WP 2: Selection and Asbru Modelling of Reference Protocols

Silvia Miksch
Technische Universität Wien
Institut für Softwaretechnik und Interaktive Systeme

silvia@ifs.tuwien.ac.at
http://www.ifs.tuwien.ac.at/~silvia

Workpackage number :	WP2			Start date or starting event:		month 0	
Participant number:	**2**	1	3	4	5		
Person-months per participant:	1	0.5	1	0.5	0.5		

Objectives

Definition of formal semantics of Asbru elements in KIV and identification of properties to verify on Asbru elements.

Description of work

A formal semantics will be defined for a sufficiently large part of Asbru, a part that should be enough for covering the selected reference protocols. In parallel, a number of interesting protocol properties will be determined and translated into the Asbru formal semantics.

In this stage, existing work on the definition of formal semantics of Asbru elements in KIV and on the identification of verifiable properties will be used.

Deliverables

D2, D3.

Milestones and expected result

The milestone will be the subset of Asbru elements to be used in the formalisation, crucial for the later steps.

The results will be a formal language to model medical protocols, on one hand, and a list of desirable/required protocol properties, on the other.

Content

Focus on

- **A'dam Properties** -> *file: adam.pdf*
- **Vienna Properties**
  - Condition Properties -> *file: condition.pdf*
  - Temporal Dimensions -> *file: time.pdf*

Properties - 14-Feb 2002

- **TERMINATION**
- **Plan Body implies Intentions**
- **Redundancies of Subplans**
  - Christoph is responsible
- **Correct Replacement of Subplans**
- **Internal Coherence**
  - Summary of all (Georg's) general properties
  - Lower Priority

A'dam Properties (sloppy/informal/vague)

- **Parameters Never Used or Asked**
  - In both directions
  - This are "gaps" in the protocol

  --> static analysis, don't do it in KIV
- **Effects : Intentions**
  - Example
    An effect can make the intention (partial) true or (partial) false

  --> in KIV: Effects ... Plan Body
- **Subplan has to Contribute to the Intention of the Plan**

  --> trivial way to do it --> NOT interesting; NOT realistic **A**

  --> chronic vs. Emergency medicine

A'dam Properties (sloppy/informal/vague)

- **Control Structure : Conditions**
  - Example:
    if in a do-all-sequentially, one filter-condition fails then there is an error in the plan

    If in a do-some-any-order, the actions of one subplan should not influence conditions of other subplans
  - important for the designer (module testing in SWE)
  - expressible and verifiable in KIV
  - more examples are possible:
    example: statisfiability of conditions
  - **NOT HIGH PRIORITY *!***

A'dam Properties (sloppy/informal/vague)

- **Combinations of Conditions**
  - Filter & Setup & Complete
    (& Some-terminology) -> Intentions
  - Variants of these
    if (filter & setup) & ~suspended & ~aborted & complete & some terminology (domain ontologies) -- then overall - intentions holds
  - Complete-conditions should be consistent (or derivable) with the Filter/Setup-conditions and the execution of the plan
    - Derivable: Filter & Setup -> [plan]Complete -> Intention
- **similar to **A****

A’dam Properties (sloppy/informal/vague)

- **Plan Library Completeness Properties**
  - for each setup condition should be a plan or set of plans that satisfied the setup condition.

    (by intentions/compete-conditions/effects)

    (i.e. there must be a plan to make the setup conditions true)
- ***NOT INCLUDED,***
  no dynamic dimensions
- *“stupid” idea (WR: deduction is the wrong way)*

A’dam Properties EXTENSIONS

- **Effects Represent *Actual Behavior*, where as Intentions represent *Expected Behavior*** (adapted)
  - Intention - may be true --> expected behavior
  - Effects - must be true --> actual behavior
- **Overall Intentions are a Subset of Effects**
  - Effects with Likelihood = 1.0 = overall Intention
- ***NOT INCLUDED***

