submitted to: Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan, August 23-29, 1997.

Is Skeletal Planning in Real-World, High-Frequency Domains Possible?

Silvia Miksch

Vienna University of Technology, Institute of Software Technology, Resselgasse 3/E188, A-1040 Vienna, Austria

Werner Horn

Austrian Research Institute for Artificial Intelligence (OFAI), Schottengasse 3, A-1010 Vienna, Austria &
Department of Medical Cybernetics and Artificial Intelligence, University of Vienna, Freyung 6, A-1010 Vienna, Austria

Yuval Shahar

Section on Medical Informatics, Stanford University, Stanford, CA 94 305 - 5479, USA

Christian Popow

NICU, Department of Pediatrics, University of Vienna, Währinger Gürtel 18-20, A-1090 Vienna, Austria

Franz Paky

Department of Pediatrics, Hospital of Mödling, Weyprechtgasse 12, A-2340 Mödling, Austria

Peter Johnson

The Sowerby Unit for Primary Care Informatics, University of Newcastle, 21 Claremont Place, Newcastle Upon Tyne,
NE2 4AA, UK

Abstract

Skeletal plans are a powerful way to reuse existing domain-specific procedural knowledge. In the **Asgaard** project, a set of tasks that support the design and the execution of skeletal plans by a human executing agent other than the original plan designer were created. The underlying requirement to develop task-specific problem-solving methods is a *modeling language*. Therefore, within the Asgaard project, a time-oriented, intention-based language, called **Asbru**, was developed. During the design phase of plans, Asbru allows to express durative actions and plans caused by durative states of an observed agent. The intentions underlying these plans are represented explicitly as temporal patterns to be maintained, achieved or avoided. The Asgaard project and the Asbru language were designed for low-frequency domains. We proved the applicability of the Asbru language in the real-world, high-frequency environment of neonatal intensive care units (NICUs). The knowledge-base of **VIE-VENT**, an open-loop monitoring and therapy planning system for artificially ventilated newborn infants, is enhanced and formulated in the Asbru syntax. We show the benefits and limitations of the time-oriented, skeletal plan representation to be applicable in real-world, high-frequency domains.

Keywords: planning, temporal reasoning, skeletal plans, high-frequency domains

1 Introduction: Improving Quality of Heath Care by Automated Support of Plan Design and Execution

Several domains require the capability of analyzing and interpreting a large number of time-oriented data in combination with temporal domain knowledge. The data are in general drawn on an irregular time grid, since timing of observations depends on some individual factors. The domain knowledge is often not available in a form which gives a proper time annotation.

We are motivated by the needs for knowledge-based support in the medical domain. Health care providers are faced with two problems: (1) the information overload resulting from modern equipment, and (2) improving the quality of health care through increased awareness of proper disease management techniques. **Clinical protocols** and **guidelines** should solve the difficulties. Clinical guideline refers to a general principle by which to determine a course of actions and clinical protocol refers to a general class of therapeutic interventions.

Appropriate clinical protocols are available for a very limited class of clinical problems only. They are not adjusted to the patient data-management system, they are partly vague and incomplete concerning their intentions and their temporal and context-dependent representation, and most often they are outdated after being developed. Extracting and formulating the knowledge structure for clinical protocols is a non trivial task. It has to made explicit the context which is implicit in the protocols.

During the last years, there have been several efforts to create automated reactive planners to support the process of protocol-based care over a significant period of time. One of these efforts is the **Asgaard** project, in which tasks to support the design and the execution of skeletal plans are defined [Shahar *et al.,* 1996]. Within the Asgaard project, a time-oriented and intention-based language, called **Asbru**, was developed to represent protocols. The Asgaard project was applied to the low-frequency domain for controlled observation and treatment of noninsulin-dependent gestational diabetes mellitus.

