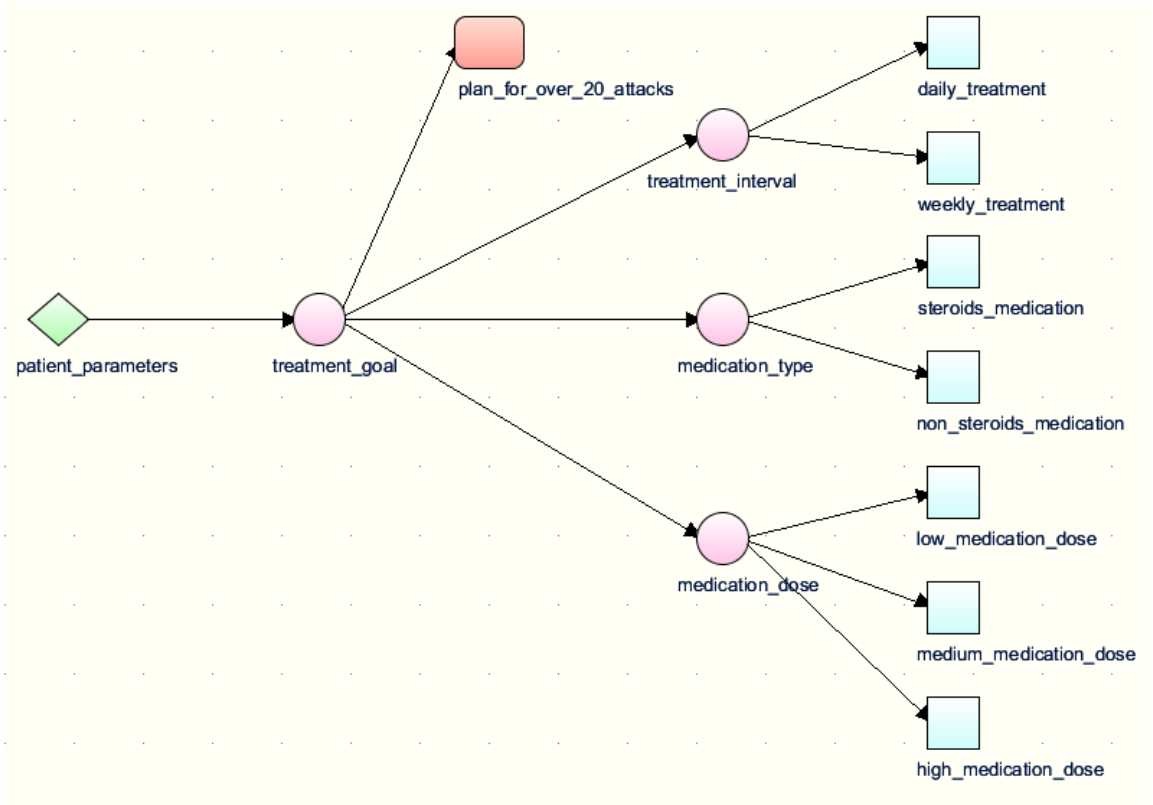
Verification and Revision Algorithm

```
procedure verify_and_revise(in initial_proforma_model, out final_profoma_model)
begin
1    translate initial_proforma_model into minizinc_model
2    solutions := solve minizinc_model
3    while solutions exist do begin
4         revise minzinc_model given solutions
5         solutions := solve minizinc_model
6    end
7    translate minzinc_model into final_proforma_model
end
```

- Translation focused on modified constraints (corresponding to revised arguments), other elements of the CIG are not changed

Case Study – a Two-Layered CIG for Asthma

An asthma CIG (based on published British and Israeli CPGs) with a secondary layer capturing patient's personal context (developed by 15 clinical experts)



Primary (clinical) data items

1. severity of attacks
2. daily limitation level
3. forced expiratory volume (FEV1)
4. number of monthly attacks

Secondary (personal) data items

1. routineness of patient's daily life
2. patient's communication level
3. level of family support

Translating CIG into CLP Model

Argument list

```
Aggressive_goal_Arg_01  {Asthma_attacks_level  = severe       } (-(3))
Aggressive_goal_Arg_02  {Asthma_attacks_level  = medium  OR Asthma_attacks_level  = low      } (3)
Aggressive_goal_Arg_03  {Monthly_attacks <= 8      } (3)
Aggressive_goal_Arg_04  {FEV1_result > 80      } (-(3))
Aggressive_goal_Arg_05  {FEV1_result <= 80      } (3)
Aggressive_goal_Arg_06  {Daily_limitation_level  = severe       } (-(3))
Aggressive_goal_Arg_10  {communication_level  = high_comm       } (1)
Aggressive_goal_Arg_11  {Daily_routine_level  = no_routine       } (-(1))
Aggressive_goal_Arg_12  {communication_level  = low_comm       } (-(1))
Aggressive_goal_Arg_13  {support_level  = frequent_supp       } (1)
```