Condition Properties

Knowledge-Based Verification of Clinical Guidelines by Detection of Anomalies

Georg Duftschmid, Silvia Miksch

- **Checking anomalies within 3 levels**
- **Anomalies [Preece et al.] & extensions**
- **Examples of 3-level checking**

Multi-level Verification

- Check for anomalies within several layers
  - Level 1: within a single component
  - Level 2: within a single plan
  - Level 3: within a plan hierarchy

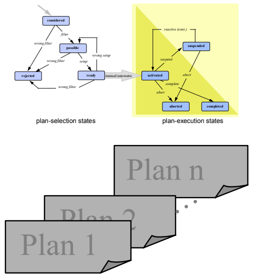
Goal: to arrive at legal or meaningful plans

	Level of Checking		
Decomposition	**Level 1**	**Level 2**	**Level 3**
Plan A			
Plan A_a			
Cond. A_1	✓		
Cond. A_2	✓	✓	
…	✓		
Cond. A_n	✓		✓
…			
Plan A_x			
Cond. X_1	✓		
Cond. X_2	✓	✓	
…	✓		
Cond. X_o	✓		

Verification / Conditions

- General vs. specific anomalies
  - General KB-anomalies [Preece et al.]
  - Additional, domain-specific anomalies

- Preece et al. define rule base anomalies



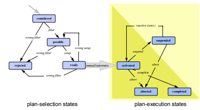
R_1: $L_{11} \wedge \ldots \wedge L_{1j} \rightarrow C_1$

$\vdots \quad \vdots \quad \vdots$

R_n: $L_{n1} \wedge \ldots \wedge L_{nk} \rightarrow C_n$

2 Rule Bases

1. generic Model of Plan States: MPS



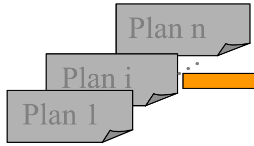
R_1 : considered(pl,pa)

R_2 : considered(pl,pa) $\wedge$ filter(pl,pa) $\rightarrow$ possible(pl,pa)

$\vdots \quad \vdots$

R_{13} : suspended(pl,pa) $\wedge$ abort(pl,pa) $\rightarrow$ aborted(pl,pa)

2. specific Plan Conditions: PC = $PC_1 \cup \ldots \cup PC_n$



R_1 : female(pa) $\wedge$ pregnant(pa) $\rightarrow$ filter(GDM-II,pa)

$\vdots \quad \vdots$

R_7 : insulin-indicator(pa) $\rightarrow$ abort(GDM-II,pa)

MPS $\cup$ PC_i defines execution control of plan i

Anomalies / Level 1

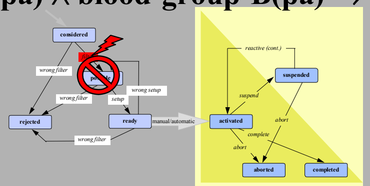
- Every condition must have a chance to be satisfied

  $(\exists\, E)\ fireable(R,PC,E)\ \forall R \in PC$

  $\cong$ unfireable rule in [Preece et al.]

Example:

blood-group-A(pa) $\wedge$ blood-group-B(pa) $\rightarrow$ filter(P, pa)



Anomalies / Level 2

- Every valid sequence of plan states must be reachable

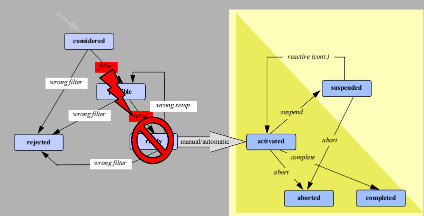
  $(\exists\, E, \sigma)\ (\forall\, R \in MPS,\ 1 \leq i \leq n)$
  $inferable(conseq(R)\sigma,\ MPS \cup PC_i,\ E)$

  $\cong$ superset of missing rule in [Preece et al.]