Treatment planning with respect to real-world medical environments in high-frequency domains entails different problem features. The available data occur at high observation frequencies (e.g., sampling rate every 10 seconds), at various regularities (e.g., continuously or discontinuously assessed data), and at various types (e.g., qualitative or quantitative data). The domain is highly reactive in the sense that immediately actions and reactions and particular crisis management are needed. Experiences within the development of **VIE-VENT** system [Miksch *et al.*, 1996], an open-loop, monitoring and therapy planning system for artificially ventilated newborn infants, indicated the non trivial data analysis problems. VIE-VENT was especially designed for practical use under real-time constraints at neonatal intensive care units (NICUs). However, the performance and the acceptance of VIE-VENT could be improved significantly incorporating protocol-based concepts.

The aim of the paper is to prove the applicability of the Asbru language in the high-frequency domain of VIE-VENT, namely the mechanical ventilation of newborn infants.

In the section 2 we explain why other representations of skeletal plans fail to meet specific requirements. The essential components of the Asbru language are introduced in section 3. Section 4 characterizes the medical problem and illustrates how it is represented in Asbru. In the last section we evaluate Asbru's applicability identifying it's strengths and limitations.

2 The Need for a New Representation of Skeletal Plans

A common strategy for the representation and the reuse of domain-specific procedural knowledge is the representation of that knowledge as a library of skeletal plans. Skeletal plans are plan schemata at various levels of detail that capture the essence of the procedure, but leave room for execution-time flexibility in the achievement of particular goals [Friedland and Iwasaki 1985]. Thus, they are usually reusable in different contexts.

On the one hand, starting in the 1970s researchers in medicine and medical informatics have recognized the importance of protocol-based care to ensure a high quality of care. A standard procedural language, known as the Arden syntax [Hripcsak *et al.*, 1994], encodes situation-action rules. Developers of the Arden syntax have promoted this Pascal-like language because of the pressing needs to facilitate exchange of guidelines among health-care institutions using existing software technology. The Arden syntax has significant limitations: The language currently supports only atomic data types, it lacks a defined semantic for making temporal comparisons or for performing data abstraction, and it provides no principled way to represent clinical guidelines that are more complex than individual situation-action rules.

On the other hand, computer-oriented knowledge interchange languages (e.g., KIF [Genesereth and Fikes, 1992]), ontologies or models for knowledge sharing (e.g., [Guarina and Giaretta 1995]), and general purpose languages to support planning (e.g., PROPEL language [Levinson 1995], O-Plan2 [Tate *et al.,* 1994]) were introduced. These traditional (plan-execution) representations have significant limitations and are not applicable in dynamically changing environments: (1) they assume instantaneous actions and effects; (2) actions often are continuous (durative) and might have delayed effects and temporally-extended goals [Bacchus and Kabanza, 1996]; (3) there is uncertainty and variability in the utility of available actions; (4) unobservable underlying processes determine the observable state of the world; (5) a goal may not be achievable; (6) sequential, concurrent, and cyclical execution of plans is necessary. The requirements of plan specifications in clinical domains are often a superset of the requirements in typical problem domains used in planning research.

A sharable skeletal-plan-execution language needs to be expressive with respect to temporal annotations and needs to have a rich set of parallel, sequential, and iterative operators. Thus, it should enable designers to express complex procedures in a manner similar to a real programming language (although typically on a higher level of abstraction). The language, however, also requires well-defined semantics for both the prescribed actions and the task-specific annotations, such as the plan's intentions and effects, and the preferences (e.g., implicit utility functions) underlying them. Thus, the executing agent's (e.g., the physician's) actions can be better supported, leading to a more flexible dialog and, in the case of the clinical domains, to a better acceptance of automated systems for guideline-based care support. Clear semantics for the task-specific knowledge roles also facilitate acquisition and maintenance of these roles.

With these requirements in mind, a syntax of a time-oriented, intention-based language, called Asbru, was designed [Miksch *et al.,* 1997]. The Asbru language is part of the Asgaard project [Shahar *et al.,* 1996].

3 The Asgaard/Asbru Project

The **Asgaard** project outlined some useful task-specific problem-solving methods to support both designer and executor of skeletal plans [Shahar *et al.,* 1996]. The project was particular oriented on medical problems with low-frequency data. It mainly supports therapeutic issues. The problem-solving methods are divided in tasks, which are performed during *design time* and *execution time* of a skeletal plan. During design time the relevant tasks are: (1) verification and (2) validation. During execution time the relevant tasks are: (1) applicability of the plan to a particular state of the world, (2) guidance in proper execution of that plan, (3) monitoring of the execution process, (4) assessment of the results of the plan, (5) critiquing the execution process and its results, and (6) assistance in the modification of the original plan.