Primary (1-6) and secondary (10-13) arguments for *aggressive treatment* option

Constraints computing support for *aggressive treatment* option in the primary and secondary layer

```
var int: pl_aggressive_goal_support;
constraint
  pl_aggressive_goal_support =
    (if asthma_attacks_level == 3 then -3 else 0 endif) +
    (if asthma_attacks_level == 2 \/ asthma_attacks_level == 1 then 3 else 0 endif) +
    (if monthly_attacks <= 8 then 3 else 0 endif) +
    (if FEV1_result > 80 then -3 else 0 endif) +
    (if FEV1_result <= 80 then 3 else 0 endif) +
    (if daily_limitation_level == 3 then -3 else 0 endif);
```

```
var int: sl_aggressive_goal_support;
constraint
  sl_aggressive_goal_support =
    (if communication_level == 3 then 1 else 0 endif) +
    (if daily_routine_level == 1 then -1 else 0 endif) +
    (if communication_level == 1 then -1 else 0 endif) +
    (if support_level == 3 then 1 else 0 endif);
```

Translating CIG into CLP Model

```
constraint
  ((pl_plan_for_over_20_attacks_support + sl_over_20_attacks_support_plan_support < 1 /\
    pl_aggressive_goal_support + sl_aggressive_goal_support < 1 /\
    pl_basic_goal_support + sl_basic_goal_support < 1) \/
  (pl_daily_interval_support + sl_daily_interval_support < 1 /\
    pl_weekly_interval_support + sl_weekly_interval_support < 1) \/
  (pl_steroids_support + sl_steroids_support < 1 /\
    pl_non_steroids_support + sl_non_steroids_support < 1) \/
  (pl_low_dose_support + sl_low_dose_support < 1 /\
    pl_medium_dose_support + sl_medium_dose_support< 1 /\
    pl_high_dose_support + sl_high_dose_support < 1))
  \/
  ((pl_plan_for_over_20_attacks_support < 0 /\
      pl_plan_for_over_20_attacks_support + sl_plan_for_over_20_attacks_support >= 1) \/
  (pl_basic_goal_support < 0 /\ pl_basic_goal_support + sl_basic_goal_support >= 1) \/
  (pl_aggressive_goal_support < 0 /\ pl_aggressive_goal_support + sl_aggressive_goal_support >= 1) \/
  (pl_daily_interval_support < 0 /\  pl_daily_interval_support + sl_daily_interval_support >= 1) \/
  (pl_weekly_interval_support < 0 /\ pl_weekly_interval_support + sl_weekly_interval_support >= 1) \/
  (pl_steroids_support < 0 /\ pl_steroids_support + sl_steroids_support >= 1) \/
  (pl_non_steroids_support < 1 /\ pl_non_steroids_support + sl_non_steroids_support >= 1) \/
  (pl_low_dose_support < 0 /\ pl_low_dose_support + sl_low_dose_support >= 1) \/
  (pl_medium_dose_support < 0 /\ pl_medium_dose_support + sl_medium_dose_support >= 1) \/
  (pl_high_dose_support < 0 /\ pl_high_dose_support + sl_high_dose_support >= 1));

 constraint
   FEV1_result > 60 \/ asthma_attack_level > 1;
 constraint
   FEV1_result > 60 \/ daily_limitation_level > 1;
 constraint
   monthly_attacks <= 8 \/ asthma_attack_level > 1;
 constraint
   monthly_attacks <= 8 \/ daily_limitation_level > 1;
```

Part of constraint associated with **violating completeness**

Part of constraint associated with **violating secondarity**

Auxiliary constraints defining valid patient cases

Solving CLP Model

- 28,656 solutions (problematic cases) for the initial CLP model – large number due to numerical values
- Only the completeness property for three decisions (all but medication dosage) was violated

Sample problematic cases

Data item	Case 1	Case 2	Case 3
asthma_attack_level	low	moderate	low
daily_limitation_level	severe	minor	minor
FEV1_result	81	81	81
monthy_attacks	5	1	5
daily_routine_level	no_routine	semi_routine	routine
communication_level	medium	medium	medium
support_level	low	medium	medium
Decisions with violated completeness	treatment_goal treatment_interval medication_type	treatment_interval	medication_type

Revising CLP Model

- Revisions introduced by a knowledge engineer cooperating with a clinical expert, focused on the primary layer
- 9 iterations (revise-solve) to obtain a model with no solutions

Summary of revisions introduced to ensure completeness (and secondarity)