Example:

male(pa) $\rightarrow$ filter(P, pa)

pregnant(pa) $\rightarrow$ setup(P, pa)



Anomalies / Level 3

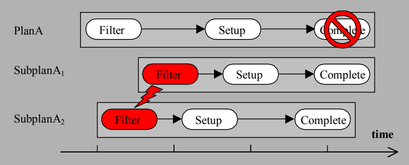
- Every plan must be able to complete

$(\exists E,\sigma)\ (\forall\ 1 \leq i \leq n,\ k \in RelevantSubplans_i)$
$[inferable(conseq(R10)\sigma,\ MPS \cup PC_i,\ E) \wedge$
$inferable(conseq(R10)\sigma,\ MPS \cup PC_k,\ E)]$
$R10 \in MPS$

Example:

male(pa) → filter(A_1, pa)

female(pa) → filter(A_2, pa)



⇒The complete-condition must be inferable for every plan and all its relevant subplans

Properties (1/2)

Internal Coherence --> low priority

√ Every Condition must have a chance to be satisfied -->low priority, possible in KIV

√ Every valid sequence of plan states must be reachable -> low priority, possible in KIV

√ Every plan must be able to complete -> low priority, possible in KIV

- The target state of a transition should be unambiguous
  - **low priority, possible in KIV**
- Child plan should not be stopped by parent
  - **low priority, possible in KIV**
- No state of a plan should be skipped
  - **low priority, possible in KIV**

Properties (2/2)

- Every plan must have a chance to execute its body
  - **low priority, possible in KIV**
- A plan must not loop in its execution
  - **low priority, possible in KIV**
- Every Condition, checked at a ”wait-state” must be defined in a way, that it makes sense to wait for its satisfaction
  - **low priority, possible in KIV**
- A Condition must only refer to states of other plans, which are reachable
  - **low priority, possible in KIV**

SUM =11 properties

Advantages & Limitations

+ Incremental verification possible according to 3 levels

+ Domain-specific knowledge is considered

+ Applicable to a wide scope of protocol representations

+ Partial checks are not computationally expensive

- Dynamic properties are ignored

- Partial checks do not guarantee a completely correct plan

Temporal Dimensions

Verification of Temporal Scheduling Constraints in Clinical Practice Guidelines

Georg Duftschmid, Silvia Miksch, Walter Gall

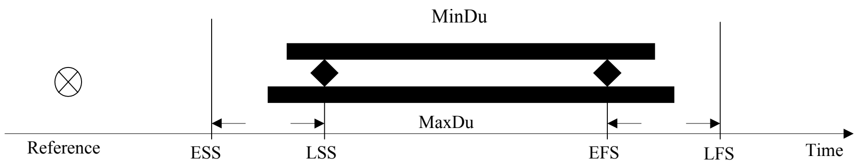
- **Purpose**
  - Temporal scheduling constraints are consistent ?
    - control flow operators and the hierarchical structure
  - More explicit representation of non-minimal constraints
  - Used by the guideline interpreter
    - to assemble feasible time intervals

Types of Scheduling Constraints

- **Temporal scheduling constraints**
- **Scheduling constraints implied by the guideline's control flow**
- **Scheduling constraints implied by a hierarchical structuring of a guideline**