3.1 Asbru: Language for representing skeletal plans

The underlying requirement to perform these problem-solving methods is a *modeling language.* Therefore, within the Asgaard project, a time-oriented and intention-based[1] language, called **Asbru**, was developed to represent and to annotate durative skeletal plans based on the task-specific ontology [Miksch *et al.,* 1997].

During the design phase of plans, Asbru allows to express durative actions and plans caused by durative states of an observed agent (e.g., many actions and plans need to be executed in parallel or every particular time point). These plans are combined with intentions of the executing agent. They are uniformly represented and organized in the *guideline-specification library.* During the execution phase an applicable plan is instantiated with distinctive arguments and state-transition criteria are added to execute and reason about different tasks.

A plan consists of a name, a set of arguments, including a time annotation (representing the temporal scope of the plan), and five components: **preferences**, **intentions**, **conditions**, **effects**, and a **plan body**. The general arguments, the time annotation, and all components are optional. A plan may consist of several subplans. Subplans are plans at a more detailed level. However, subplans have the same structure as plans (compare the example in Figure 2).

Preferences bias or constrain the selection of a plan to achieve a given goal and express a kind of behavior of the plan (e.g., implicit utility functions, a specification of prohibited or obligatory resources).

Intentions are high-level goals at various levels of the plan, an annotation specified by the designer, which supports tasks such as critiquing and modification. Intentions are temporal patterns of executing-agent actions and external-world states that should be maintained, achieved, or avoided. Intentions may consist of:

(1) *Intermediate states:* the state(s) that should be maintained, achieved, or avoided during the applicability of the plan (e.g., the blood-gas levels are slightly below or slightly above the target range);
(2) *Intermediate actions:* the action(s) that should take place during the execution of the plan (e.g., minimize level of mechanical ventilation);
(3) *Overall state patterns:* the overall pattern of states that should hold after finishing the plan (e.g., patient had less than one high blood-glucose value per 30 minutes); and
(4) *Overall action patterns:* the overall pattern of actions that should hold after finishing the plan (e.g., avoid hand-bagging).

Conditions are temporal patterns, sampled at a specified frequency, that need to hold at particular plan steps to induce a particular state transition of the plan instance. Different conditions are specified that enable transition from one plan state into another. A plan is completed when the completed conditions become true, otherwise the plan's execution suspends or aborts. Aborting a plan's execution is often due to a failure of the plan or part of it. The following conditions are distinguished:

(1) *Filter-preconditions* need to hold initially if the plan is applicable, but can not be achieved (e.g., ventilator support required). They are necessary for a state to become *possible*;
(2) *Setup-preconditions* need to be achieved to enable a plan to start (e.g., inspiratory oxygen concentration F_iO_2 is less than 80%) and allow a transition from a *possible* plan to an *activated* plan;
(3) *Suspend-conditions* determine when a started plan has to be suspended - certain conditions (*protection intervals)* need to hold (e.g., blood gas has been extremely above the target range for at least five minutes);
(4) *Abort-conditions* determine when a started, suspended, or restarted plan has to be aborted (e.g., the increase of the blood-gas level is too-fast for at least 30 seconds);
(5) *Complete-conditions* determine when a started or restarted plan has to be completed successfully (e.g., returning to spontaneous breathing); and

[1] The meaning of *intentions* in general and for planning tasks in particular has been examined in philosophy [Bratman 1987] and in Artificial Intelligence [Pollack 1992]. In Asbru, intentions are viewed as temporally extended goals at various abstraction levels [Bacchus and Kabanza, 1996].

(6) *Restart-conditions* determine when a suspended plan has to be restarted (e.g., blood gas level is back to normal or is only slightly increased).

Effects either describe the functional relationship between the plan arguments and measurable parameters or specify an overall effect of a plan. Effects have a likelihood annotation—a probability of occurrence.