		# Arguments					
		Primary Layer			Secondary Layer		
Decision	**Option**	**Initial CIG**	**Final CIG**	**Change**	**Initial CIG**	**Final CIG**	**Change**
treatment_goal	basic_goal	4	27	23	2	2	0
	aggressive_goal	6	37	31	4	4	0
	plan_for_over_20_attacks	1	1	0	0	0	0
treatment_interval	weekly_interval	7	18	11	3	3	0
	daily_interval	4	20	16	1	1	0
medication_type	non_steroids	5	15	10	1	1	0
	steroids	6	25	19	2	2	0
medication_dose	low_dose	6	6	0	3	3	0
	medium_dose	5	5	0	2	2	0
	high_dose	8	8	0	2	2	0

Examples of Revisions

Initial CIG

Argument list

non_steroids_Arg_01 {result_of(treatment_goals) = Aggressive_goal } (-(3))
non_steroids_Arg_02 {Asthma_attacks_level = severe } (-(3))
non_steroids_Arg_03 {Daily_limitation_level = minor } (6)
non_steroids_Arg_04 {Asthma_attacks_level = low } (-(3))
non_steroids_Arg_05 {communication_level = high_comm } (1)

Introduced revisions
- modified support (→ initial 03 and final 10)
- additional conditions (→ initial 04 and final 02-04)
- removed arguments (→ initial 01)
- added arguments (→ final 07-09)

Final CIG

Argument list

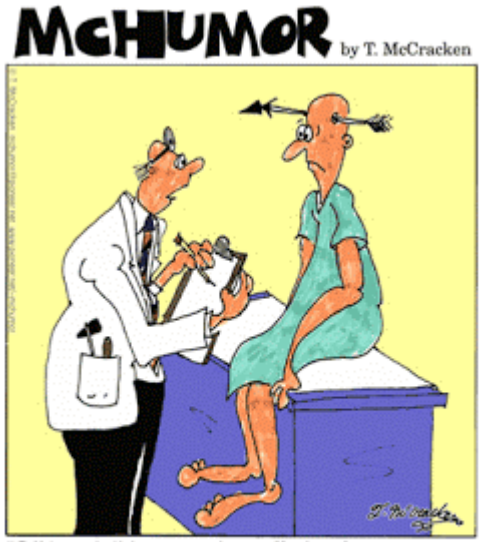
non_steroids_Arg_01 {result_of(treatment_goals) = Basic_goal } (6)
non_steroids_Arg_02 {Asthma_attacks_level = low AND Monthly_attacks <= 4 } (6)
non_steroids_Arg_03 {Asthma_attacks_level = low AND Monthly_attacks > 4 AND Monthly_attacks <= 8 } (3)
non_steroids_Arg_04 {Asthma_attacks_level = low AND Monthly_attacks > 8 } (3)
non_steroids_Arg_05 {Asthma_attacks_level = severe AND Monthly_attacks > 4 AND Monthly_attacks <= 8 } (-(3))
non_steroids_Arg_06 {Asthma_attacks_level = severe AND Monthly_attacks > 8 } (-(3))
non_steroids_Arg_07 {FEV1_result > 20 AND FEV1_result <= 40 } (-(3))
non_steroids_Arg_08 {FEV1_result > 60 AND FEV1_result <= 80 } (3)
non_steroids_Arg_09 {FEV1_result > 80 } (3)
non_steroids_Arg_10 {Daily_limitation_level = minor } (3)
non_steroids_Arg_11 {Asthma_attacks_level = low AND Monthly_attacks <= 4 AND Daily_limitation_level = minor } (3)
non_steroids_Arg_12 {Asthma_attacks_level = low AND Monthly_attacks > 4 AND Monthly_attacks <= 8 AND Daily_limitation_level = minor } (3)
non_steroids_Arg_13 {Asthma_attacks_level = moderate AND Monthly_attacks <= 4 AND Daily_limitation_level = minor } (3)
non_steroids_Arg_14 {Asthma_attacks_level = moderate AND Monthly_attacks > 4 AND monthly_attacks <= 8 AND Daily_limitation_level = minor } (3)
non_steroids_Arg_15 {Communication_level = high_comm} (1.0)

Conclusions

- Successful application of CLP to verify two-layered CIGs in PROforma and to control the revision process
- Significant support for a knowledge engineer → ability to verify and revise more complex models in reasonable time

---

- Automatic revision of support associated with specific arguments (reformulation of a CLP model)
- More sophisticated (e.g., considering organizational resources) and automatically introduced modulation options



"Off hand, I'd say you're suffering from an arrow through your head, but just to play it safe, I'm ordering a bunch of tests."

Thank you for your attention

Questions?