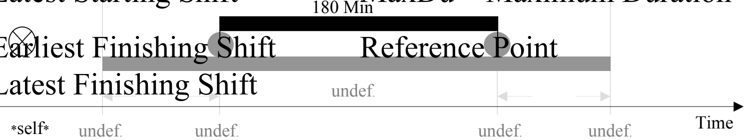
Asbru's Time Annotation

`Definition: [[ESS, LSS], [EFS, LFS], [MinDu, MaxDu], Reference]`

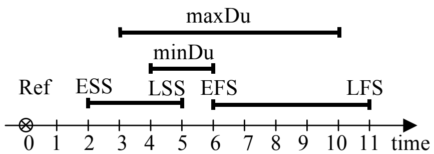


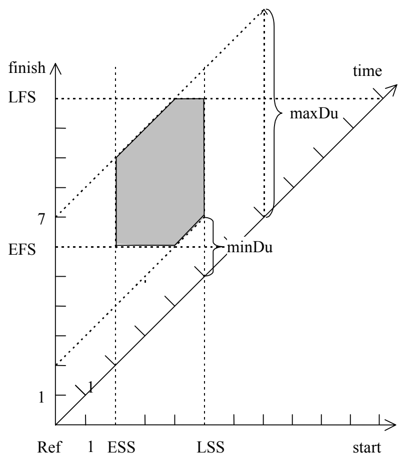
`Example: [[_, _], [_, _], [180 MIN, _], *self*]`

ESS Earliest Starting Shift   MinDu Minimum Duration
LSS Latest Starting Shift   MaxDu Maximum Duration
EFS Earliest Finishing Shift   Reference Point
LFS Latest Finishing Shift

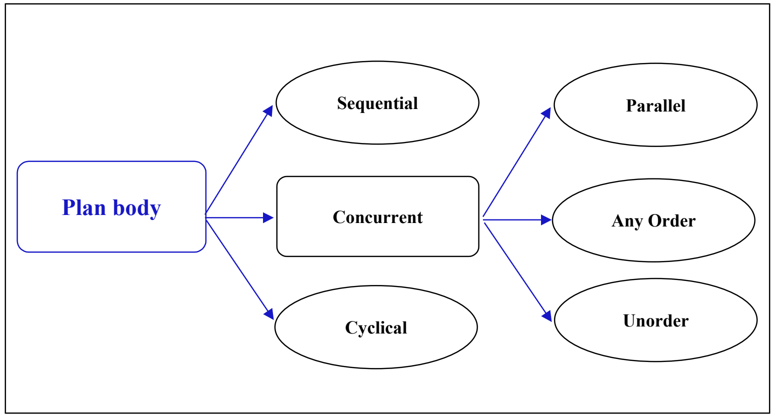


Asbru's Time Annotation





Plan Body

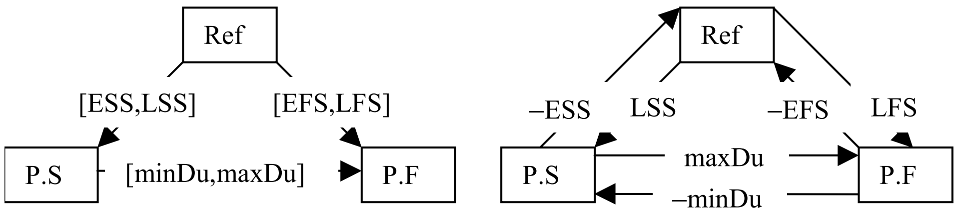


Example 1

```
(P1  [[_,_],[_,390],[50,370],Ref]
   do-seq-ordered ((P2 [[_,_],[_,_],[70,100],_]),
(P3  [[_,_],[_,_],[120,150],_]),
                     (P4  [[_,_],[_,_],[130,160],_])))

(P2 do-parallely ((P5 [[_,_],[_,_],[90,_],_]),
 (P6  [[40,_],[_,_],[_,_],Ref])))

(P3 do-cyclically ((P7 [[_,_],[_,_],[20,_],_]
retry=[10,_] exec=[5,_])))

(P4 do-arbitrarily ((P8 [[_,_],[_,_],[120,160],_]),
  (P9  [[_,_],[_,_],[20,_],_])))

(P8 do-seq-unordered ((P10 [[_,_],[_,_],[90,_],_]),
    (P11  [[_,_],[_,_],[80,_],_])))
```


Constraint vs Distance Graph

- Simple Temporal Problem [Dechter, Meiri, Perl, 1992]
- Constraint graph (left) and Distance graph (right) of plan activation
  (P [[ESS, LSS], [EFS, LFS], [minDu, maxDu], Ref]).