The **plan body** is a set of plans to be executed either *sequentially*, *concurrently*, or *cyclically*. A sequential plan specifies a set of plans that are executed in sequence; for continuation, all plans included have to be completed successfully. Concurrent plans can be executed in parallel or in any order. A cyclical plan includes a plan that can be repeated and optional temporal and continuation arguments that can specify its behavior.

A plan in the guideline-specification (plan) library is composed hierarchically. A decomposition of a plan into its subplans is always attempted by the execution interpreter, unless the plan is not found in the guideline-specification library, thus representing a nondecomposable plan (informally, an *action* in the classical planning literature). This can be viewed as a ”semantic“ stop-condition. Such a plan is referred to the agent for execution, which may result in an interaction with a user or an external call of a program.

During the execution phase, an applicable plan is instantiated. At execution time, a set of mutually exclusive *plan states* describes the status of a plan instance. *State-transition criteria* specify transitions between states. The set of states is {*possible*, *activated*, *completed*, *suspended*, *aborted*}.

3.2 Temporal Patters and Annotations

Intentions, world states, and prescribed actions are temporal patterns. A temporal pattern is either a *parameter preposition*—a parameter (or its abstraction), its value, a context, and a time annotation (e.g., the *state* abstraction of the blood-gas parameter is NORMAL, as defined in the context of weaning therapy, during a certain time period)—, a *combination* of multiple parameter propositions, or a *plan-state* associated to an instantiated plan (plan pointer) and a time annotation.

The time annotations used allow to represent uncertainty in starting time, ending time, and duration [Dechter *et al.*, 1994; Rit 1986]. The time annotation supports multiple time lines (e.g., different zero-time points and time units) by providing *reference annotations*. Temporal shifts from the reference annotation are defined to represent the uncertainty in starting time, ending time, and duration, namely earliest starting shift (ESS), latest starting shift (LSS), earliest finishing shift (EFS), latest finishing shift (LFS), minimal duration (MinDu), and maximal duration (MaxDu). The temporal shifts are associated with time units (e.g., minutes, days). Thus, a temporal annotation is written as ([ESS, LSS], [EFS, LFS], [MinDu, MaxDu], REFERENCE). ESS, LSS, EFS, LFS, MinDu, and MaxDu can be "unknown" or "undefined" to allow incomplete time annotation.

To allow temporal repetitions, sets of cyclical time points and cyclical time annotations are defined. Short-cuts are used to allow to start a plan immediately at the current time (using the symbol *now*), to use the activation of a plan as reference point (using the symbol *self*), or to allow that a condition holds during the span of time over which the plan is executed (using the symbol *).

For example, the parameter proposition "the level of blood gas is normal or above the normal range in the context of controlled ventilation-therapy for at least three hours, using the activation of the plan as reference point", is written in Asbru as:

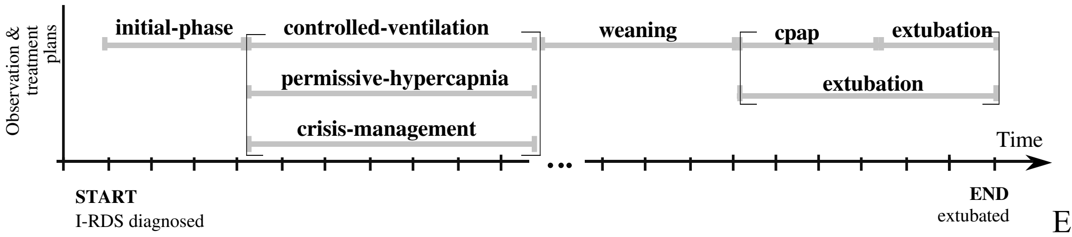
```
(STATE(BG) (OR NORMAL ABOVE-NORMAL)
   controlled-ventilation
   [[_, _], [_, _], [180 MIN,_], *self*])
```

4 Asbru in High-Frequency Domains

In the following, we will show how a treatment protocols for artificial ventilated newborn infants will be formulated in the Asbru syntax.