Properties Checks

- **Consistency**
  – Does not contain negative cycles
- **Minimally**
  – a *Plan Activation (PA) minimal* if and only if the constraint network, given by the three intervals of the associated time annotation, is minimal.

  – Dechter and colleagues : A constraint network is minimal if it is *tighter* than any *equivalent* constraint network ...

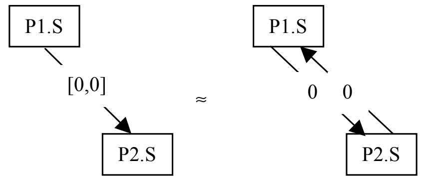
Check: Inconsistencies

- Specification of Control Flow
- Parent-Child Relation
- Links to the Time Line

Specification of Control Flow

- Sequential execution of plans with defined order

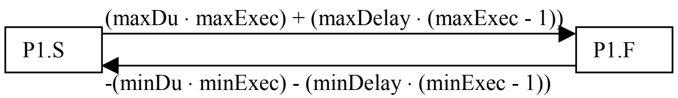
- Parallel execution of plans

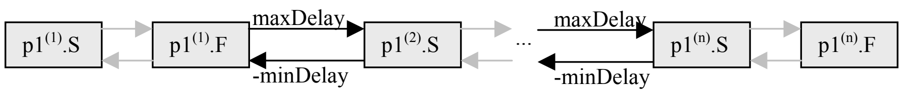


Specification of Control Flow

- Arbitrary execution of plans

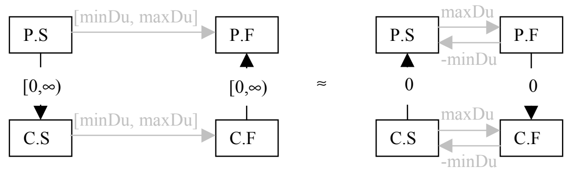
- Cyclic execution of plans



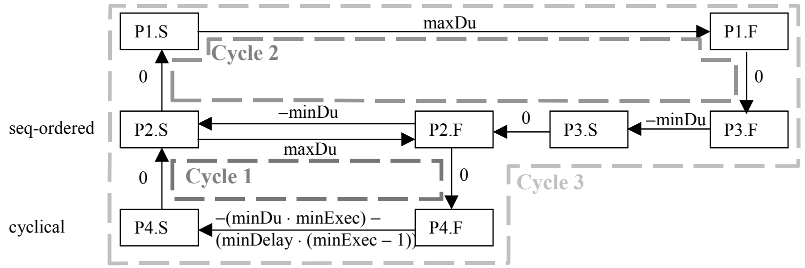


Parent-Child Relation

(a)

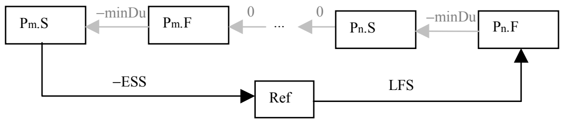


(b)

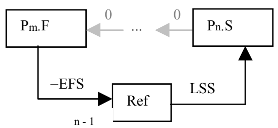


Links to the Time Line

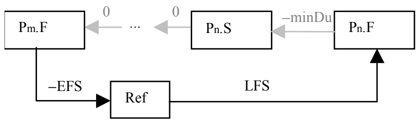
- Sequential execution of plans



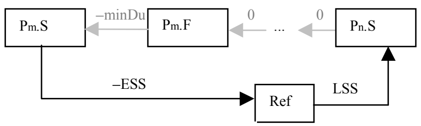
Inconsistent, if: $P_m.ESS+\sum_{i=m}^{n} P_i.minDu > P_n.LFS$



Inconsistent, if: $P_m.EFS+\sum_{i=m+1}^{n-1} P_i.minDu > P_n.LSS$



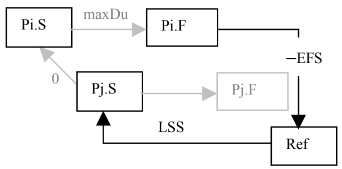
Inconsistent, if: $P_m.EFS+\sum_{i=m+1}^{n} P_i.minDu > P_n.LFS$



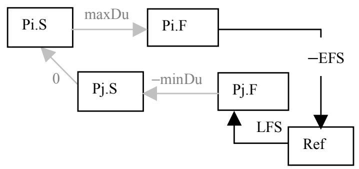
Inconsistent, if: $P_m.ESS+\sum_{i=m}^{n-1} P_i.minDu > P_n.LSS$

Links to the Time Line

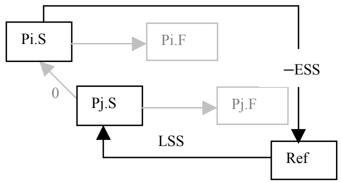
- Parallel Execution of Plans



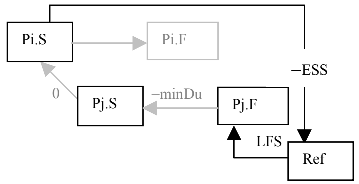
Inconsistent if: $P_i.EFS – P_i.maxDu > P_j.LSS$



Inconsistent if: $P_i.EFS – P_i.maxDu > P_j.LFS – P_j.minDu$



Inconsistent if: $P_i.ESS > P_j.LSS$



Inconsistent if: $P_i.ESS > P_j.LFS – P_j.minDu$

Links to the Time Line

- **Arbitrary Execution of Plans**
  – PA + ∞
  – no negative cycle

- **Cyclic Execution of Plans**
  – cycle in cycle
  – each single cyclic plan may itself become inconsistent

Non-Minimalities

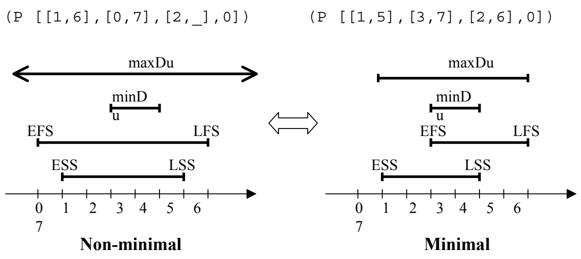
- **Single Plan Activation**
  – $ESS \le EFS$ and $LSS \le LFS$
  – $MAXIMUM\ (0, EFS – LSS) \le minDu \le MINIMUM\ (EFS – ESS, LFS – LSS)$
  – $MAXIMUM\ (EFS – ESS, LFS – LSS) \le maxDu \le LFS – ESS$

Meaning

– 1: SI {*equal, overlaps, meets, before*} FI
– 2,3: depending SI, FI --> yields bounds for DI