Artificial ventilation has greatly contributed towards the improvement of the mortality and morbidity of premature newborn infants [Perlman *et al.*, 1995]. Enhanced knowledge about the pathophysiological mechanisms of barotrauma and oxygen toxicity led to the development of patient-tailored strategies of mechanical ventilation and helped to reduce harmful side effects of respirator therapy. However, standardized clinical treatment protocols for infants' respiratory distress syndrome (I-RDS) are partly vague and incomplete concerning their intentions and their temporal and context-dependent representation. Therefore, we acquired the implicit or not mentioned intentions and conditions from domain experts.

Figure 1 illustrate the top-level treatment protocol for I-RDS. After I-RDS is diagnosed, a plan dealing with limited monitoring possibilities is activated, called **initial-phase**. Then follows, depending on the severity of the disease, three different kinds of plans, **controlled-ventilation**, **permissive-hypercapnia**, or **crisis-management**. Only one plan at time can be activated, however the order of execution and the activation frequency of the three different plans are depending on the severity of the disease. The brackets in Figure 1 illustrate this circumstance. Additionally, it is important to continue with the plan **weaning** only after a successful completion of the plan **controlled-ventilation**. After a successful execution of the plan **weaning**, the extubation should be initiated. The extubation can be either a single plan **extubation** or a sequential execution of the subplans **cpap** and **extubation**.



EMBED

Figure 1: Treatment protocol for infants' respiratory distress syndrome (I-RDS).

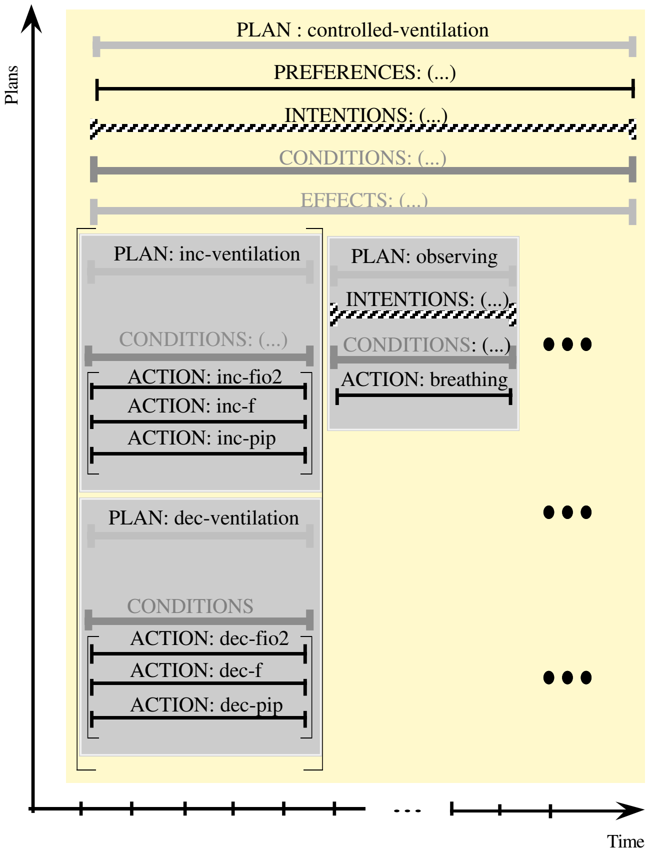


Figure 2: Subplans of the treatment protocol for infants' respiratory distress syndrome (I-RDS).

Figure 2 is a zoom-in of Figure 1 showing the subplan **controlled-ventilation** and its possible subplans using the Asbru language. In Figure 2 two notations are used: uppercase letters followed by colons (":") indicate elements of Asbru and lowercase letters indicate particular plans, subplans, or actions. The plan **controlled-ventilation** is decomposed in two subplans, decreasing or increasing the ventilation setting (plan **inc-ventilation** and **dec-ventilation**) and the plan **observing**. The frequency of these two plans cannot be specified in advance. The number of activation periods depends on the health condition of the patient. The points (●●●) in Figure 2 indicate these unknown repetitions. The subplan **inc-ventilation** is decomposed in three subplans, **inc-fio2**, **inc-f**, or **inc-pip**. These three subplans are nondecomposable plans (actions). Additionally, only one of these three actions can be activated at each time period, which is illustrated with the brackets. The same decomposition holds for the subplan **dec-ventilation** with **dec-fio2**, **dec-f**, or **dec-pip**, respectively.

The following specification shows the treatment protocol I-RDS in Asbru syntax:

```
(PLAN I-RDS-therapy
   <... parts deleted ...>
(DO-ALL-SEQUENTIALLY
   (initial-phase)
   (one-of-controlled-ventilation)
   (weaning)
   (one-of-cpap-extubation)))

(PLAN one-of-controlled-ventilation
   <... parts deleted ...>
(DO-SOME-ANY-ORDER
   (controlled-ventilation)
   (permissive-hypercapnia)
   (crisis-management)
   CONTINUATION-CONDITION controlled-
   ventilation))
```

The continuation condition specifies which subplans must be completed successfully to continue with the next plan. In the subplan **one-of-controlled-ventilation** the CONTINUATION-CONDITION guarantees that it is only possible to start the plan **weaning**, when plan **controlled-ventilation** had been completed successfully. The alternative subplans **permissive-hypercapnia** or **crisis-management** are applied too, however the whole plan **one-of-controlled-ventilation** will never be completed successfully without a final successful completion of subplan **controlled-ventilation**.

```
(PLAN controlled-ventilation
(PREFERENCES (SELECT-METHOD BEST-FIT))
(INTENTION:INTERMEDIATE-STATE
   (MAINTAIN STATE(BG) NORMAL
         controlled-ventilation *))
(INTENTION:INTERMEDIATE-ACTION
   (MAINTAIN STATE(RESPIRATOR-SETTING) LOW
         controlled-ventilation *))
(SETUP-PRECONDITIONS
   (FiO2 (<= 80) * *now*)
   (PIP (<= 30) * *now*)
   (f (<= 80) * *now*))
(ACTIVATED-CONDITIONS AUTOMATIC)
(ABORT-CONDITIONS ACTIVATED
   (OR (FiO2 (> 80) * [[_, _], [_, _],
         [30 SEC, _], *self*])
       (PIP (> 30) * [[_, _], [_, _],
         [30 SEC, _], *self*])
       (f (> 80) *  [[_, _], [_, _],
         [30 SEC, _], *self*])
       (RATE(BG) TOO-STEEP
          controlled-ventilation
          [[_, _], [_, _], [180 MIN,_],
  *self*]))
   (SAMPLING-FREQUENCY 10 SEC))
(COMPLETE-CONDITIONS
   (FiO2 (<= 50) * [[_, _], [_, _],
      [180 MIN, 180 MIN], *self*])
   (PIP (<= 23) * [[_, _], [_, _],
      [180 MIN, 180 MIN], *self*])
   (f (<= 60) *  [[_, _], [_, _],
      [180 MIN, 180 MIN], *self*])
   (patient (NOT DYPNOEIC) * [[_, _], [_, _],
      [180 MIN, 180 MIN], *self*]))
   (STATE(BG) (OR NORMAL ABOVE-NORMAL)
      controlled-ventilation
      [[_, _], [_, _], [180 MIN,_], *self*])
   (SAMPLING-FREQUENCY 10 MIN))
(DO-ALL-SEQUENTIALLY
   (one-of-increase-decrease-ventilation)
   (observing)))
```

The intentions of subplan **controlled-ventilation** are to maintain a normal level of the blood-gas values and the lowest level of mechanical ventilation (as defined in the context of controlled ventilation therapy) during the span of time over which the subplan is executed. This subplan is activated immediately, if inspiratory oxygen concentration FiO2 ≤ 80%, peak inspiratory pressure PIP ≤ 30, and inspiratory frequency f ≤ 80. The subplan must be aborted, if FiO2 > 80%, PIP > 30, f > 80, or the increase of the blood-gas level is too steep (as defined in the context of controlled ventilation-therapy) for at least 30 seconds. The sampling frequency of the abort condition is 10 seconds. The subplan is completed successfully, if FiO2 ≤ 50%, PIP ≤ 23, f≤ 60, the patient is not dyspnoeic, and the level of blood gas is normal or above the normal range (as defined in the context of controlled ventilation-therapy) for at least three hours. The sampling frequency of the complete condition is 10 minutes. The body of the subplan **controlled-ventilation** consists of a sequential execution of the two subplans **one-of-increase-decrease-ventilation** and **observing**.