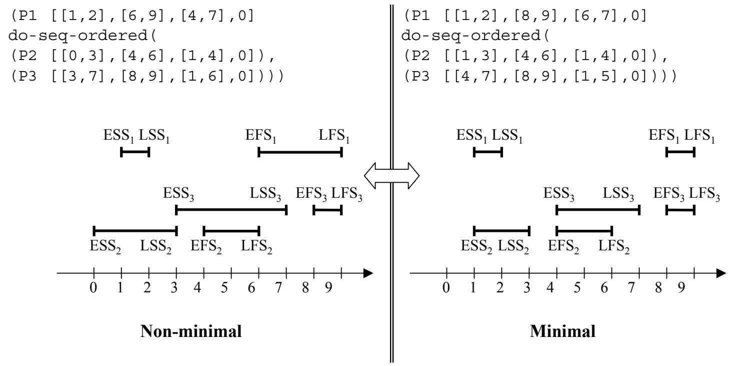
Non-Minimalities

- PA

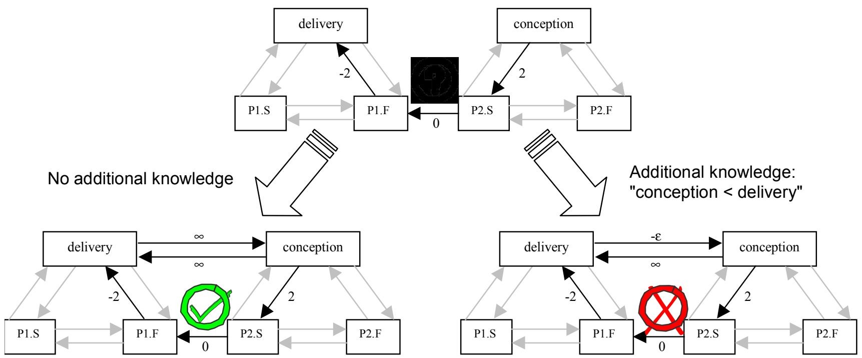


Non-Minimalities

- Network of PA



Verification with Multiple Time Lines



Verification Methods

- **Floyd-Warshall's *all-pairs-shortest-paths algorithm***
  - detect inconsistencies within a network of PAs

$$
\begin{array}{l}
\textit{for } i,j = 1 \textit{ to } n \textit{ do } d_{ij} \leftarrow a_{ij}; \\
\textit{for } i = 1 \textit{ to } n \textit{ do } d_{ii} \leftarrow 0; \\
\quad \textit{for } k = 1 \textit{ to } n \textit{ do} \\
\quad\quad \textit{for } i,j = 1 \textit{ to } n \textit{ do} \\
\quad d_{ij} \leftarrow \min(d_{ij}, d_{ik} + d_{kj});
\end{array}
$$

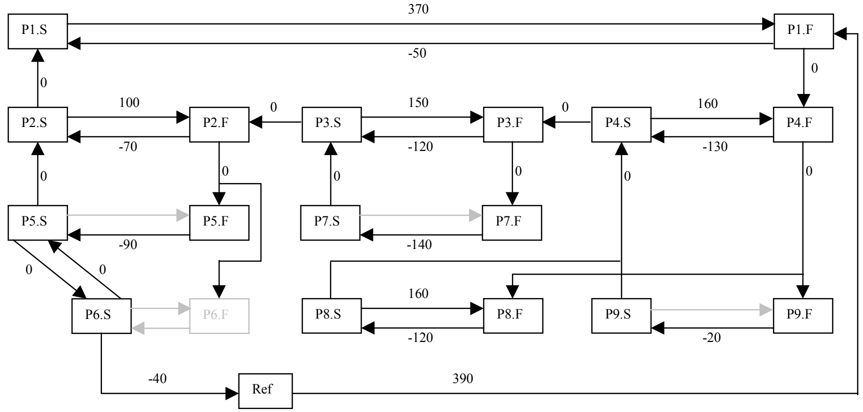
Matrix

(a) weights a_{ij};

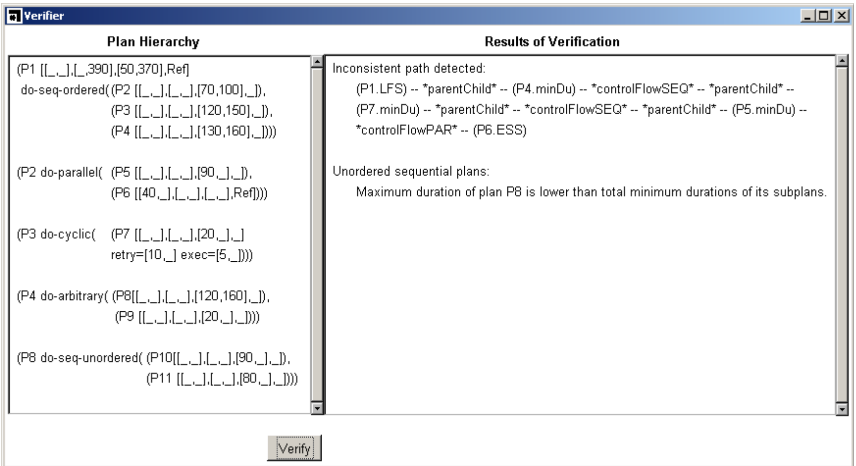
(b) minimum distances d_{ij}; for plan P2 and subplans of Example