5 Benefits and Limitations

Applying the Asbru language to represent time-oriented skeletal plans is a very effective tool to acquire the domain knowledge needed in a structured way. Clear semantics for the task-specific knowledge facilitate acquisition and maintenance. Asbru places a particular emphasis on an expressive representation for time-oriented actions and world states in combination with the underlying intentions as temporal patterns to be maintained, achieved or avoided. It allows to use different granularities and reference points to represent multiple time lines. Asbru's representation includes the duration of actions, their success or failure, and allows time annotation of events, actions/plans, and world states with uncertainty in their appearances. Asbru has a rich set of sequential, concurrent, or cyclical operators, which enable to express complex procedures. Preferences, intentions, conditions, effects, and actions, are specified on various detailed levels depending of their appearances and evidences. All components of Asbru are optional. The expressive representation results in an uniformly represented and organized plan-specification library.

Nevertheless, the expressive representation of Asbru still has some limitation. In general, the medical experts were hard pressed to fill all the slots of the Asbru language and there was very little procedural, pre-compiled knowledge (protocols) found. It seemed over-specified compared to flow diagram or algorithm type of protocol, which the medical experts are used to work with. The medical experts found it hard to define their intentions or metrics by which to measure the success or failure of an individual action. The best experts often have their own personal metrics by which they judge the success or failure of an action they have taken. But these metrics are usually very arbitrary, based on empirical factors, and difficult to extract from the expert. Further, these metrics may differ from one expert to another quite widely. The expert cannot usually be pushed into providing the evidence for these metrics.

The flexibility of the time annotation is one of the main benefits of Asbru. The choice to select different reference points is heavily used in VIE-VENT's plans. Further, the ability to define different sampling intervals as shown in the above example is essential in high-frequency domains. On the one hand, it is the only way to be able to react fast (within 30 seconds) in critical situations. On the other hand, it allows to check for long-term stability on a 10 minutes sampling frequency with appropriate filtering of data. In summary, the acquisition of the temporal patterns and time annotations needed is still quite difficult. In real-world high-frequency domains, the temporal dimensions are vague or unknown most often. The position of a measurement in the sequence of time-ordered data influences the abstraction of higher-level concepts, namely, recent measurements are more important than historical measurements. Additionally, information about the frequency and the duration of temporal patterns in the past are needed (e.g., "three episodes of extremely high levels of blood-gas values during the last 3 hours occurred, the episodes lasted at least 5 minutes.").

Asbru provides sequential, concurrent, or cyclical operators to represent plans. However, it is quite difficult to cope with all possible orders of plan execution and all the exception conditions that might arise. Clinical protocols are a way of pre-compiling decisions that must be made, in which experts knowledge is distilled into a form of procedural knowledge. The trouble is that this by necessity can only cover a small subset of the possible situations and possible paths through. Physicians use a lot of background knowledge in thinking about what is the best thing to do, which is hard to acquire and to represent as a skeletal plan (or computerized protocol). Additionally, a clear semantic for alternative parts of plans (protocols) is needed. This shows also the vague—nearly missing—plan revision strategies in Asbru.

The iterating condition in the definition of a cyclical plan in Asbru depends on a temporal interval (pattern). The iterating condition should also be a plan state, to allow, i.e., a plan which iterates a component to decide which sub-component to do. If the sub-component abandons, or the duration state is violated, it needs to be able to iterate at that point. The iterations may occur at random times. Additionally, it should be possible to count activation of plan states to express particular orders of plan execution (e.g., plan A aborted three times, then plan A completed successfully, then start plan B).

6 Conclusion

Representing complex execution plans, such as clinical protocols, and the intentions underlying them in a sharable and acquirable form is imperative for useful, flexible automated assistance in the execution of these plans. In the manifold domains of clinical medicine and intensive care, such a task-specific representation is crucial for dissemination of modern clinical knowledge, since the use of clinical protocols will set up standards in the provision of high quality of care.

We outlined the basic concepts of an effective time-oriented representation of skeletal plans, called Asbru and proved the applicability of the Asbru syntax in the context of ventilator management in neonatal intensive care, which is based on the accurate analysis of high-frequency data.

The different knowledge-roles represented in the Asbru language facilitate the acquisition of knowledge, however filling all required slots is a demanding task. Asbru still includes some vagueness in a few definitions, like cyclical time annotations, iterating plans, the semantic of the state-transition criteria, scoping rules, and plan revision strategies. But its main strengths are the explicit representation of the intentions of the defined plans, the transparency of the time annotation of these plans, and the flexibility in the usage of different reference points and different time granularities.

References

[Bacchus and Kabanza, 1996] F. Bacchus and F. Kabanza. Planning for temporally extended goals. *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, AAAI Press/The MIT Press, Menlo Park, CA,1215–1222, 1996.

[Bratman 1987] M.E. Bratman. *Intention, plans and practical reason*. Cambridge, MA: Harvard University Press, 1987.

[Dechter *et al.*, 1994] R. Dechter, L. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1-3), 1991.

[Friedland and Iwasaki 1985] P. E. Friedland and Y. Iwasaki. The concept and implementaion of skeletal plans. *Automated Reasoning*, 1(2):161-208, 1985

[Genesereth and Fikes, 1992] Michael R. Genesereth and Richard E Fikes, *Knowledge interchange format, version 3.0 reference manual*. Computer Science Department, Stanford University, Technical Report Logic-92-1, 1992.

[Guarina and Giaretta 1995] N. Guarina and P. Giaretta. Ontologies and knowledge bases. N. J. I. Mars, editor. *Towards very large knowledge base*, Amsterdam: IOS Press, 1995.

[Hripcsak *et al.*, 1994] G. Hripcsak, P. Ludemann, T. A. Pryor, O.B. Wigertz, and P.D. Clayton. Rationale for the Arden syntax. *Computers and Biomedical Research*, 27:291-324, 1994.

[Levinson 1995] Richard Levinson. A general programming language for unified planning and control. *Artificial Intelligence*, 76(1-2):319-375, 1995.

[Miksch *et al.*, 1996] Silvia Miksch, Werner Horn, Christian Popow, and Franz Paky. Utilizing Temporal Data Abstraction for Data Validation and Therapy Planning for Artificially Ventilated Newborn Infants. *Artificial Intelligence in Medicine*, 8(6):543-576, 1996.

[Miksch *et al.*, 1997] Silvia Miksch, Yuval Shahar, and Peter Johnson. Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans. *7th Workshop on Knowledge Engineering: Methods & Languages*, Milton Keynes, UK, January 1997.

[Perlman *et al.*, 1995] M. Perlman, O. Claris, Y. Hoa, P. Pandit, H. Whyte, M. Chipman, and P. Lui, Secular Changes in the outcomes to eighteen to twenty-four months of age of extremely low birth weight infants with adjustment for changes in risk factors and severity of illness. *Journal of Pediatrics*, 126:75-87, 1995.

[Pollack 1992] Martha Pollack. The use of plans. *Artificial Intelligence*, 57(1):43-68, 1992.

[Rit 1986] J.-F. Rit. Propagating temporal constraints for scheduling. *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, Morgan Kaufmann, Los Altos, CA, 383-8, 1986.

[Shahar *et al.*, 1996] Yuval Shahar, Silvia Miksch, and Peter Johnson. A task-specific ontology for design and execution of time-oriented skeletal plans. Brain Gaines, editor. *Proceedings of the Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Alberta, Canada, November, 1996.

[Tate *et al.*, 1994] A. Tate, B. Drabble, and R. Kibry. O-Plan2: an open architecture for command, planning, and control. M. Zweber and M.S. Fox, editors, *Intelligent scheduling*, 213-39. Morgan Kaufmann, San Francisco, CA, 1994.