		P2.S	P2.F	P5.S	P5.F	P6.S	P6.F	Ref
(a)	P2.S	0	100	701	701	701	701	701
	P2.F	-70	0	701	0	701	0	701
	P5.S	0	701	0	701	0	701	701
	P5.F	701	701	-90	0	701	701	701
	P6.S	0	701	0	701	0	701	-40
	P6.F	701	701	701	701	701	0	701
	Ref	701	701	701	701	701	701	0
(b)	P2.S	0	100	10	100	10	100	-30
	P2.F	-90	0	-90	0	-90	0	-130
	P5.S	0	100	0	100	0	100	-40
	P5.F	-90	10	-90	0	-90	10	-130
	P6.S	0	100	0	100	0	100	-40
	P6.F	611	701	611	701	611	0	571
	Ref	611	701	611	701	611	701	0

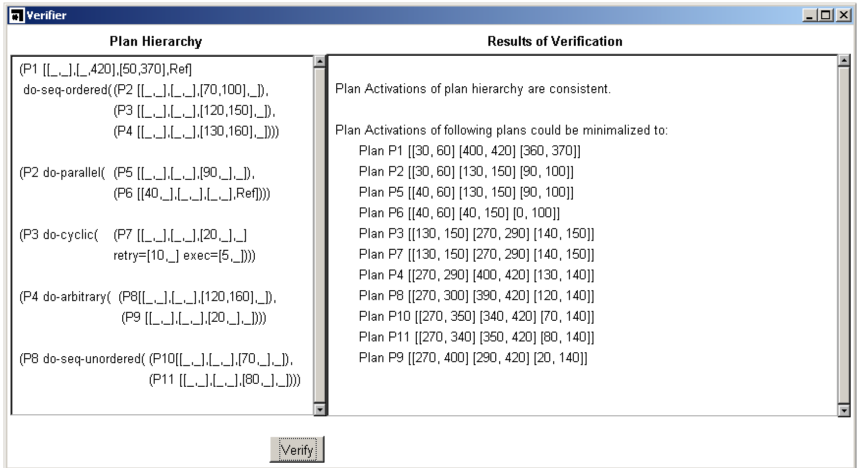
Example 1



Example 1



Example 1



To Do ---> D3 (1/2)

- **Validation- Medical Knowledge**
  - Mar and Kitty joined Session in A'dam
    - Jaundice and Diabetes
    - send them to Peter, sends comments back
    - start second week of January
  - Topics
    - Properties - Indicators
      - plus Informal Medical Reasoning
    - Intentions
      - plus Informal Medical Reasoning
    - (possible in KIV: Saftey Models)
  - Output: Summary

TO DO ---> D3 (2/2)

- **Verification Part**
  - Input:
    - A'dam Properties
    - Vienna Properties: Condition Properties,Temporal Dimensions
  - Michael: What is suitable in KIV
  - Output:
    - Selection of these Properties
    - Comments on non-selected properties
    - Distinguish on which level: original, Asbru, KIV
    - Annotate which level
  - Deadline April 2002

Deliverables (D2, D3 : WP 2)

- **WS**
  - responsible: Andi (Silvia)
- **D2**
  - Operational Semantics
    - IDPT-Submission plus extensions
    - DL: March 15, 2002
    - responsible: Michael (Andi)

Deliverables (D2, D3 : WP 2)

- **D3**
  - Properties
  - 2 Parts
    - Conceptual
      - medical (original) = Kitty (Mar)
      - general description
        - » free text and examples
        - » priority list
      - DL: March 15, 2002
    - Implementation
      - semantics (Asbru) = free text: Silvia (Michael)
      - formulized 1-2 properties in KIV (Michael)
      - DL: March 15, 2002